

# 20 GIỜ ĐẦU TIÊN

CÁCH HỌC NHANH BẤT CỨ THỨ GÌ



**JOSH  
KAUFMAN**

# MỤC LỤC

[Lưu ý độc giả](#)

[Chân dung tác giả – người nghiên học](#)

[Mười nguyên tắc học kỹ năng nhanh](#)

[Mười nguyên tắc học hiệu quả](#)

[Yoga](#)

[Lập trình](#)

[Gõ 10 ngón](#)

[Cờ vây](#)

[Lướt ván](#)

[Lời bạt](#)

# Lưu ý độc giả

Cuộc đời thì ngắn ngủi, còn những điều phải học lại quá nhiều.

–GEOFFREY CHAUCER<sup>(1)</sup>, tác giả bài thơ *Nghị hội của đám đông*, 1374

...

“Có rất nhiều việc tôi muốn làm... nhưng lại có quá ít thời gian”. Đó là chuyện thường thấy của cuộc sống hiện đại.

Hãy dành một chút thời gian để xem có bao nhiêu việc bạn muốn học cách làm. Danh sách của bạn có những việc gì? Điều gì ngăn cản bạn bắt đầu?

Thường là vì hai lý do: Thời gian và kỹ năng.

Có một sự thật không lấy gì làm dễ chịu, đó là: Những trải nghiệm đáng giá nhất trong đời thường đòi hỏi phải có kỹ năng ở một cấp độ nào đó. Cần phải có thời gian và nỗ lực mới có được kỹ năng – mà thời gian thì chúng ta không có, còn nỗ lực lại là thứ chúng ta ngại gom góp.

“Một ngày nào đó tôi sẽ học, khi tôi có thời gian”.

Thành thật mà nói, ngồi xem tivi và lướt web bao giờ cũng dễ dàng hơn... Vì vậy đó chính là điều mà hầu hết chúng ta vẫn làm, và để cho khao khát của chúng ta vẫn mãi chỉ là những giấc mơ.

Vẫn còn một sự thật không lấy gì làm dễ chịu, đó là: Có rất nhiều việc chẳng có gì thú vị cho tới khi chúng ta giỏi việc đó. Kỹ năng nào cũng có cái mà tôi gọi là *rào cản thất vọng* – một giai đoạn bạn không thạo (kỹ năng đó) và đau đớn nhận ra sự thật đó. Tại sao phải bắt đầu một việc mà bạn biết là mình sẽ không giỏi việc đó chứ?

Chẳng phải sẽ rất tuyệt nếu có thể thành thạo những kỹ năng mới mà không cần phải lo lắng nhiều sao? Làm sao để nhanh chóng vượt qua rào cản thất vọng, có thể chuyển tới phần thú vị? Làm sao để tốn ít thời gian cho việc bối rối và lúng túng, có thêm thời gian vui vẻ?

Liệu có thể học được những kỹ năng mới theo cách tốn ít thời gian và công sức hơn không?

Từ kinh nghiệm của riêng tôi, xin trả lời là: Có thể.

Cuốn sách này là cuộc tìm kiếm của cá nhân tôi nhằm kiểm tra tính nghệ thuật và khoa học của *việc học kỹ năng nhanh chóng* – làm thế nào để học bất cứ kỹ năng nào nhanh nhất có thể. Mục đích của cuốn sách này là giúp bạn học được những kỹ năng mới trong khoảng thời gian kỷ lục.

Theo kinh nghiệm của tôi, sẽ mất khoảng 20 giờ thực hành để phá vỡ rào cản thất vọng: để đi từ mức hoàn toàn không biết gì về việc bạn đang cố gắng làm tới mức có thể làm tốt việc đó một cách đáng ngạc nhiên.

Cuốn sách này là cách tiếp cận mang tính hệ thống để học những kỹ năng mới nhanh nhất có thể. Phương pháp mang tính phổ quát. Vấn đề không phải là bạn muốn học ngoại ngữ, viết tiểu thuyết, vẽ chân dung, khởi nghiệp hay lái máy bay. Nếu bạn đầu tư ít nhất 20 tiếng để học những điều cơ bản của một kỹ năng, bạn sẽ ngạc nhiên với mức độ thành thạo mà bạn có thể đạt được.

Bất kể kỹ năng bạn muốn học là gì, cuốn sách này sẽ giúp bạn đạt được kỹ năng đó với ít thời gian và năng lượng hơn. Chỉ cần một chút nỗ lực tập trung, có chiến lược, bạn sẽ thấy mình nhanh chóng tiến bộ mà không thất vọng.

Trong cuốn sách này, chúng ta sẽ bắt đầu với những nguyên tắc học kỹ năng nhanh: Làm cách nào để học những kỹ năng mới nhanh nhất có thể. Những ý tưởng và bài tập thực hành này không hề phức tạp, nên sẽ không mất quá nhiều thời gian để học.

Sau đó, tôi sẽ giải thích cách áp dụng những nguyên tắc này trong thực tế bằng cách chỉ cho các bạn biết, tôi đã học năm kỹ năng mới dưới đây trong vòng chưa tới 20 tiếng mỗi kỹ năng với không quá 90 phút thực hành mỗi ngày như thế nào.

- Phát triển bài tập yoga cá nhân
- Viết chương trình máy tính dựa trên web
- Học lại cách gõ 10 ngón
- Khám phá boardgame<sup>(2)</sup> cổ nhất và phức tạp nhất trong lịch sử

Lướt ván

Tôi hi vọng cuốn sách này có thể khuyến khích bạn phủ bụi danh sách những việc “muốn làm” của bạn, kiểm tra lại danh sách đó và cam kết học được điều mới mẻ nào đó.

*Josh Kaufman*

*Fort Collins, Colorado, Mỹ*

# Chân dung tác giả – người nghiên học

---

Ebook miễn phí tại : [www.SachMoi.net](http://www.SachMoi.net)

**Sáng nào thức dậy, tôi cũng quyết tâm phải thay đổi thế giới và vui vẻ. Đôi khi điều này khiến cho việc lập kế hoạch theo ngày của tôi trở nên khó khăn.**

**–E. B. WHITE, tác giả cuốn *Mạng nhện của Charlotte***

...

**X**in chào các bạn, tên tôi là Josh Kaufman, và tôi là một người nghiên học.

Các giá sách ở nhà và ở văn phòng của tôi đều đầy ắp sách, dụng cụ và những thiết bị chưa từng được sử dụng, đủ các thể loại. Hầu hết những thứ đó đều dần dần bị bụi phủ.

Tôi có một danh sách dài ngoằng những thứ “cần học”. Giỏ hàng trên trang Amazon.com hiện tại của tôi có 241 món hàng – tất cả đều là những cuốn sách tôi muốn đọc. Tôi không thể bước vào một hiệu sách mà không quay ra với ba hoặc bốn cuốn sách mới, để thêm vào “bộ sưu tập” 852 cuốn mà tôi đang có.

Mỗi ngày, tôi lại nghĩ ra một ý tưởng cho một dự án khác hay một trải nghiệm mới, và tôi lại thêm vào danh sách vẫn không ngừng dài thêm của mình có tên là “một ngày nào đó/có thể”. Chỉ nhìn vào những việc mà tôi muốn học cách làm cũng đủ khiến tôi thấy choáng ngợp, vì vậy tôi không mấy khi nhìn vào danh sách đó.

Tôi muốn học cách phát triển công ty xuất bản của mình. Tôi muốn học cách quay và biên tập video. Tôi muốn sản xuất được một chương trình phát thanh. Tôi muốn học cách tổ chức những buổi thảo luận và giảng dạy tốt hơn.

Tôi có những ý tưởng cho sản phẩm mới, nhưng tôi không biết làm thế nào để phát triển nó. Tôi có các ý tưởng về chương trình máy tính mới, nhưng tôi không biết làm thế nào để tạo ra chúng. Tôi có nhiều ý tưởng viết lách trong

đầu hơn là thời gian và năng lượng để viết ra.

Tôi muốn học cách vẽ. Tôi muốn học cách chèo thuyền kayak. Tôi muốn học cách câu cá bằng ruồi nhân tạo. Tôi muốn học leo núi. Tôi muốn có thể chơi được ghi ta, đàn ukulele, piano và violin điện.

Có những trò chơi tôi đã thích trong nhiều năm như trò cờ vây, nhưng tôi vẫn chưa học được cách chơi. Có những trò chơi tôi đã biết cách chơi như cờ tướng, nhưng tôi không thực sự giỏi, vì vậy những trò đó không thú vị lắm, và tôi không chơi chúng thường xuyên.

Tôi thích ý tưởng chơi gôn, nhưng trận nào tôi chơi cũng đều hóa thành trò cười cho mọi người. (Tôi thường nói rằng tôi chơi gôn ma-ra-tông vì khi đánh được 18 lỗ, tôi đã chạy được cả một vòng ma-ra-tông.)

Có vẻ như mỗi ngày tôi lại thêm một số kỹ năng mới vào danh sách những việc tôi muốn mình có thể làm được – danh sách dài vô tận. Khi có quá nhiều thứ cần học, tôi sẽ có ít thời gian hơn.

Về bản chất, tôi là người thích tự mình làm việc của mình. Nếu có việc gì cần làm, tôi muốn tự mình làm hơn là nhờ người khác giúp đỡ. Ngay cả khi người khác có thể làm việc đó nhanh hơn hoặc tốt hơn tôi, tôi cũng không nỡ tước mất cái “thú” học để trải nghiệm của bản thân.

Để làm cho mọi việc phức tạp hơn, Kelsey, vợ tôi, còn mở một công ty riêng, cung cấp các khóa đào tạo liên tục cho những giảng viên dạy yoga. Công ty mang lại lợi ích cho cả hai chúng tôi, vì thế luôn có rất nhiều việc cần phải làm.

Để cuộc sống trở nên thú vị hơn, chúng tôi đã chào đón thành viên mới của gia đình, con gái Lela. Lela mới được chín tháng tuổi khi tôi viết cuốn sách này.

Trước khi có Lela, tôi và Kelsey đã thống nhất, nếu có con, chúng tôi muốn đặt việc tự mình nuôi con lên hàng đầu. Một trong những lý do chính khiến tôi từ bỏ công việc quản lý trước đây của mình tại một tập đoàn thuộc danh sách Fortune 500 là để có thể làm việc linh hoạt ở nhà, tự đặt lịch trình cho mình, và dành càng nhiều thời gian càng tốt bên gia đình.

Tôi và Kelsey cùng chia sẻ trách nhiệm làm cha mẹ. Vì cả hai chúng tôi đều làm việc tại nhà nên Kelsey làm việc buổi sáng khi tôi chăm sóc Lela. Buổi chiều, Kelsey trông con, còn tôi làm việc tới lúc ăn tối. Như vậy tôi có khoảng

25 tiếng để làm việc mỗi tuần, chưa kể khoảng thời gian tôi có thể tranh thủ được khi Lela chợp mắt.

Sau khi Lela chào đời, tôi có cảm giác tôi không có đủ thời gian để làm việc, nói gì tới chuyện học thêm kỹ năng mới. Với một người nghiện học như tôi, đó là cả một vấn đề bức xúc.

Tôi không muốn từ bỏ việc học và hoàn thiện bản thân, kể cả khi có những trọng trách mới. Tôi không có nhiều thời gian rảnh, nhưng tôi sẵn sàng đầu tư những gì tôi có theo cách thông minh nhất có thể.<sup>14</sup>

Đó chính là lý do khiến tôi quan tâm tới việc học kỹ năng nhanh: Phương pháp học kỹ năng mới một cách nhanh chóng.

Tôi muốn tiếp tục học thêm các kỹ năng, nhưng tôi không muốn quá trình đó kéo dài mãi. Tôi muốn nắm bắt được các điểm mấu chốt một cách nhanh chóng để có thể tiến bộ vượt bậc mà không phải cảm thấy thất vọng thường xuyên .

Tôi tin là bạn có thể hiểu được. Bạn có bao nhiêu thời gian “rảnh” mỗi ngày sau khi hoàn thành công việc và nghĩa vụ với gia đình? Bạn có cảm thấy bạn cần 36 đến 48 tiếng một ngày để cuối ngày mới có thời gian ngồi xuống học một thứ gì đó mới mẻ?

Có một câu nói thế này: “Hãy làm việc theo cách thông minh hơn, đừng làm việc chăm chỉ hơn”. Điều này có nghĩa là quá trình học kỹ năng thực ra không phải là số giờ bạn dành cho nó... mà là điều bạn thu lượm được từ quãng thời gian đó.

## Ông vớ vẩn quá, Malcolm Gladwell

Năm 2008, Malcolm Gladwell viết cuốn sách *Những kẻ xuất chúng*. Trong cuốn sách đó, tác giả đã cố gắng giải thích lý do khiến một số người thành công hơn những người khác.

Một trong những ý tưởng mà Gladwell nhắc đi nhắc lại là “nguyên tắc 10.000 giờ”. Dựa trên nghiên cứu của tiến sỹ K. Anders Ericsson của Đại học bang Florida, để đạt được biểu hiện xuất chúng cần 10.000 giờ (bình quân) để luyện tập.



10.000 giờ tương đương với 8 giờ luyện tập mỗi ngày trong khoảng ba năm rưỡi liên tục, không ngừng nghỉ, không có cuối tuần và cũng không có nghỉ lễ. So với tiêu chuẩn 260 ngày làm việc một năm không gián đoạn thì khoảng thời gian đó tương đương với một công việc toàn thời gian trong gần 5 năm. Đó là trong trường hợp bạn dành 100% thời gian và tận dụng 100% năng lượng cùng nỗ lực của mình.

Trên thực tế, cấp độ tập trung chú ý này thực sự là một thử thách. Ngay cả những người thuộc đẳng cấp thế giới trong nhiều lĩnh vực cạnh tranh gay gắt (ví dụ như trong âm nhạc và thể thao chuyên nghiệp) cũng chỉ có thể tập trung năng lượng trong khoảng ba tiếng rưỡi để luyện tập mỗi ngày. Điều đó có nghĩa là có thể mất cả thập kỷ hoặc nhiều thời gian hơn nữa để phát triển một kỹ năng tới mức thành thạo.

Nghiên cứu của tiến sỹ Ericsson chỉ ra rằng, về bản chất, nếu bạn muốn thành thạo một kỹ năng mới, bạn sẽ phải trải qua một quá trình dài. Để trở thành người giỏi nhất thế giới về bất cứ việc gì cũng đòi hỏi nhiều năm luyện tập không ngừng nghỉ. Nếu bạn không chịu bỏ thời gian và công sức, bạn sẽ chỉ luôn theo sau và bị lu mờ bởi những người đó.

*Những kẻ xuất chúng* lọt vào danh sách những cuốn sách bán chạy nhất và nằm trong danh sách này suốt ba tháng. Chỉ qua một đêm, “nguyên tắc 10.000 giờ” đã có mặt ở khắp mọi nơi.

Không những phải dành thời gian để luyện tập... giờ bạn còn phải bỏ ra 10.000 giờ nữa? Hầu hết chúng ta đều tự cho mình là người may mắn nếu có thể dành ra vài tiếng mỗi tuần. Vậy thì tại sao phải quan tâm tới việc bỏ ra quá nhiều thời gian như vậy để giỏi một việc gì đó?

Ôi Chúa ơi, hãy nhìn vào công việc của tôi và thất vọng đi!  
Trước khi từ bỏ hết hi vọng, hãy xét tới điều này.

Có một phần trong nghiên cứu của tiến sỹ Ericsson rất dễ bị xem thường: Đó là nghiên cứu về *biểu hiện ở mức độ chuyên gia*. Nếu bạn mong muốn trở thành một Tiger Woods tiếp theo, có lẽ bạn cần phải dành ra ít nhất 10.000 tiếng luyện tập bài bản và có hệ thống tất cả các khía cạnh của môn. Hầu hết các môn thủ chuyên nghiệp đều bắt đầu chơi từ khi còn nhỏ và đã luyện tập không ngừng nghỉ trong ít nhất 17 năm. Để có thể mang đẳng cấp thế giới cần phải có

thời gian.

Ngược lại, chuyện gì sẽ xảy ra nếu thắng giải PGA không phải là mục tiêu của bạn? Chuyện gì sẽ xảy ra nếu bạn chỉ muốn chơi gôn đủ tốt để có thể chơi thoải mái, không làm bản thân mất mặt, có một khoảng thời gian vui vẻ và có cơ hội chiến thắng trong giải đấu các câu lạc bộ địa phương?

Đó lại là một vấn đề hoàn toàn khác. Muốn đạt đẳng cấp thế giới phải cần tới 10.000 giờ nỗ lực tập trung, nhưng phát triển khả năng để thể hiện *đủ tốt vì mục đích riêng của bản thân* thì không đòi hỏi phải đầu tư nhiều như thế.

Nói vậy không có nghĩa là làm giảm giá trị của việc “luyện tập bài bản” mà tiến sỹ Ericsson đã đề cập: Luyện tập có mục đích và có hệ thống nhằm cải thiện kỹ năng. Luyện tập bài bản là cốt lõi của việc học hỏi kỹ năng. Vấn đề là cần luyện tập bài bản đến mức nào để đạt được mục tiêu của bạn. Thường thì điều đó không cần nhiều như bạn tưởng.

## Chất lượng, không phải số lượng

Tiếp nhận khái niệm *vừa đủ* là chìa khóa để học kỹ năng nhanh. Trong cuốn sách này, chúng ta sẽ thảo luận về việc phát triển khả năng, chứ không phải sự thành thạo đạt đẳng cấp thế giới. Chúng ta sẽ xử lý khúc cua của đường cong học hành và nấn thẳng nó càng nhanh càng tốt.

Hãy để 10.000 giờ cho những người chuyên nghiệp. Chúng ta sẽ bắt đầu với 20 giờ nỗ lực chuyên tâm, tập trung, thông minh.

Chúng ta đang nhắm tới những kết quả mà bản thân mong muốn chỉ với một phần nỗ lực. Có thể bạn sẽ không bao giờ giành được huy chương vàng, nhưng bạn sẽ đạt được những phần thưởng bạn quan tâm trong thời gian ngắn hơn.

Nếu cuối cùng bạn quyết định phải thành thạo kỹ năng đó, bạn sẽ có cơ hội tốt hơn để thành công nếu bạn bắt đầu với 20 tiếng học kỹ năng nhanh. Bằng cách biết bản thân đang hướng tới điều gì, học những nguyên tắc cơ bản, luyện tập theo cách thông minh và phát triển một quy trình luyện tập, bạn sẽ tiến bộ nhanh hơn và đều đặn hơn. Bạn sẽ đạt được trình độ chuyên gia trong thời gian kỷ lục.

# Thế nào là học kỹ năng nhanh?

Học kỹ năng nhanh là một quá trình – một cách chia nhỏ kỹ năng mà bạn đang cố làm chủ thành những phần nhỏ nhất có thể, xác định phần nào là quan trọng nhất, sau đó chú tâm luyện tập phần đó trước tiên. Chỉ đơn giản vậy thôi.

Học kỹ năng nhanh có bốn bước chính:

**Chia nhỏ** kỹ năng thành những phần kỹ năng nhỏ nhất có thể;

**Học** những phần kỹ năng nhỏ nhất để có thể luyện tập theo cách thông minh và tự điều chỉnh trong suốt quá trình luyện tập;

**Loại bỏ** những rào cản về mặt sinh lý, tinh thần và tình cảm xuất hiện trong quá trình luyện tập;

**Luyện tập** những phần kỹ năng quan trọng nhất trong vòng 20 tiếng là ít nhất.

Học kỹ năng nhanh không phải là môn khoa học thần tốc. Chỉ đơn giản là bạn quyết định điều cần luyện tập, tìm ra cách tốt nhất để luyện tập, dành thời gian để luyện tập, sau đó luyện tập cho tới khi bạn đạt được mục tiêu của mình.

Chẳng có gì thần kỳ trong đó đâu – chỉ là sự nỗ lực thông minh, có chiến lược, được đầu tư vào việc bạn quan tâm mà thôi. Chỉ cần chuẩn bị một chút, bạn sẽ có được những kỹ năng mới một cách nhanh chóng, tốn ít công sức hơn.

Nói như vậy không có nghĩa là kết quả sẽ có ngay lập tức. Khao khát muốn được thỏa mãn ngay lập tức chính là một trong những lý do chính khiến mọi người không học được kỹ năng mới một cách nhanh chóng.

## Quan niệm sai trong “Ma trận”

Bạn có nhớ cảnh trong phim *Ma trận* khi Keanu Reeves mở mắt ra, chớp mắt vài lần và thì thầm “Tôi biết Kung fu” không?

Đáng tiếc là tôi phải nói cho bạn biết: Học kỹ năng nhanh không có nghĩa là

nhanh như thế.

Hollywood đã chơi xỏ chúng ta một vô liên quan tới học kỹ năng nhanh. Dù chắc chắn sẽ thật tuyệt nếu có thể học cách điều khiển một chiếc trực thăng Bell 212 trong vòng 5 giây bằng cách tải trực tiếp phần mềm vào não chúng ta, nhưng khoa học hiện nay còn cách xa khoa học viễn tưởng.

Cho tới khi điều này trở thành hiện thực, “nhanh” có nghĩa là dành thời gian ít hơn so với bình thường để học một kỹ năng nếu bạn trải qua một quá trình giống như hầu hết mọi người: mù mờ, lung tung và không nhất quán.

Một trong những kỹ năng chuyên nghiệp đầu tiên mà tôi học được là phát triển trang web: có thể xây dựng những trang web hữu ích, có công dụng. Bắt đầu với trang web cơ bản Angelfire.com năm 1996, tôi đã tự học cách đọc và viết HTML và CSS (ngôn ngữ chung của web), sử dụng Adobe Photoshop để chỉnh sửa hình ảnh, định cấu hình máy chủ web, và duy trì các hệ thống hỗ trợ công việc của tôi.

Tôi không được học những việc đó từ thời cấp ba hay đại học. Mặc dù tôi đã hoàn thành chương trình cử nhân chuyên ngành hệ thống thông tin doanh nghiệp, nhưng những kiến thức tôi đã học được trên giảng đường hoàn toàn không giống với những gì tôi vẫn làm hàng ngày.

Tôi đã học kỹ năng phát triển web bằng cách thử ngẫu nhiên và tìm hiểu trong quá trình thực hiện. Mỗi khi gặp một công nghệ hoặc công cụ mới có thể cải thiện trang web hoặc làm giảm khối lượng công việc của tôi, tôi liền thử nghiệm ngay. Sau một khoảng thời gian dài, các kỹ năng của tôi đều được cải thiện.

Cách học hỏi các kỹ năng phát triển web ngẫu hứng của tôi chỉ phục vụ một mục đích: Tôi có được một công việc dựa trên những kỹ năng đó, và hiện tại tôi đã “bán” được các thông tin về web để kiếm sống. Nhiệm vụ đã được hoàn thành, nhìn từ một góc độ nào đó.

Mặt khác, tôi đã phải học hỏi mọi thứ một cách khó khăn. Bạn chắc chắn có thể đạt tới trình độ của tôi trong vòng chưa đầy 15 năm nếu bạn tiếp cận chúng theo cách mang tính hệ thống. Nhưng nếu bạn luyện tập những kỹ năng này một cách thông minh, bạn hoàn toàn có thể đạt được trình độ ngang ngửa tôi trong vòng một tháng.

Đó chính là điều tôi muốn nói khi sử dụng cụm từ *học kỹ năng nhanh*. Nếu

bạn có thể học hầu hết những gì tôi biết về thiết kế web trong vòng một tháng tập trung thay vì 15 năm thì đó là cả một sự tiến bộ rõ rệt. Điều đó cũng hoàn toàn có thể xảy ra.

Thời gian bạn cần để học một kỹ năng mới chủ yếu phụ thuộc vào lượng thời gian tập trung mà bạn có thể đầu tư cho việc chủ động luyện tập và trải nghiệm thông minh, cũng như phụ thuộc vào mức độ thành thạo mà bạn mong muốn.

Đừng kỳ vọng sẽ đạt được kết quả chỉ sau một đêm. Bạn chỉ nên kỳ vọng tổng thời gian bạn cần đầu tư cho việc học sẽ ít hơn so với khi bạn lao vào quá trình học mà không có chiến lược cụ thể.

Trước khi chúng ta khám phá phương pháp này một cách chi tiết, có một điều bạn nên biết: Học kỹ năng nhanh không có điểm gì chung với “học cách học” ở trường. Học thuật gần như không có gì trùng với học kỹ năng, chứ chưa nói tới việc nhanh chóng thành thạo kỹ năng đó.

## Học kỹ năng so với việc học bình thường

Cũng như rất nhiều học sinh trung học ở Mỹ, tôi đã từng học ngoại ngữ. Trong bốn năm liền, ngày nào tôi cũng có mặt ở lớp học tiếng Tây Ban Nha. Điểm số của tôi rất cao, đều là điểm A.

Hiện nay, ngoài một số câu *hola*, *cómo estás* và *muuy bien*, tôi không thể nói chuyện được với một người Tây Ban Nha bản xứ để tự cứu mình. (Tôi thậm chí còn không biết phải nói gì nếu tôi có một ngày tồi tệ.)

Ngược lại hoàn toàn, bạn tôi, Carlos Miceli lại trưởng thành cùng với tiếng Tây Ban Nha ở Argentina. Hồi cấp ba, Carlos muốn nói tiếng Anh trôi chảy nên cậu ấy đã quyết định sẽ nỗ lực để nói chuyện với người Anh bản xứ càng nhiều càng tốt. Trong quá trình đó, cậu ấy đã phát hiện ra Skype và lập ra trang web riêng của mình để có thể nói và viết tiếng Anh thường xuyên.

Carlos chưa bao giờ theo một khoá học nào. Cậu ấy cũng hoàn toàn không biết các quy tắc ngữ pháp tiếng Anh. Cậu ấy còn không thể nói cho bạn biết cậu ấy biết tiếng Anh ở mức độ nào. Điều đó quả thực không quan trọng. Quan trọng là cậu ấy có thể viết và nói tiếng Anh trôi chảy.

Tiến sỹ Stephen Krashen của Đại học Nam California là một chuyên gia trong lĩnh vực học ngoại ngữ thứ hai. Một trong những nhận định chính của

Krashen là *học hỏi, tiếp thu* ngoại ngữ, hoàn toàn khác với *học* ngoại ngữ.

Khi còn đi học, tôi đã học rất nhiều tiếng Tây Ban Nha. Tôi đã học hàng nghìn từ vựng, chia động từ và quy tắc ngữ pháp. Tôi đã học tốt tới mức bài kiểm tra nào cũng đạt điểm rất cao.

Tuy nhiên, những bài kiểm tra đó lại chẳng liên quan gì tới khả năng của tôi trong việc luyện tập kỹ năng nói tiếng Tây Ban Nha và hiểu được những gì người Tây Ban Nha nói với tốc độ bình thường của họ. Nếu mục tiêu của tôi là có thể nói tiếng Tây Ban Nha trôi chảy, thì chỉ một vài tuần nỗ lực nói chuyện với người Tây Ban Nha chắc chắn sẽ đem lại kết quả tốt hơn nhiều so với bốn năm học.

Nhưng thời kỳ đó, nói được tiếng Tây Ban Nha trôi chảy lại không phải là mục tiêu của tôi. Tôi chỉ muốn qua được kỳ kiểm tra cuối cùng mà thôi. Ngược lại, Carlos lại không đến lớp mà bắt tay vào luyện tập. Thay vì chia động từ, Carlos lại luyện tập điều thực sự quan trọng: Giao tiếp với người khác bằng tiếng Anh.

Xét từ góc độ hiệu quả và giá trị lâu dài thì cách tiếp cận của Carlos hơn hẳn cách của tôi. Điều này không cần phải bàn cãi.

## Giá trị đích thực của việc học

Nói như vậy không có nghĩa là việc học kỹ năng mà bạn muốn thành thạo không quan trọng. Học là điều quan trọng, nhưng không phải theo cách bạn nghĩ. Việc học những khái niệm liên quan tới kỹ năng giúp bạn *tự biên tập* hoặc *tự chỉnh sửa* khi bạn đang luyện tập.

Nếu bạn biết cách chia động từ trong tiếng Tây Ban Nha, bạn có thể dễ dàng tự chỉnh sửa câu nói của bạn hơn khi bạn nói chuyện với người bản xứ. Nếu bạn học các từ vựng thông dụng, bạn có thể hiểu rõ hơn những gì người bản địa nói, đồng thời nhớ được từ hoặc cụm từ thích hợp để sử dụng khi bị tắc.

Tiến sỹ Krashen gọi đó là *giả thuyết giám sát*. Việc học giúp bạn lập kế hoạch và thay đổi bản thân trong khi luyện tập. Đó chính là lý do tại sao việc học lại quan trọng. Vấn đề chỉ nảy sinh khi chúng ta nhầm lẫn giữa việc học với việc học hỏi kỹ năng.

Nếu bạn muốn thành thạo một kỹ năng mới, bạn cần luyện tập nó theo ngữ

cảnh. Việc học sẽ hỗ trợ việc luyện tập, nhưng không thể thay thế cho việc luyện tập được. Nếu biểu hiện là điều quan trọng thì chỉ riêng việc học là không bao giờ đủ.

## Học kỹ năng với rèn luyện

Có sự khác biệt lớn giữa học kỹ năng và rèn luyện. Rèn luyện, trong hoàn cảnh này, có nghĩa là cải thiện một kỹ năng mà bạn đã học được thông qua việc lặp đi lặp lại kỹ năng đó. Đó là việc xảy ra sau khi bạn đã học được một kỹ năng cơ bản nếu bạn muốn tiếp tục tiến bộ.

Lấy việc chạy việt dã làm ví dụ. Hầu hết chúng ta đều học được kỹ năng chạy từ khi còn nhỏ. Ngoài việc đặt chân này trước chân kia và chạy cho tới khi cán mốc 42 km, chẳng có kỹ năng nào mới cần phải học hỏi.

Tuy nhiên, cần phải rất nỗ lực mới có thể tăng cường sức khỏe và cải thiện độ dẻo dai để hoàn thành phần chạy việt dã. Quá trình tăng cường và cải thiện đó chính là rèn luyện. Càng rèn luyện nhiều, bạn càng trở nên khỏe mạnh và càng sớm hoàn thành phần chạy việt dã.

Liên quan tới chạy việt dã còn có một yếu tố nữa, đó là làm cách nào để đăng ký tham gia cuộc đua, làm thế nào để đủ tiêu chuẩn tham gia những sự kiện lớn như giải chạy việt dã Boston.

Chẳng hạn, một vấn đề nhỏ như ma sát giữa áo và da bạn sẽ không phải là vấn đề lớn nếu bạn chỉ chạy 5 km, nên hầu hết những người chạy đều không lăn tăn gì về vấn đề đó. Nhưng ma sát sẽ trở thành vấn đề lớn nếu bạn chạy khoảng 42 km.

Rèn luyện và học hỏi chắc chắn sẽ giúp cho việc hoàn thành cuộc đua trở nên dễ dàng hơn, nhưng đó không phải là tiếp thu kỹ năng. Nếu không tiếp thu kỹ năng thì việc rèn luyện là điều không thể và chẳng có ích gì. Chuẩn bị và chăm sóc sức khỏe có thể khiến cho việc tiếp thu kỹ năng trở nên dễ dàng hơn, nhưng hai việc đó không thể thay thế được việc rèn luyện.

Tuy nhiên, học lại cách chạy ở cấp độ cơ bản là tiếp thu kỹ năng. Những thủ thuật như chạy bộ khí công (*Chirunning*) giúp người chạy học được kỹ năng di chuyển theo cách tối thiểu hóa sức lực và tránh mất đà khi tiến về phía trước giữa các bước sai. Luyện tập một chút, người chạy có thể học được kỹ năng cốt lõi của việc chạy và điều này sẽ được củng cố khi rèn luyện sau này.

## Tiếp thu kỹ năng với giáo dục và đào tạo

Bất chấp những nỗ lực cao thượng của các giáo viên và giáo sư trên khắp thế giới, những phương pháp giáo dục và đào tạo hiện đại lại chẳng có tác động gì tới việc tiếp thu kỹ năng.

Tiếp thu kỹ năng đòi hỏi phải rèn luyện kỹ năng đó. Việc này yêu cầu phải có những giai đoạn chuyên tâm tập trung và kéo dài. Nó đòi hỏi tính sáng tạo, linh động và tự do để hình thành tiêu chuẩn thành công của bản thân bạn.

Đáng tiếc là hầu hết phương pháp giáo dục và đào tạo hiện đại nhất chỉ yêu cầu việc tuân thủ đơn giản. Mục tiêu chính không phải là tiếp thu được những kỹ năng hữu ích, mà là để chứng thực việc hoàn thành bộ tiêu chuẩn tùy tiện nhất, được hình thành bởi những ủy ban tiêu chuẩn, vì mục tiêu thông qua một số phẩm chất nhất định mà một vài bên thứ ba nào đó có vẻ quan tâm.

Sáng tạo, linh động và tự do thử nghiệm – các yếu tố quan trọng của việc tiếp thu kỹ năng nhanh – lại mâu thuẫn với quá trình đào tạo. Nếu các tiêu chuẩn quá linh động, thì chúng đã không phải là các tiêu chuẩn, đúng không nào?

Đáng tiếc là giáo dục và đào tạo khắt khe có thể *ngăn chặn* đáng kể việc tiếp thu kỹ năng. Vấn đề chính là chi phí cơ hội: Nếu điều kiện cần thiết để được đào tạo lại khắt khe tới mức làm giảm khả năng dành thời gian rèn luyện kỹ năng theo yêu cầu, thì quá trình đào tạo đó lợi bất cập hại.

Lấy một người thông minh, có động lực và quan tâm tới việc hình thành một công ty phần mềm làm ví dụ. Việc hoàn thành tám bằng cử nhân về khoa học máy tính tại một trường đại học danh giá thường mất ít nhất bốn năm.

Cuối giai đoạn bốn năm, anh chàng sinh viên mới tốt nghiệp này đã bỏ ra hàng nghìn giờ học thuật toán và phân tích các trình biên dịch<sup>(1)</sup> để vượt qua hàng chục kỳ thi, nhưng vẫn không hề gần mục tiêu thành lập công ty phần mềm của mình. Anh chàng sinh viên đáng thương này của chúng ta đã ghi nhớ được rất nhiều thứ về lập trình máy tính, ít nhất là tạm thời, nhưng vẫn không biết làm thế nào để tạo ra được một chương trình máy tính mà mọi người thấy hữu dụng, đáng mua.

Việc hình thành một công ty phần mềm đòi hỏi phải tiếp thu những kỹ năng mới: học ngôn ngữ lập trình, thiết lập và duy trì hệ thống máy tính, tìm kiếm các công cụ và chương trình sẵn có, tạo mẫu thử nghiệm, tìm kiếm người sử



dụng ban đầu, tìm nguồn tài trợ cần thiết và xử lý những nhiệm vụ hành chính doanh nghiệp thông thường.

Có sự chông chéo nào giữa việc hình thành một doanh nghiệp và việc có được một chứng chỉ giáo dục không? Chắc chắn là có. Nhưng cần phải lưu ý là: Hầu hết nỗ lực của việc có được chứng chỉ đều được dành cho quá trình đáp ứng các yêu cầu. Liệu những yêu cầu đó có giúp bạn đạt được những kỹ năng cần có để tồn tại được trong thế giới thực hay không lại là mối quan tâm thứ ba.

Trong cuốn sách đầu tiên của tôi, *MBA cá nhân: Bậc thầy nghệ thuật kinh doanh* (The Personal MBA: Master the Art of Business) (2010), tôi đã giải thích tại sao tôi quyết định bỏ chương trình đào tạo đại học về kinh doanh để tự học những nguyên tắc về thực tế kinh doanh hiện đại và xây dựng công ty riêng của mình. Bằng cách không học trường kinh doanh và dành thời gian để gây dựng doanh nghiệp thực sự, tôi đã học được rất nhiều và tiết kiệm được 150.000 đô la trong suốt cả quá trình. Xét từ việc tôi muốn hoàn thành, việc dành thời gian để tự mình tiếp thu những kỹ năng kinh doanh còn hiệu quả và tốt hơn việc tới học trường kinh doanh.

Nếu bạn muốn làm tốt bất cứ việc gì coi trọng biểu hiện thực tế, thì bạn cần phải rèn luyện kỹ năng đó trong bối cảnh thực tế. Chỉ học thôi thì không bao giờ đủ.

## Sinh lý thần kinh về kỹ năng: Tính mềm dẻo của não bộ và bộ nhớ cơ bắp

Một điều cuối cùng trước khi chúng ta bước vào tìm hiểu chi tiết về tiếp thu kỹ năng nhanh: Bạn phải nhận thức được là bạn có khả năng học kỹ năng mới.

Điều này có vẻ kỳ cục, nhưng thật dễ tin rằng các kỹ năng của bạn là cố định – rằng bạn giỏi, có tài năng, hoặc có năng khiếu về việc gì đó... hoặc không.

Trong cuốn *Tư duy: Tâm tâm lý học về thành công* (Mindset: The New Psychology of Success) (2007), nhà tâm lý học Carol Dweck đã trích dẫn phần lớn nội dung của một nghiên cứu chỉ ra rằng, các cá nhân thường có một trong hai cách nhìn nhận trí óc của họ làm việc như thế nào.

Theo tiến sỹ Dweck, những người có tư duy “cố định” cho rằng kỹ năng và tài năng là những thứ mang tính bẩm sinh, rằng con người sinh ra đã có một số

khả năng nhất định. Nếu người có tư duy cố định “không giỏi toán” thì có nỗ lực luyện toán thế nào cũng chỉ là sự lãng phí. Sao phải bận tâm khi bạn không bao giờ có thể giỏi (toán) được nhỉ?

Ngược lại, những người có tư duy “phát triển” lại cho rằng, kỹ năng và tài năng sẽ phát triển cùng với sự rèn luyện và tính kiên nhẫn. Nếu một người có tư duy phát triển làm sai một vài phép toán, điều đó không có nghĩa là bởi vì họ không có tài năng toán học, mà chỉ là họ chưa rèn luyện đủ. Với sự rèn luyện và bền bỉ, việc họ có thể làm chủ kỹ thuật đó chỉ là vấn đề thời gian.

Có tin tốt cho bạn đây, nếu bạn thuộc nhóm những người có tư duy cố định: Có nghiên cứu chỉ ra rằng, tất cả các bộ não đều có khả năng cải thiện kỹ năng và khả năng thông qua việc rèn luyện. Khuynh hướng di truyền có tồn tại, nhưng chỉ là thứ yếu so với sức mạnh của việc rèn luyện một cách thông minh, có tập trung. Bạn có thể cải thiện bất cứ kỹ năng nào, miễn là bạn sẵn sàng rèn luyện.

Não bộ của con người có tính *mềm dẻo* – thuật ngữ được các nhà khoa học về thần kinh sử dụng để chỉ việc bộ não có thể thay đổi về mặt vật lý để phản ứng với môi trường, hành động và hệ quả của những hành động đó. Khi bạn học bất kỳ kỹ năng mới nào, dù là kỹ năng vật lý hay tinh thần, thì hệ thần kinh trong não bộ của bạn cũng thay đổi khi bạn rèn luyện kỹ năng đó.

Theo lời tiến sỹ John Medina (*Quy luật của bộ não* (Brain Ruler), 2009) thì “nơ-ron hình thành cùng nhau sẽ kích thích lẫn nhau”, tạo thành những kiểu mới độc đáo trong vòng tròn vật lý của não bộ. Theo thời gian, các nơ-ron của bạn bắt đầu kích thích theo cách hiệu quả hơn để phản ứng với những phản hồi bạn nhận được từ môi trường của mình khi rèn luyện.

Nếu bạn làm việc về *kỹ năng động* (nghĩa là kỹ năng liên quan tới chuyển động), thì lúc đầu bạn sẽ thấy khá dè dặt và chậm chạp. Bạn phải nghĩ về mọi việc bạn đang làm, và bạn thường phạm phải những sai lầm khó chịu. Học những điều cơ bản là một cuộc vật lộn không ngừng nghỉ.

Khi bạn rèn luyện, sự phối hợp giữa các cơ sẽ trở nên tự động và đồng bộ với các quá trình tinh thần của bạn. Bạn có được khả năng tập trung tốt hơn đối với những yếu tố không dễ phát hiện trong việc bạn đang làm, và bạn học được phương pháp điều chỉnh cách tiếp cận đối với những phản hồi bạn nhận được từ môi trường.

Bạn bắt đầu làm những việc phát huy tác dụng nhiều hơn và làm những việc

không phát huy tác dụng ít đi. Cuối cùng, bạn có thể thực hiện mà không cần chú ý tới mọi chi tiết.

Theo ngôn ngữ học thuật, quá trình phổ biến này được gọi là học kỹ năng theo “mô hình ba giai đoạn”, và nó được ứng dụng cho cả các kỹ năng về mặt thể chất và tinh thần. Ba giai đoạn đó là:

1. **Giai đoạn nhận thức (ban đầu)** – hiểu việc bạn đang cố gắng làm, nghiên cứu, nghĩ về quá trình và chia nhỏ các kỹ năng thành những phần có thể xử lý được.
2. **Giai đoạn liên tưởng (trung gian)** – rèn luyện nhiệm vụ, chú ý tới phản hồi của môi trường và điều chỉnh cách tiếp cận của bạn dựa trên những phản hồi đó.
3. **Giai đoạn tự quản (sau cùng)** – thực hiện kỹ năng một cách hiệu quả và thành thạo mà không cần suy nghĩ hoặc cân nhắc một cách không cần thiết tới quá trình.

Quá trình tiếp thu kỹ năng thần kinh học này xảy ra ở mọi lúc, ngay cả khi bạn đang đọc câu này. Không có cái gọi là não ngưng trệ. Não của bạn luôn học hỏi, mã hóa và củng cố các kỹ năng mới.

Như tiến sỹ Dweck đã nói trong cuốn *Tư duy* (Mindset) thì: “Não bạn cũng giống như một loại cơ, bạn càng dùng nhiều thì nó càng phát triển”. Bạn càng luyện tập nhiều thì kỹ năng của bạn càng nhanh nhạy, hiệu quả và tự động.

Đó quả thực là một tin tuyệt vời đối với việc tiếp thu kỹ năng. Nếu tâm trí và cơ thể bạn có thể học hỏi để biểu hiện theo những cách mới và tốt hơn, thì chúng ta có thể xác định được cách để đẩy nhanh quá trình đó.

# Mười nguyên tắc học kỹ năng nhanh

---

Ebook miễn phí tại : [www.SachMoi.net](http://www.SachMoi.net)

**Tôi nhận ra rằng trở thành bậc thầy về karate không phải là học 4.000 thế mà là thực hiện một số thế 4.000 lần.**

**– CHET HOLMES, tác giả cuốn *Cỗ máy bán hàng cuối cùng***

...

**B**ây giờ chúng ta đã biết rõ tiếp thu kỹ năng có nghĩa là gì. Hãy cùng kiểm tra xem làm cách nào để thực hiện được điều đó một cách nhanh chóng. Nội dung của chương này là giúp bạn có được một bảng liệt kê những mục cần kiểm tra khi tiếp thu bất cứ kỹ năng mới nào.

Tôi nhận thấy thật hữu ích khi cho rằng những nguyên tắc này là cách để nuôi cây “ám ảnh tạm thời”. Tiếp thu kỹ năng nhanh diễn ra tự nhiên khi bạn tò mò và quan tâm tới điều gì đó tới mức, những quan tâm khác đều bị gạt qua một bên, ít nhất là tạm thời.

Hãy nghĩ về những nguyên tắc này như cách để xác định một kỹ năng đáng để ám ảnh tạm thời, tập trung vào kỹ năng đó, gạt hết những rào cản hoặc những điều gây xao nhãng khiến bạn không rèn luyện hiệu quả được.

Dưới đây là 10 nguyên tắc chính của việc tiếp thu kỹ năng nhanh:

1. Chọn một dự án đáng yêu.
2. Tập trung năng lượng của bạn vào chỉ một kỹ năng.
3. Xác định mức độ thành thạo mà bạn mong muốn.
4. Chia kỹ năng thành những kỹ năng nhỏ hơn.
5. Tìm được những công cụ then chốt.

- Loại bỏ những rào cản để luyện tập.
- Dành thời gian chuyên tâm luyện tập.
- Tạo vòng phản hồi nhanh.
- Luyện tập tính giờ.
- Nhấn mạnh số lượng và tốc độ.

Nhiều nguyên tắc trong số này đối với bạn là bình thường và ỏn. Hãy nhớ: Chỉ biết những nguyên tắc này thôi thì chưa đủ. Bạn cần phải thực sự sử dụng chúng để gặt hái được thành quả.

## 1. Chọn một dự án đáng yêu

Karl Popper là một trong những triết gia vĩ đại nhất của thế kỷ XX. Ông là người đã phổ biến ý tưởng có thể chứng minh một điều gì đó là sai một cách khoa học. Theo thuật ngữ chuyên môn, nếu bạn không thể chứng minh được điều gì đó là sai bằng cách quan sát hoặc thử nghiệm thì đó không phải là khoa học thật sự.

Popper nói rất nhiều điều thông thái, nhưng tôi nghĩ một trong những câu nói thông thái nhất của ông là: “Điều tuyệt vời nhất có thể xảy đến với con người là tìm thấy vấn đề, “phải lòng” vấn đề đó và sống chỉ để cố gắng giải quyết vấn đề đó, trừ khi có một vấn đề khác còn đáng yêu hơn xuất hiện”.

Nếu bạn muốn có một công thức sống thỏa mãn, hiệu quả, thì bạn không được đi ngược với câu nói đó.

Tiếp thu kỹ năng nhanh đòi hỏi phải chọn được một vấn đề hoặc một dự án đáng yêu. Bạn càng hứng khởi với kỹ năng bạn muốn học bao nhiêu thì bạn càng nhanh học được kỹ năng đó.

Trong thực tế, tìm được một dự án đáng yêu là vấn đề mang tính cá nhân. Chẳng hạn, học nói và viết tiếng Trung phổ thông không nằm trong danh sách những kỹ năng cần học hiện tại của tôi, vì tôi không có nhu cầu cấp bách phải học ngay lúc này, và còn rất nhiều dự án khác mà tôi quan tâm hơn. Nếu tôi quyết định chuyển sang học kỹ năng nói tiếng Trung phổ thông trong tương lai, thì điều đó có thể trở nên đáng yêu, nhưng tôi vẫn chưa quyết định.

Mặt khác, tôi đặc biệt quan tâm tới việc học cách chơi cờ vây, trò chơi trên bàn cờ chiến lược cổ nhất thế giới, có nguồn gốc từ Trung Quốc hơn 3.000 năm trước. Đó là một trò chơi tuyệt vời, và tôi muốn học cách chơi từ khi tôi vô tình biết tới trò chơi này vài năm trước.

Học cách chơi cờ vây đòi hỏi phải tìm hiểu. Luật chơi rất đơn giản, nhưng đọc chính xác những kiểu suy luận của những quân cờ đen và trắng trên bàn cờ lại là cả một thử thách. Máy tính đã thống trị cờ nhiều năm nay, nhưng ngay cả những chiếc máy tính tốt nhất cũng gặp khó khăn khi đấu với một người chơi cờ vây nhiều kinh nghiệm.

Bình thường, với những việc bạn quan tâm, bạn sẽ học nhanh hơn những việc bạn không quan tâm. Gần đây tôi đặc biệt quan tâm tới việc học cách chơi cờ vây, vì thế tôi sẽ học cách chơi cờ vây trước, và để dành tiếng Trung phổ thông lại học sau.

Nếu bạn tập trung vào việc học kỹ năng quan trọng nhất (nghĩa là dự án bạn yêu thích nhất) trước những việc khác, bạn sẽ học được kỹ năng đó nhanh hơn trong khoảng thời gian ngắn hơn.

## 2. Tập trung năng lượng của bạn vào chỉ một kỹ năng

Một trong những sai lầm dễ mắc phải nhất khi học kỹ năng mới là bạn cố gắng học quá nhiều kỹ năng cùng một lúc.

Vấn đề đó chỉ là một bài toán đơn giản: Học kỹ năng mới đòi hỏi phải dành thời gian tập trung và chuyên tâm chú ý. Nếu bạn chỉ có một hoặc hai tiếng mỗi ngày để rèn luyện và học hỏi, và bạn dùng khoảng thời gian và năng lượng đó cho 20 kỹ năng khác nhau thì chắc chắn không có kỹ năng nào được nhận đủ thời gian và năng lượng để đạt được tiến bộ đáng kể.

Một số người khó tiếp thu được nguyên tắc này hơn so với những người khác. Cá nhân tôi luôn là một kiểu “người Phục hưng”: Có hàng trăm thứ thuộc hàng trăm lĩnh vực khác nhau mà tôi muốn học tại một thời điểm nào đó. Về mặt tình cảm, thật khó để tôi có thể quyết định trì hoãn học những điều mới mà tôi đã khám phá hoặc nghe nói.

Tuy nhiên, khi tôi cố gắng học tất cả mọi thứ cùng một lúc, tôi lại không thực sự học được gì. Thay vì tiến bộ, tôi lại mất quá nhiều thời gian để chuyển giữa các kỹ năng khác nhau, bối rối và chuyển sang một việc gì đó khác. Đó chính là

công thức để tiếp thu kỹ năng cực kỳ chậm.

Hãy chọn một và chỉ một kỹ năng mới mà bạn muốn học. Hãy dành toàn bộ năng lượng cùng sự tập trung của bạn vào việc học hỏi kỹ năng đó, và gạt những kỹ năng khác sang chế độ chờ. David Allen, tác giả cuốn *Để hoàn thành việc* (Getting Things Done) (2002), đã gợi ý lập danh sách “một ngày nào đó/có lẽ” – danh sách những việc bạn muốn khám phá một lúc nào đó trong tương lai, nhưng chưa đủ quan trọng để bạn tập trung vào chúng ngay lúc này. Bằng cách thêm một việc vào danh sách đó, bạn sẽ tạm thời giải phóng bản thân khỏi trách nhiệm hành động hay suy nghĩ về ý tưởng đó cho tới khi bạn quyết định chuyển nó sang trạng thái hoạt động.

Tôi không thể nhấn mạnh đủ nghĩa là gì. Tập trung vào chỉ một kỹ năng quan trọng là việc vô cùng cần thiết đối với việc học kỹ năng nhanh. Bạn không vĩnh viễn từ bỏ những kỹ năng khác, chỉ là bạn để dành chúng cho sau này mà thôi.

### 3. Xác định mức độ thành thạo mà bạn mong muốn

*Mức độ thành thạo mong muốn* là cụm từ đơn giản. Định nghĩa “đủ tốt” có nghĩa là gì. Bạn muốn thể hiện kỹ năng mà bạn đang học tốt tới mức nào?

Mức độ thành thạo mong muốn là câu khẳng định ngắn gọn về việc bạn muốn thành thạo kỹ năng đó ở mức độ nào. Hãy nghĩ về nó như một câu miêu tả về điều bạn đang cố đạt được, và điều bạn có thể làm khi đạt được. Mức độ thành thạo mong muốn của bạn càng chi tiết, cụ thể bao nhiêu thì càng tốt bấy nhiêu.

Việc xác định mức độ thành thạo giúp bạn tưởng tượng ra sẽ như thế nào khi thể hiện theo một cách cụ thể nào đó. Một khi bạn đã quyết định được chính xác bạn muốn hoặc cần phải tốt tới mức nào, thì sẽ dễ hơn để xác định làm cách nào để đạt được như vậy. Nói như Charles Kettering, nhà phát minh hệ thống khởi động ô tô điện, thì: “Một vấn đề khi được xác định rõ ràng có nghĩa là đã được giải quyết một nửa”.

Cách bạn xác định mức độ thành thạo mong muốn phụ thuộc vào nguyên nhân ban đầu bạn chọn học kỹ năng đó. Nếu mục đích của bạn chỉ là học cho vui, thì mục tiêu của bạn là khi bạn không còn cảm thấy khó chịu và bắt đầu thấy thích việc rèn luyện. Nếu mục đích của bạn là để thành thạo, thì mức độ

thành thạo tới thiểu ban đầu mà bạn có thể chấp nhận được là như thế nào?

Khi bạn đã đạt được mức độ thành thạo mong muốn ban đầu, bạn vẫn có thể chọn tiếp tục nếu bạn muốn. Mức độ thành thạo mong muốn tốt nhất có vẻ như nằm ngoài tầm với, chứ không phải nằm ngoài khả năng có thể.

Về mặt nguyên tắc, mức độ thành thạo mục tiêu của bạn càng thoải mái, bạn càng nhanh chóng học được kỹ năng đó. Nếu bạn tư duy theo tư duy của bậc thầy thế giới, thì có vẻ như đó là một tiêu xảo: Bạn đang hạ thấp rào cản để bạn có thể “thắng” nhanh hơn, đúng không?

Đó chính xác là điều chúng ta đang làm, và không phải là tiêu xảo gì. Hãy nhớ, trình độ bậc thầy thế giới không phải là mục tiêu cuối cùng của việc học kỹ năng nhanh. Chúng ta đang nhắm tới khả năng tiếp thu và hiệu quả với tốc độ lớn nhất, chứ không phải nhắm tới sự hoàn hảo.

Quan trọng là phải nhớ rằng một số kỹ năng có lưu ý về độ an toàn mà bạn nên đưa vào mức độ thành thạo mục tiêu của mình. Bị thương (hoặc mất mạng) vì học kỹ năng mới có thể hủy bỏ mục đích.

#### 4. Phân chia kỹ năng thành những kỹ năng nhỏ

Hầu hết những việc mà chúng ta cho là kỹ năng thực ra đều là một nhóm những kỹ năng nhỏ hơn. Một khi bạn đã xác định được một kỹ năng để tập trung, bước tiếp theo là *phá vỡ* nó – chia nhỏ nó thành những phần nhỏ nhất có thể. Chẳng hạn, chơi gôn là kỹ năng có nhiều chi tiết phụ: Chọn đúng gậy, phát bóng, đánh ra ngoài hố cát...

Một khi kỹ năng được “phá vỡ” phù hợp, sẽ dễ dàng hơn nhiều để xác định kỹ năng nhỏ nào là quan trọng nhất. Bằng cách tập trung vào những kỹ năng nhỏ quan trọng trước, bạn sẽ nhanh chóng tiến bộ mà không cần phải tốn nhiều công sức.

Việc chia nhỏ một kỹ năng cũng giúp bạn dễ dàng tránh được cảm giác quá tải. Bạn không phải luyện tập tất cả các phần của một kỹ năng cùng một lúc. Thay vào đó, sẽ hiệu quả hơn nhiều nếu tập trung vào những kỹ năng nhỏ có thể đem lại kết quả tổng thể khả quan hơn.

Việc chia nhỏ kỹ năng trước khi bắt đầu cũng giúp bạn xác định những phần kỹ năng không quan trọng đối với người mới bắt đầu luyện tập. Bằng cách loại



bỏ kỹ năng nhỏ hoặc những thủ thuật không quan trọng ngay từ đầu quá trình, bạn có thể đầu tư nhiều thời gian và năng lượng cho việc học những kỹ năng nhỏ quan trọng trước.

## 5. Tìm được những công cụ then chốt

Hầu hết các kỹ năng đều có những điều kiện tiên quyết để luyện tập và thực hành. Thật khó để chơi quần vợt nếu bạn không có vợt, hoặc khó có thể học lái trực thăng nếu bạn không được tiếp cận với một chiếc trực thăng.

Bạn cần phải tiếp cận với công cụ, yếu tố hoặc môi trường nào trước khi có thể luyện tập hiệu quả? Làm cách nào bạn có thể có được những công cụ tốt nhất mà bạn có thể tìm thấy và có khả năng mua?

Dành một chút thời gian để xác định những công cụ cần thiết trước khi bắt đầu luyện tập sẽ giúp bạn tiết kiệm được thời gian quý báu. Bằng cách đảm bảo bạn có những nguồn lực cần thiết trước khi bắt đầu, bạn có thể tối đa hóa thời gian luyện tập của mình.

## 6. Loại bỏ những rào cản để luyện tập

Có rất nhiều thứ có thể cản trở việc luyện tập của bạn, khiến cho việc học bất kỳ kỹ năng nào cũng trở nên khó khăn hơn nhiều. Những rào cản này có thể là:

- **Nỗ lực luyện tập trước.** Chẳng hạn như sử dụng công cụ không đúng chỗ, không tìm được đúng công cụ trước khi luyện tập, hoặc bỏ qua việc xây dựng các điều kiện cần thiết.
- **Sử dụng nguồn gián đoạn, ngắt quãng.** Chẳng hạn như sử dụng thiết bị đi mượn hoặc phụ thuộc vào nguồn có thời gian vận hành hạn chế.
- **Những điều gây xao lãng.** Chẳng hạn như ti vi, điện thoại kê, email đến.
- **Rào cản tâm lý.** Chẳng hạn như sợ hãi, nghi ngờ và xấu hổ.

Mỗi một yếu tố này đều có thể khiến cho việc bắt đầu luyện tập khó khăn hơn, và từ đó giảm tốc độ học hỏi của bạn.

Dựa vào sức mạnh tinh thần để vượt qua những rào cản này là một chiến lược chắc chắn thất bại. Chúng ta có rất nhiều sức mạnh tinh thần để có thể sử dụng mỗi ngày, và sẽ là tốt nhất nếu sử dụng sức mạnh tinh thần đó một cách thông minh.

Cách tốt nhất để đầu tư sức mạnh tinh thần vào việc hỗ trợ việc tiếp thu kỹ năng là sử dụng nó để loại bỏ những rào cản nhẹ nhàng đối với việc luyện tập. Bằng cách sắp xếp lại môi trường của bạn để khiến cho việc bắt đầu luyện tập dễ dàng nhất có thể, bạn sẽ học được kỹ năng trong khoảng thời gian ít hơn rất nhiều.

## 7. Dành thời gian chuyên tâm luyện tập

Thời gian bạn dành để học hỏi một kỹ năng mới cần phải lấy từ đâu đó. Đáng buồn là chúng ta thường muốn học kỹ năng mới và tiếp tục thực hiện nhiều hoạt động khác mà chúng ta thích, như xem ti vi, chơi trò chơi điện tử...

“Tôi sẽ học khi tôi tìm được thời gian”, chúng ta thường tự nói với mình như vậy.

Sự thật là đây: “tìm được” thời gian là chuyện hoang đường.

Chưa có ai từng “tìm được” thời gian cho *bất cứ việc gì*, theo nghĩa bất ngờ phát hiện ra một ngăn hàng thời gian đặc biệt, kiểu như bạn tình cờ nhặt được tờ 20 đô la trong túi áo khoác của mình vậy.

Nếu bạn chờ tìm được thời gian để làm việc gì đó, thì bạn sẽ không bao giờ làm được việc đó. Nếu bạn muốn tìm được thời gian, bạn cần phải *tạo ra* thời gian.

Bạn có 24 tiếng để đầu tư mỗi ngày, 1.440 phút, không hơn không kém. Bạn sẽ không bao giờ có nhiều thời gian hơn. Nếu bạn ngủ khoảng 8 tiếng một ngày, bạn còn 16 tiếng để tùy ý sử dụng. Một vài tiếng trong số đó sẽ được dùng để chăm sóc bản thân bạn và những người bạn thương yêu. Số khác được dành để làm việc. Còn lại là thời gian bạn có để học kỹ năng. Nếu bạn muốn cải thiện kỹ năng của mình nhanh nhất có thể, thì bạn có thể dành thời gian học càng nhiều càng tốt.

Cách tốt nhất để tạo ra thời gian học kỹ năng là xác định khoảng thời gian ít giá trị sử dụng, sau đó chọn cách loại bỏ chúng. Giống như làm thí nghiệm, tôi

đề nghị nên có một bảng theo dõi đơn giản, ghi lại cách bạn sử dụng thời gian trong vài ngày. Tất cả những gì bạn cần chỉ là một cuốn sổ.

Kết quả của bảng theo dõi thời gian này sẽ làm bạn bất ngờ: nếu bạn chọn cắt bỏ những khoảng thời gian ít giá trị sử dụng, bạn sẽ có thêm nhiều thời gian để học kỹ năng. Thời gian bạn có thể dành ra mỗi ngày càng nhiều thì tổng thời gian cần phải bỏ ra để học kỹ năng mới càng ít. Tôi khuyến khích nên tạo ra ít nhất 90 phút để luyện tập mỗi ngày bằng cách cắt bỏ càng nhiều hoạt động ít giá trị càng tốt.

Tôi cũng khuyến khích bạn nên *cam kết trước* là sẽ hoàn thành ít nhất 20 tiếng luyện tập. Một khi đã bắt đầu, bạn cần tiếp tục luyện tập cho tới khi đạt tới mốc 20 giờ. Nếu bạn bị kẹt, hãy tiếp tục thực hiện: Bạn không thể dừng cho tới khi bạn đạt được mức độ thành thạo mong muốn hoặc (cho tới khi) đã đầu tư 20 tiếng. Nếu bạn không sẵn sàng đầu tư ít nhất 20 tiếng thì hãy chọn kỹ năng khác để học.

Lý do của việc này rất đơn giản: Giai đoạn đầu của quá trình học kỹ năng thường có cảm giác khó hơn so với độ khó thực sự của chúng. Bạn thường bối rối, và sẽ vấp phải những vấn đề cùng rào cản không mong muốn. Thay vì từ bỏ khi bạn vấp phải khó khăn đơn giản nhất, việc cam kết trước sẽ dành ra 20 tiếng sẽ khiến cho việc kiên định với con đường bạn đã chọn trở nên đơn giản hơn.

Hãy nghĩ về cách tiếp cận này như một bài tập về tính gan góc: Bạn sẽ không để một vấn đề nhỏ xíu, ngớ ngẩn ngăn cản bạn làm việc bạn đã quyết định là mình thực sự muốn làm. Hoặc là bạn sẽ giải quyết vấn đề, hoặc là bạn sẽ làm hết sức cho tới khi đạt mốc 20 tiếng. Đến lúc đó, bạn sẽ ở vào vị trí tốt hơn để quyết định sẽ tiếp tục như thế nào.

## 8. Tạo vòng phản hồi nhanh

“Phản hồi nhanh” nghĩa là nhận thông tin chính xác về cách bạn biểu hiện nhanh nhất có thể. Thời gian nhận phản hồi chính xác càng chậm thì thời gian học kỹ năng càng lâu.

Chẳng hạn, lấy nghệ thuật làm pho mát là ví dụ. Thường phải mất cả tháng hoặc cả năm để hoàn thành quá trình hóa học tinh vi nhằm tạo ra pho mát nguyên chất, và không có cách nào để đẩy nhanh quá trình đó mà không làm

hồng kết quả. Nếu phải mất sáu tháng để xác định pho mát của bạn có ngon hay không, thì sự chậm trễ trong phản hồi khiến cho việc học kỹ năng nhanh trở nên khó khăn hơn.

Theo lẽ tự nhiên, phản hồi nhanh sẽ dẫn tới việc học kỹ năng nhanh. Nếu phản hồi đến ngay lập tức, hoặc chỉ chậm trễ một chút, thì sẽ dễ dàng hơn rất nhiều để kết nối thông tin đó với hành động của bạn và đưa ra những điều chỉnh thích hợp.

Hình thức phản hồi tốt nhất là gần như ngay lập tức. Đó chính là lý do tại sao những kỹ năng như lập trình có thể trở thành chất gây nghiện êm dịu như vậy: Bạn tạo ra một thay đổi nhỏ, và chỉ một phần nghìn giây sau máy tính đã có thể cho bạn biết thay đổi đó có tác dụng hay không. Nếu bạn không thích phản hồi, hãy tạo ra thay đổi mới và thử lại.

Có rất nhiều nguồn phản hồi hữu ích. Như Atul Gawande, bác sỹ phẫu thuật kỳ cựu và một người chơi quần vợt không chuyên, đã giải thích trong một bài báo đăng trên tờ *The New Yorker* thì những huấn luyện viên và người hướng dẫn có kinh nghiệm có thể cho bạn những phản hồi ngay lập tức về cách bạn đang biểu hiện và gợi ý cách điều chỉnh cần thiết.

Huấn luyện viên không phải là nguồn phản hồi nhanh duy nhất. Những thiết bị ghi hình như máy quay phim cũng có thể giúp bạn xem lại hình ảnh của mình khi bạn thực hiện kỹ năng. Những công cụ như chương trình máy tính, thiết bị hỗ trợ đào tạo và các thiết bị khác có thể ngay lập tức chỉ cho bạn biết khi nào bạn mắc sai lầm hoặc có điều gì đó không thích hợp.

Nguồn phản hồi nhanh mà bạn tích hợp vào quá trình luyện tập của bạn càng nhiều thì bạn càng nhanh học được kỹ năng.

## 9. Luyện tập tính giờ

Bộ não của chúng ta được hình thành để học – để chú ý các kiểu mẫu, kích thích những hành động tiềm năng và xác định điều gì có thể xảy ra tiếp theo. Não bộ không phải được hình thành để ước lượng chính xác thời gian – phải mất bao lâu cho việc gì đó hoặc bạn phải dành bao nhiêu thời gian để làm việc gì đó.

Trong giai đoạn đầu của quá trình luyện tập kỹ năng mới, rất dễ để đánh giá quá cao lượng thời gian bạn đã dành ra để luyện tập. Khi bạn không giỏi (và

bạn biết điều đó), thời gian dường như dài lê thê, và có vẻ như bạn đã luyện tập được một quãng thời gian lâu hơn so với thời gian thực tế bạn bỏ ra.

Giải pháp cho việc này là luyện tập tính giờ. Hãy mua một thiết bị đếm ngược tử tế và hẹn thời gian đếm ngược trong 20 phút. Chỉ có một nguyên tắc là một khi bạn bắt đầu tính giờ, bạn cần phải luyện tập cho tới khi hết giờ. Không có ngoại lệ.

Thủ thuật đơn giản này sẽ giúp cho việc hoàn thành những thời kỳ luyện tập dài hơi trở nên dễ dàng hơn, ngay cả khi bạn mệt mỏi hoặc chán nản.

Thời kỳ luyện tập bạn hoàn thành được càng nhiều thì việc học kỹ năng của bạn càng nhanh hơn. Hãy dành thời gian để luyện tập từ ba đến năm lần mỗi ngày, và bạn sẽ thấy bản thân tiến bộ trong một thời gian rất ngắn.

## 10. Nhấn mạnh số lượng và tốc độ

Khi bắt đầu học một kỹ năng mới, bạn sẽ bị thôi thúc phải tập trung vào việc luyện tập thật hoàn hảo – đây chính là công thức tạo ra sự thất vọng. Tất nhiên là biểu hiện của bạn sẽ chẳng thể nào gần tới mức hoàn hảo được.

Thay vì cố gắng trở nên hoàn hảo, hãy tập trung vào việc luyện tập càng nhiều, càng nhanh càng tốt, trong khi vẫn duy trì được mức độ “đủ tốt”.

Trong cuốn *Nghệ thuật và nỗi sợ* (Art & Fear) (2001), hai tác giả David Bayles và Ted Orland đã chia sẻ một giai thoại rất thú vị về giá trị của số lượng:

*Trong ngày khai giảng, thầy giáo dạy làm gốm tuyên bố sẽ chia lớp thành hai nhóm. Ông nói, tất cả những người ở phía bên trái của xưởng làm gốm sẽ được đánh giá căn cứ vào số lượng sản phẩm mà họ làm ra, còn tất cả những người bên phải sẽ được đánh giá dựa trên chất lượng sản phẩm.*

*Phương pháp của ông rất đơn giản: Vào ngày kết thúc khóa học, ông sẽ đem theo chiếc cân và cân thành quả của nhóm “số lượng”: 25 cân đạt điểm A, 20 cân đạt điểm B,... Tuy nhiên, nhóm được đánh giá dựa vào “chất lượng” chỉ cần làm ra được một chiếc bình – một chiếc hoàn hảo – để đạt được điểm A.*

*Đến lúc xếp hạng có một chuyện kỳ lạ đã xảy ra: Những tác phẩm đạt*

*chất lượng cao nhất đều là tác phẩm của nhóm được đánh giá dựa trên “số lượng”. Có vẻ như trong khi nhóm “số lượng” bận bịu hoàn thành khối lượng công việc và học hỏi từ sai lầm của mình, nhóm “chất lượng” chỉ ngồi đó suy ngẫm về sự hoàn hảo, và cuối cùng chẳng có gì để thể hiện sự nỗ lực của họ ngoài mô lý thuyết và một đồng đất sét.*

Kỹ năng là kết quả của việc luyện tập chuyên tâm, bền bỉ, và trong giai đoạn đầu luyện tập, số lượng và tốc độ hoàn toàn áp đảo chất lượng. Bạn luyện tập càng nhanh và thường xuyên thì bạn càng nhanh chóng học được kỹ năng đó.

Nói như vậy không có nghĩa là bạn nên bỏ qua hình thức luyện tập tử tế. Một số kỹ năng, đặc biệt là những kỹ năng đòi hỏi phải hành động hoặc chuyển động cơ thể, thì yêu cầu là phải đạt đến một sự thể hiện tốt nhất. Nếu bạn luyện tập kỹ thuật trong hội họa, bắt chước Jackson Pollack<sup>(1)</sup> với hàng trăm bức vẽ một ngày sẽ chẳng thể giúp gì cho bạn nếu mục tiêu của bạn là vẽ một bức chân dung giống như thật. Kỹ thuật là điều quan trọng.

Trước tiên, cần phải đảm bảo bạn đang luyện tập bằng một hình thức đủ tốt để có thể đạt được mức độ thành thạo mong muốn của mình. Một khi bạn đã luyện tập với dạng thức tốt trong ít nhất 80 đến 90% thời gian, hãy đẩy nhanh tốc độ để học được kỹ năng nhanh hơn.

Đó chính là 10 nguyên tắc đơn giản để đảm bảo bạn sẽ luyện tập kỹ năng quan trọng theo cách hiệu quả và năng suất nhất có thể.

## Có tác dụng không?

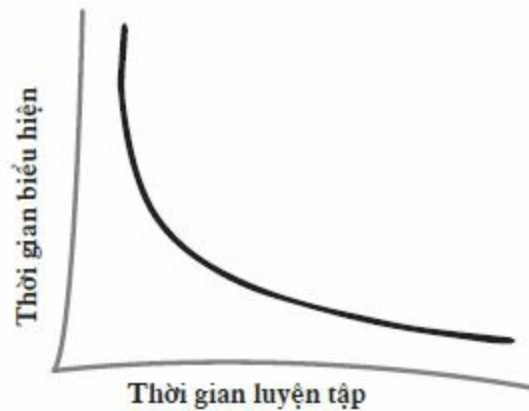
Phương pháp này có thực sự giúp bạn học kỹ năng nhanh hơn không? Nghiên cứu đã chỉ ra là hoàn toàn có thể.

Trong những nghiên cứu mang tính học thuật về việc học kỹ năng dẫn truyền và khởi động, các nhà nghiên cứu đã phát hiện ra một điểm chung: Khi những người tham gia nghiên cứu bắt đầu luyện tập một kỹ năng mới, biểu hiện của họ thường được cải thiện một cách đáng kể trong một khoảng thời gian rất ngắn. Không cần phải luyện tập quá nhiều để đi từ “rất chậm và cực kỳ kém” sang “nhanh vừa phải và giỏi đáng kể”.

Trong văn học, điều này được coi là “luật luyện tập năng lực”, và thường được nhắc đi nhắc lại. Tác dụng của nó được biết đến rộng rãi trong giới nghiên

cứu về học kỹ năng ít nhất là từ năm 1926, và nó đã được sử dụng rất nhiều lần trong các nghiên cứu về cả kỹ năng thể chất và tinh thần. Thậm chí có một nghiên cứu còn cho rằng: “Bất cứ lý thuyết nào về học kỹ năng nếu không phù hợp với chức năng học của luật năng lực thì đều có thể bị loại bỏ ngay lập tức”.

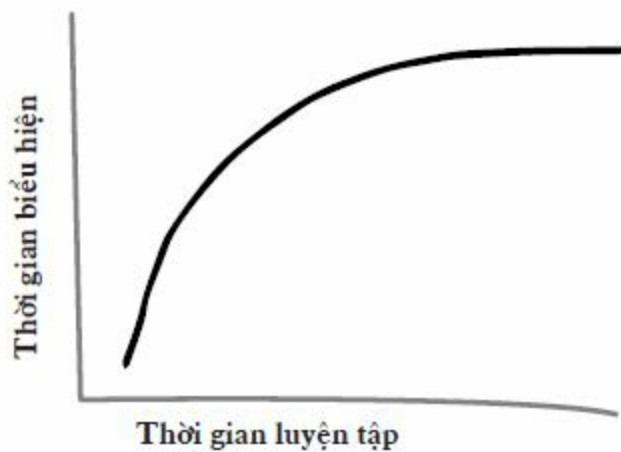
Các nghiên cứu mang tính học thuật đã chỉ ra rằng “luật luyện tập năng lực” có hình vòng cung như thế này, với thời gian biểu hiện trên trục y và kết quả luyện tập trên trục x:



Vì thời gian là một đại lượng tăng, nên đường cong dốc xuống. Bằng cách luyện tập, sẽ tốn ít thời gian hơn để hoàn thành nhiệm vụ đã đề ra.

Thật thú vị khi lưu ý rằng, nếu bạn đặt lại tên cho trục y là “bạn giỏi tới mức nào” (nghĩa là, bạn định nghĩa biểu hiện bằng thuật ngữ chung chung hơn so với một đơn vị thời gian), bạn sẽ có khái niệm được biết đến rộng rãi, đó là *đường cong nhận thức*.

Mô hình chung của đường cong nhận thức trông giống như thế này: Khi bạn bắt đầu, bạn sợ, nhưng bạn sẽ tiến bộ rất nhanh khi bạn học những phần quan trọng nhất của kỹ năng. Sau khi nhanh chóng đạt đến một mức độ nào đó của một kỹ năng, tỉ lệ tiến bộ của bạn giảm, và hệ quả là sự tiến bộ sẽ trở nên chậm hơn nhiều.



Ngược với cách sử dụng thông thường, “đường cong nhận thức có dạng dốc đứng” là tốt chứ không xấu. Biểu đồ giải thích rõ lý do: Đường cong nhận thức có dạng dốc đứng cho thấy tỉ lệ học kỹ năng rất nhanh. Đường cong càng dốc, bạn càng học được nhiều trong mỗi đơn vị thời gian.

Bạn có thể nghĩ về bảng liệt kê những mục cần kiểm tra mà tôi đã phác họa như một cách để làm cho đường cong nhận thức của bạn dốc đứng hơn một cách có chủ đích. Bản thân các nguyên tắc chỉ là những thủ thuật đơn giản, khiến cho hai giai đoạn mang tính lý thuyết đầu tiên của quá trình học kỹ năng (*nhận thức* và *liên tưởng*) trở nên dễ thực hiện hơn.

Một khi bạn bắt đầu luyện tập một điều gì đó mới mẻ, kỹ năng của bạn sẽ tự nhiên được cải thiện một cách đáng kể trong một khoảng thời gian rất ngắn. Vấn đề chỉ là bắt đầu luyện tập nhanh nhất có thể. Không phải là *nghĩ* về việc luyện tập hay *lo lắng* về việc luyện tập, mà là *thực sự* luyện tập.

Thật sự rất dễ để có cảm giác như bạn đang đầu tư rất nhiều thời gian cho một kỹ năng mà không luyện tập tí gì. Nếu bạn muốn học một điều gì đó trong khoảng thời gian dài, bạn mơ về việc sẽ giỏi việc đó, nhưng bạn lại ngần ngại để bắt đầu, bạn có thể mất hàng năm năng lượng tinh thần và tình cảm mà không tiến bộ được chút nào. Nếu bạn không biết bạn đang cố gắng đi đâu hoặc không có chiến lược rõ ràng để đến được đó, bạn có thể làm lãng phí một phần năng lượng.

Mười nguyên tắc này được thiết kế để giúp bạn loại bỏ những gì không hiệu quả và thay thế nó bằng các hoạt động cần thiết cho quá trình học kỹ năng. Thời gian và năng lượng bạn dành cho hai giai đoạn đầu của quá trình học kỹ năng càng nhiều, và thời gian dành cho những việc không giúp gì cho bạn càng ít thì bạn càng nhanh chóng học được kỹ năng đó. Chỉ đơn giản vậy thôi.



## Thế còn mãi mê thì sao?

Đây không phải là cách duy nhất để học kỹ năng mới, nhưng chắc chắn nó là cách linh động nhất. Những phương pháp khác có thể đem đến kết quả tương tự, nhưng lại đòi hỏi phải đánh đổi tương ứng.

Phương pháp phổ biến nhất để học kỹ năng nhanh là *mãi mê*: Thay đổi hoàn toàn môi trường của bạn theo cách tạo hứng khởi cho việc luyện tập cẩn thận và liên tục. Nếu bạn muốn học tiếng Pháp, chẳng hạn như vậy, thì học một cách mãi mê đồng nghĩa với việc sống ở Pháp khoảng vài tuần hoặc một tháng.

Nhìn chung là cách này hiệu quả. Nếu bạn chuyển tới Pháp, bạn sẽ buộc phải rèn luyện kỹ năng nói tại mọi thời điểm khi bạn ở đây. Sau khoảng vài ngày khó chịu để làm quen với môi trường mới xung quanh, bạn sẽ nhận thấy kỹ năng của mình tiến bộ với tốc độ rất nhanh.

Mãi mê có tác dụng vì nó đảm bảo bạn hoàn thành những giờ rèn luyện quan trọng đầu tiên mà không thất bại: Bạn không thể chạy trốn khỏi môi trường của mình, vì thế việc rèn luyện diễn ra hoàn toàn tự động.

Bất lợi của sự mãi mê là nó thường đòi hỏi phải coi kỹ năng là ưu tiên hàng đầu trong một khoảng thời gian dài. Nếu bỏ lại tất cả những việc khác, đóng gói hành lý và chuyển tới Pháp là lựa chọn có thể thực hiện được của bạn thì học tiếng Pháp thông qua sự mãi mê là một chiến lược tốt.

Đáng buồn là, hầu hết chúng ta đều có những cam kết mà chúng ta không thể (hoặc không muốn) bỏ lại: gia đình, công việc, khoản vay thế chấp phải trả... Trong những trường hợp này, phương pháp này khó hoặc không thể thực hiện được.

Trong tình huống tồi tệ nhất, ý tưởng học mãi mê trở thành một rào cản thực sự: Nếu bạn tiếp tục chờ một cơ hội học mãi mê trước khi học một kỹ năng mới, bạn có thể gây lãng phí khoảng thời gian quý báu lên tới vài năm.

Hãy đón nhận cơ hội học mãi mê khi nó tới, nhưng đừng phụ thuộc vào nó. Những thủ thuật này được thiết kế để giúp bạn học được những kỹ năng mới, ngay cả khi bạn chỉ có một hoặc hai giờ rảnh mỗi ngày.

## Kích hoạt lại những kỹ năng cũ

Quan trọng là bạn phải chú ý rằng những nguyên tắc này hữu dụng ngay cả khi kỹ năng mà bạn muốn học không hẳn là kỹ năng mới đối với bạn. Hoàn toàn có thể sử dụng những thủ thuật này để học lại những kỹ năng cũ trong khoảng thời gian kỷ lục.

Chẳng hạn, tôi đã học cách chơi kèn trumpet từ hồi cấp ba, và tôi đã tập chăm chỉ để chơi khá tốt loại kèn này. Nhưng từ khi tốt nghiệp và học đại học, tôi không hề động tới nó.

Nếu tôi quyết định nhắc chiếc kèn trumpet lên, tôi sẽ không phải mất quá nhiều thời gian luyện tập để kích hoạt lại kỹ năng (chơi đàn) của mình. Tôi đã biết những kỹ năng nhỏ cần thiết, vì thế tôi sẽ tập trung vào cách đặt môi (kiểm soát cơ quanh môi trong khi thổi hơi vào miệng kèn), đọc nốt nhạc, nhớ lại vị trí đặt ngón tay tương ứng, và xem lại lý thuyết âm nhạc cơ bản (nhịp, phách, chức năng và biểu hiện).

Sẽ chỉ mất khoảng vài tiếng luyện tập để học lại kỹ năng đó. Việc kích hoạt lại chủ yếu đòi hỏi phải có thời gian, loại bỏ rào cản đối với việc luyện tập và luyện tập tính giờ.

Bắt đầu nghiêm túc là đã làm được một nửa

**Đôi khi bạn sẽ muốn từ bỏ chiếc đàn ghi-ta. Bạn sẽ ghét đàn ghi-ta. Nhưng nếu bạn cứ gắn với nó, bạn sẽ được thưởng xứng đáng.**

**– JIMI HENDRIX, nghệ sĩ chơi ghi-ta điện nổi tiếng**

Bạn không cần sử dụng tất cả những nguyên tắc này cho mỗi kỹ năng mà bạn học, nhưng bạn sẽ luôn thấy ít nhất một vài trong số đó là cần thiết.

Tôi thấy có tác dụng khi nghĩ về những nguyên tắc này như một bảng liệt kê những mục cần kiểm tra. Bất cứ khi nào bạn quyết định học một điều gì đó mới mẻ, chỉ cần xem qua bảng liệt kê những mục cần kiểm tra đó và xác định xem những nguyên tắc nào có thể ứng dụng cho dự án của bạn.

Đây là bảng liệt kê những mục cần kiểm tra để học kỹ năng:

1. Chọn một dự án đáng yêu.

2. Tập trung năng lượng của bạn vào chỉ một kỹ năng.
3. Xác định mức độ thành thạo mà bạn mong muốn.
4. Phân chia kỹ năng thành những kỹ năng nhỏ.
5. Tìm được những công cụ then chốt.
6. Loại bỏ những rào cản để luyện tập.
7. Dành thời gian chuyên tâm luyện tập.
8. Tạo vòng phản hồi nhanh.
9. Luyện tập tính giờ.
0. Nhấn mạnh số lượng và tốc độ.

Chính là vậy đấy. Hãy áp dụng bảng liệt kê những mục cần kiểm tra này cho kỹ năng quan trọng hiện tại của bạn, và việc rèn luyện của bạn sẽ trở nên hiệu quả, năng suất hơn, giúp bạn học kỹ năng nhanh hơn.

Như tôi đã nói, phương pháp này không phải là môn khoa học thần tốc. Nó chỉ đơn giản là kiến thức, chiến lược và sự chuẩn bị thông thường áp dụng cho kỹ năng mà bạn muốn cải thiện. Không hơn, không kém.

Giờ chúng ta hãy cùng kiểm tra xem học và nghiên cứu có thể khiến cho quá trình học kỹ năng của bạn hiệu quả hơn như thế nào.

## Mười nguyên tắc học hiệu quả

---

**Không vấn đề nào có thể chống lại được cuộc đổ bộ của việc suy nghĩ liên tục.**

**–VOLTAIRE**

...

Như chúng ta đã thảo luận ở chương 1, học không giống với việc tiếp thu kỹ năng. Tuy nhiên, điều đó không có nghĩa là học không quan trọng. Tiến hành nghiên cứu một chút trước khi bạn lao vào luyện tập có thể tiết kiệm thời gian, năng lượng và sự dững cảm chịu đựng của bạn.

Học giúp việc luyện tập của bạn hiệu quả hơn – việc này giúp bạn dành nhiều thời gian luyện tập hơn cho những kỹ năng nhỏ quan trọng nhất trước tiên.

Với tinh thần đó, dưới đây là 10 nguyên tắc quan trọng của việc học hiệu quả:

1. Nghiên cứu kỹ năng và các chủ đề liên quan.
2. Khai mở tâm trí.
3. Xác định hình mẫu tinh thần và móc nối tinh thần.
4. Tưởng tượng điều ngược lại với điều bạn muốn.
5. Nói chuyện với người đang thực hành kỹ năng để đặt ra những triển vọng.
6. Loại bỏ những điều gây xao lãng trong môi trường của bạn.
7. Sử dụng sự củng cố và lặp lại cách quãng để ghi nhớ.
8. Tạo lịch trình và bảng liệt kê những mục cần kiểm tra.
9. Dự đoán và kiểm nghiệm dự đoán.

o. Trân trọng hệ thống sinh học của bạn.

## 1. Nghiên cứu kỹ năng và các chủ đề liên quan

Hãy dành ra 20 phút tìm kiếm trang web, ghé vào cửa hàng sách hoặc lướt qua các chồng sách trong thư viện địa phương để tìm sách và các nguồn tài liệu liên quan tới kỹ năng bạn muốn học. Mục tiêu là xác định được ít nhất ba cuốn sách, đĩa DVD hướng dẫn, tài liệu hoặc nguồn tham khảo khác có liên quan tới kỹ năng bạn muốn học.

Trước khi bạn hoảng sợ, hãy nhớ là bạn không cần phải dành hàng giờ để ghi nhớ những tài liệu đó. Ngược lại, thời gian dành để đọc hoặc xem không phải là thời gian dành cho việc luyện tập.

Không phải là bạn đang nhồi nhét kiến thức để tham gia một kỳ thi đâu. Mục đích của việc nghiên cứu ban đầu này là để xác định những kỹ năng nhỏ quan trọng nhất, những yếu tố thiết yếu và những công cụ cần thiết cho việc luyện tập nhanh nhất có thể. Những điều bạn biết trước về kỹ năng đó càng nhiều thì sự chuẩn bị của bạn càng thông minh hơn. Mục tiêu là để thu thập được lượng kiến thức sâu rộng về kỹ năng đó nhanh nhất có thể, tạo ra một cái nhìn tổng quan về quá trình học kỹ năng đó.

Đối với việc học kỹ năng nhanh, đọc lướt tốt hơn là đọc kỹ. Bằng cách chú ý tới những ý tưởng và công cụ xuất hiện đi xuất hiện lại trong các ngữ cảnh khác nhau, bạn có thể tin tưởng độ chính xác của những mô hình mà bạn chú ý và chuẩn bị việc luyện tập tương ứng.

Nếu bạn muốn nướng được chiếc bánh sừng bò hoàn hảo, hãy đọc một vài cuốn sách liên quan tới việc nướng bánh và bột bánh. Thay vì sáng tạo lại quá trình, bạn sẽ tìm thấy những kỹ thuật đã có sẵn – những kỹ thuật đã được hoàn thiện sau nhiều năm bởi những bậc thầy trong lĩnh vực này. Nếu bạn thấy những kỹ thuật hoặc quá trình giống nhau được miêu tả trong nhiều nguồn khác nhau, thì chắc chắn đó là những điều quan trọng cần phải biết.

Một khi bạn tìm thấy điều gì đó có vẻ là kỹ thuật hữu ích nhất, bạn có thể thử nghiệm nó trong chính căn bếp của bạn, tiết kiệm cho bạn hàng loạt lần thử và sai.

## 2. Khai mở tâm trí

Một số nghiên cứu ban đầu của bạn có chứa các khái niệm, kỹ thuật và ý tưởng mà bạn không hiểu. Thường thì sẽ có một số thứ có vẻ đặc biệt quan trọng, nhưng bạn lại không hiểu nó có nghĩa gì. Bạn sẽ đọc những từ mà bạn không đoán ra nghĩa, và thấy người dạy kỹ năng làm những việc mà bạn không thể hiểu được.

Đừng hốt hoảng. Sự mơ hồ ban đầu của bạn là điều hoàn toàn bình thường. Thực ra, đó còn là điều tuyệt vời. Hãy bỏ qua sự mơ hồ để tiến về phía trước.

Nghiên cứu ban đầu là một trong những cách tốt nhất để xác định các kỹ năng nhỏ và ý tưởng quan trọng, nhưng rất có khả năng là bạn sẽ không biết chúng có nghĩa là gì. Ý nghĩa sẽ xuất hiện sau, khi bạn đã bắt đầu luyện tập.

Tiến sỹ Stephen Krashen, chuyên gia ngôn ngữ mà tôi đã đề cập đến lúc trước, gọi đây là *đầu vào vừa mức*. Dĩ nhiên, thông tin mới mà bạn tiếp nhận không phải là dễ hiểu lắm vì nó không liên quan tới bất cứ thứ gì bạn biết hoặc đã có kinh nghiệm. Theo thời gian, chính thông tin đó sẽ trở nên dễ hiểu hơn khi bạn đã có một vài kinh nghiệm dặt lưng làm vốn. Nói như giảng sư dạy yoga danh tiếng T. K. V. Desikachar: “Nhận biết được sự mơ hồ thực ra là một dạng thức của sự rõ ràng”.

Nhận biết được bản thân đang bối rối là một việc rất có giá trị. Việc đó có thể giúp bạn định nghĩa chính xác điều bạn thấy bối rối, mơ hồ, mà việc này lại giúp bạn xác định được bạn cần phải nghiên cứu hoặc phải làm gì tiếp theo để giải quyết sự bối rối, mơ hồ đó.

Nếu bạn không bối rối với ít nhất là một nửa nội dung nghiên cứu ban đầu của bạn, thì có nghĩa là bạn đang học không nhanh như khả năng có thể của bạn. Nếu bạn bắt đầu cảm thấy sợ hãi hoặc do dự về tốc độ hiện tại của bạn thì có nghĩa là bạn đang đi đúng đường. Giả dụ bạn đang làm một dự án đáng yêu, lúc đầu bạn càng bối rối, mơ hồ thì áp lực bên trong buộc bạn phải làm sáng tỏ mọi việc càng lớn, và bạn sẽ học càng nhanh.

Không sẵn sàng khai mở tâm trí là rào cản cảm xúc lớn nhất đối với việc tiếp thu kỹ năng nhanh. Cảm thấy mình ngốc chẳng có gì thú vị cả, nhưng tự nhắc nhở bản thân rằng luyện tập có thể giúp bạn chuyển từ bối rối, mơ hồ sang rõ ràng nhanh nhất có thể.

### 3. Xác định hình mẫu tinh thần và móc nối tinh thần

Khi tiến hành nghiên cứu, thường thì bạn sẽ bắt đầu chú ý tới các hình mẫu: Các ý tưởng và kỹ thuật lặp đi lặp lại nhiều lần.

Những khái niệm này được gọi là *hình mẫu tinh thần*, và chúng thực sự rất quan trọng. Hình mẫu tinh thần là đơn vị cơ bản của việc học – đó là cách để hiểu và gọi tên được một sự vật hoặc một mối quan hệ tồn tại trên thế giới này. Khi bạn thu thập được các hình mẫu tinh thần chính xác, sẽ dễ dàng hơn để đoán được điều gì sẽ xảy ra khi bạn thực hiện một hành động cụ thể. Hình mẫu tinh thần giúp bạn dễ thảo luận với người khác về kinh nghiệm của bản thân hơn.

Và đây là ví dụ chứng minh: Gần đây tôi có giúp bố tôi lập một trang web. Khi thực hiện, tôi đã cố gắng giải thích tôi đang làm gì. Lúc đầu, việc đó khiến cả hai bố con tôi đều mệt mỏi. Tôi thường xuyên sử dụng những từ như “máy chủ” và bố tôi hoàn toàn không hiểu tôi đang nói về điều gì.

Khi đã hiểu được máy chủ là một chiếc máy tính đặc biệt, truyền tải trang web tới người yêu cầu, và máy chủ là chiếc máy tính *khác* với chiếc máy tính mà chúng tôi đang sử dụng, thì bố tôi thấy dễ hiểu việc bố con tôi đang làm hơn. Trong trường hợp này, máy chủ là một hình mẫu tinh thần – một khi bạn đã quen với thuật ngữ này, việc hiểu được quá trình lập một trang web sẽ dễ dàng hơn rất nhiều.

Bạn cũng sẽ chú ý tới một vài thứ có vẻ giống với thứ gì đó mà bạn đã thân thuộc. Đó là những *móc nối tinh thần*: những phép so sánh, ẩn dụ bạn có thể sử dụng để nhớ các khái niệm mới.

Trong trường hợp máy chủ web, hãy tưởng tượng đến một người thủ thư. Khi bạn tới thư viện và hỏi mượn một cuốn sách, người thủ thư sẽ tìm trên các kệ sách với hàng trăm hoặc hàng ngàn cuốn sách để tìm đúng cuốn mà bạn cần. Khi người thủ thư tìm thấy cuốn sách, họ sẽ mang nó đến cho bạn. Nếu không tìm được cuốn sách đó, người thủ thư sẽ nói với bạn là: “Tôi không thể tìm được cuốn sách mà bạn yêu cầu”.

Đó chính xác là cách mà máy chủ web hoạt động. Khi bạn yêu cầu một trang web cụ thể nào đó, máy chủ sẽ tìm kiếm trang web đó trong bộ nhớ. Nếu tìm thấy, nó sẽ mang đến cho bạn. Nếu máy chủ không tìm thấy trang web đó, nó sẽ gửi cho bạn một tin nhắn “Lỗi 404: Trang web không tìm được”. Nghĩ về

phần mềm máy chủ như một “thủ thư máy tính” là cách hữu hiệu để nghĩ về cách vận hành của hệ thống.

Nếu bạn có thể xác định được hình mẫu tinh thần và móc nối tinh thần trong nghiên cứu ban đầu của bạn càng nhiều thì bạn càng dễ sử dụng chúng trong quá trình luyện tập.

#### 4. Tưởng tượng điều ngược lại với điều bạn muốn

Có một cách khác thường để có được hiểu biết về một kỹ năng mới, đó là suy ngẫm về thảm họa thay vì sự hoàn hảo.

Sẽ thế nào nếu bạn làm sai mọi việc? Sẽ thế nào nếu bạn nhận được kết quả tệ nhất?

Đây là phương pháp giải quyết vấn đề có tên là *phép nghịch đảo*, và nó rất hữu dụng trong việc học những điều cần thiết. Bằng cách nghiên cứu điều ngược lại của điều bạn muốn, bạn có thể xác định được những yếu tố quan trọng nhưng không thể nhìn thấy ngay lập tức.

Lấy việc chèo thuyền kayak vượt thác làm ví dụ. Tôi cần biết điều gì nếu tôi muốn chèo thuyền kayak trên một con sông lớn có nước chảy xiết, đá lờm chờm?

Và đây là phép nghịch đảo: Sẽ thế nào nếu có chuyện không hay xảy ra?

• Tôi sẽ bị lật úp xuống dòng nước và không thể ngoi lên được.

• Tôi sẽ làm ngập chiếc thuyền kayak của mình, khiến nó bị chìm hoặc ngập, kết quả là mất luôn chiếc thuyền.

• Tôi làm mất mái chèo.

• Tôi bị đập đầu vào đá.

• Tôi bị hất văng khỏi thuyền, mắc kẹt trong xoáy nước (một điểm trên sông mà nước tạo thành một vòng xoáy như máy giặt) và không thể thoát ra ngoài được.



Nếu tôi cố gắng để làm tất cả những việc đó cùng một lúc khi ở giữa dòng nước dữ, có lẽ tôi sẽ mất mạng – viễn cảnh tồi tệ nhất có thể xảy ra.

Suy nghĩ đáng sợ này rất có ích vì nó chỉ ra một số kỹ năng rất quan trọng của việc chèo thuyền kayak:

- 1. Học cách lật lại thuyền khi bị lật mà không bị hất văng ra ngoài.
- 2. Học cách ngăn không cho nước tràn vào thuyền kayak, tát nước nếu cần thiết.
- 3. Học cách để không bị mất mái chèo trong dòng nước xiết.
- 4. Học và có sự phòng ngừa an toàn khi thả thuyền trôi quanh những tảng đá lớn.
- 5. Thăm dò dòng sông trước khi chèo thuyền để tránh hoàn toàn những điểm nguy hiểm.

Sự mô phỏng tinh thần này cũng giúp tôi lập được một danh sách những thứ cần mua: tôi cần phải đầu tư một chiếc áo phao, mũ bảo hiểm và những dụng cụ an toàn khác.

Lúc này, thay vì (1) vượt sông (2) vui vẻ (3) không thiệt mạng, tôi đã có một danh sách những kỹ năng nhỏ để luyện tập và (một danh sách) những hành động cần phải thực hiện để đảm bảo tôi thực sự sẽ được vui, giữ được đồ đạc và sống sót được sau cuộc hành trình.

Phép nghịch đảo đã phát huy tác dụng.

## 5. Nói chuyện với người đang thực hành kỹ năng để đặt ra những triển vọng

Học trong giai đoạn đầu giúp bạn đặt ra những kỳ vọng thích hợp: Thế nào là biểu hiện phù hợp đối với người mới bắt đầu?

Khi bạn bắt đầu học một kỹ năng mới, thường thì bạn sẽ đánh giá thấp tính phức tạp của nhiệm vụ đó, hoặc đánh giá thấp số lượng các thành tố liên quan

cần phải hoàn thành để thực hiện tốt kỹ năng đó. Nếu kỹ năng có liên quan tới khả năng tạo dựng uy tín xã hội thì sự thần bí đi kèm với nó cũng có thể che mờ những kỳ vọng ban đầu.

Rất nhiều người muốn trở thành ngôi sao nhạc rock nhấc cây đàn ghi-ta điện lên chỉ để thấy thật khó để có thể vừa chơi giỏi, vừa hát đúng nhạc, vừa trông thật hoành tráng cùng một lúc được. Một phần của vấn đề nằm ở chỗ “trở thành một ngôi sao nhạc rock” không phải là một kỹ năng đơn lẻ. Đó là một tập hợp những kỹ năng nhỏ có liên quan, mà mỗi kỹ năng lại đòi hỏi phải chuyên tâm luyện tập mới tiến bộ được.

Nói chuyện với những người đã đạt được kỹ năng đó trước bạn giúp xóa tan những hiểu lầm và làm tương trước khi bạn đầu tư thời gian và công sức vào đó. Bằng cách biết có thể trông chờ vào điều gì khi thực hiện, bạn sẽ thấy dễ duy trì được mối quan tâm của bản thân với việc luyện tập hơn, và tránh được cảm giác thất vọng, chán nản trong giai đoạn đầu của quá trình.

## 6. Loại bỏ những điều gây xao lãng trong môi trường của bạn

Những điều gây xao lãng là kẻ thù số một của quá trình học kỹ năng nhanh. Chúng giết chết sự rèn luyện chuyên tâm, mà thiếu sự rèn luyện chuyên tâm thì sẽ dẫn tới việc học kỹ năng chậm (hoặc không học). Bạn có thể chặn trước việc này bằng cách dành ra ít phút để dự báo và loại bỏ (hoặc giảm bớt) càng nhiều điều gây xao lãng càng tốt trước khi bạn bắt đầu luyện tập.

Những điều gây xao lãng quan trọng nhất xuất phát từ hai nguồn chính là điện tử và sinh vật.

Ti vi, điện thoại và internet là những thiết bị điện tử gây xao lãng. Hãy tắt chúng đi, rút phích nguồn, chặn chúng lại hoặc nhấc chúng ra khỏi môi trường của bạn trong lúc bạn luyện tập, trừ khi chúng thực sự cần thiết cho việc luyện tập.

Người nhà, đồng nghiệp và thú cưng là những đối tượng gây xao lãng. Bạn không thể ngăn cản người khác, nhưng có thể báo cho họ biết trước là bạn sẽ không giúp được gì trong lúc luyện tập. Điều này làm tăng khả năng họ sẽ tôn trọng thời gian luyện tập của bạn mà không làm phiền bạn.

Càng có ít yếu tố gây xao lãng trong lúc bạn luyện tập bao nhiêu thì bạn càng nhanh chóng học được kỹ năng bấy nhiêu.

## 7. Sử dụng sự củng cố và lặp lại cách quãng để ghi nhớ

Để có thể sử dụng tài liệu mà bạn đã học được trong lúc luyện tập, bạn cần phải có khả năng nhớ nhanh lại những ý tưởng có liên quan. Nhiều kỹ năng đòi hỏi phải có khả năng ghi nhớ, ít nhất là ở một mức độ nào đó.

Vấn đề nằm ở đây: Trí nhớ của bạn không hoàn hảo. Bất cứ khi nào bạn học được điều gì mới, bạn có thể sẽ quên nó trừ khi bạn xem lại khái niệm đó trong một khoảng thời gian nhất định nào đó. Quá trình lặp lại này giúp củng cố ý tưởng, và giúp não bạn dung nạp khái niệm vào bộ nhớ dài hạn.

Các nhà nghiên cứu đã phát hiện ra rằng trí nhớ của chúng ta đi theo *đường cong phân rã*: Các khái niệm mới cần được củng cố thường xuyên, nhưng thời gian bạn biết khái niệm càng lâu thì bạn không cần thường xuyên xem lại khái niệm đó mới có thể nhớ lại chính xác được.

*Củng cố và lặp lại cách quãng* là kỹ thuật ghi nhớ giúp bạn xem lại những khái niệm và thông tin quan trọng một cách hệ thống. Những ý tưởng khó nhớ được xem lại thường xuyên, còn những khái niệm dễ hơn và cũ hơn được xem lại ít hơn.

Những chương trình phần mềm thẻ flash như Anki, SuperMemo và Smart giúp việc củng cố và lặp lại cách quãng trở nên rất đơn giản. Hệ thống lặp lại cách quãng phụ thuộc vào mô hình tổng kết “thẻ flash”, và bạn cần tạo ra những tấm thẻ flash của riêng mình. Bằng cách tạo ra những tấm thẻ flash trong lúc chia nhỏ kỹ năng, bạn đã bán một mũi tên trúng hai đích.

Khi bạn đã tạo ra được những tấm thẻ flash, sẽ chỉ mất vài phút mỗi ngày để xem lại chúng. Bằng cách hệ thống hóa quá trình tổng kết và theo dõi khả năng nhớ lại, những hệ thống này có thể giúp bạn học những ý tưởng, kỹ thuật mới trong khoảng thời gian kỷ lục. Nếu bạn liên tục xem lại những bộ thẻ nhớ thì bạn sẽ nhớ được những khái niệm và ý tưởng cần thiết rất nhanh.

Quan trọng là bạn phải nhớ rằng học kỹ năng không chỉ dừng lại ở việc học theo nghĩa học thuật. Nếu bạn quan tâm tới việc ghi nhớ các khái niệm, ý tưởng hoặc từ vựng chỉ để vượt qua một kỳ thi, bạn không cần gì hơn ngoài việc lặp lại cách quãng.

Kỹ thuật này phát huy tác dụng tốt nhất trong những trường hợp cần phải nhớ lại thông tin. Nếu bạn đang học những từ vựng thông dụng để biết thêm một ngoại ngữ mới thì củng cố và lặp lại cách quãng là biện pháp cực kỳ hữu

ích. Trong những trường hợp không cần phải nhớ lại thông tin, tốt hơn hết là bạn nên bỏ qua những tấm thẻ flash để tối đa hóa thời gian luyện tập và thử nghiệm.

## 8. Tạo lịch trình và bảng liệt kê những mục cần kiểm tra

Nhiều kỹ năng cùng có chung một số đặc điểm: hình thành, chuẩn bị, duy trì... Tạo ra một hệ thống đơn giản là cách tốt nhất để đảm bảo những yếu tố quan trọng này mà tốn ít công sức nhất có thể.

*Bảng liệt kê những mục cần kiểm tra* có ích cho việc ghi nhớ những việc cần phải thực hiện mỗi khi bạn luyện tập. Đó là cách để hệ thống hóa quá trình, giải phóng sự chú ý của bạn để tập trung vào những vấn đề quan trọng hơn.

*Lịch trình* là cấu trúc đảm bảo bạn tiếp cận kỹ năng theo một cách thống nhất. Hãy nghĩ tới cầu thủ bóng rổ – người đã xây dựng thói quen ném tự do từ trước. Lau tay lên quần, thả lỏng vai, bắt bóng từ trọng tài, đập bóng ba lần, dừng khoảng ba giây và ném. Đó chính là lịch trình.

Tạo lịch trình và bảng liệt kê những mục cần kiểm tra giúp bạn luyện tập hiệu quả hơn. Hai việc đó cũng giúp việc luyện tập của bạn dễ hình dung hơn, qua đó giúp bạn tận dụng lợi thế của việc nhắc lại về mặt tinh thần, mà điều này có thể giúp ích cho việc rèn luyện về mặt thể chất.

## 9. Dự đoán và kiểm nghiệm dự đoán

Một phần của quá trình học kỹ năng là thử nghiệm: Thử những việc mới để xem chúng có hiệu quả không.

Sự kiểm nghiệm đích thực của việc học hiệu quả không phải là dự đoán. Dựa trên điều bạn biết, liệu bạn có thể đoán được một thay đổi hoặc một thí nghiệm sẽ thành ra thế nào trước khi bạn thực hiện nó không?

Hình thành thói quen đưa ra dự đoán và kiểm nghiệm dự đoán sẽ giúp bạn học kỹ năng nhanh hơn. Đó là biến thể của một phương pháp khoa học có bốn yếu tố chính:

**Quan sát** – gần đây bạn đang quan sát điều gì?

**Hiểu biết** – bạn đã biết gì về chủ đề đó?

**Giả thuyết** – bạn nghĩ điều gì sẽ cải thiện biểu hiện của bạn?

**Kiểm nghiệm** – bạn sẽ thử làm gì tiếp theo?

Tôi khuyến khích bạn sử dụng một cuốn sổ ghi chép hay một công cụ nào đó để theo dõi những thử nghiệm của bạn, và hình thành giả thuyết khi bạn luyện tập. Bằng cách ghi chép những dự đoán của bạn và tạo ra các ý tưởng mới, bạn sẽ có thêm nhiều thí nghiệm để kiểm nghiệm.

## 10. Trân trọng hệ thống sinh học của bạn

Bộ não và cơ thể bạn là những hệ thống sinh học có các nhu cầu sinh lý như cần nước, thức ăn, luyện tập, nghỉ ngơi và ngủ. Thật sự rất dễ để chèn ép, khát khe với bản thân – điều gây phản tác dụng. Nếu không có đầu vào thích hợp, cơ thể và tâm trí bạn không thể cho ra những kết quả hữu ích được.

Theo Tony Schwartz, tác giả cuốn *Sức mạnh của việc toàn tâm toàn ý* (The Power of Full Engagement) (2004) và *Để giỏi bất cứ việc gì* (Be Excellent at Anything) (2011), chu trình học tối ưu là khoảng 90 phút chuyên tâm tập trung. Nếu nhiều hơn, cơ thể và tâm trí bạn sẽ cần được nghỉ ngơi. Hãy sử dụng cơ hội đó để luyện tập, nghỉ ngơi, ăn cơm, chợp mắt hoặc làm việc khác.

Nguyên tắc này rất hợp với nguyên tắc luyện tập tính giờ. Bằng cách đặt đồng hồ hẹn giờ từ 60 đến 90 phút trước khi bạn bắt đầu luyện tập hoặc nghiên cứu, bạn sẽ dễ nhớ đến việc nghỉ ngơi khi đã xong việc.

Bạn cũng có thể chia việc luyện tập của mình thành nhiều phần nhỏ hơn, có thể nghỉ nhanh ở giữa nếu cần: 20 phút tập, 10 phút nghỉ, 20 phút tập, 10 phút nghỉ,...

Gian lận

**Quy tắc ở mọi nơi. Chúng ta đều đang cố gắng hoàn thành việc gì đó.**

## **-THOMAS EDISON, nhà phát minh**

Bạn không cần sử dụng tất cả những nguyên tắc này cho các kỹ năng mà bạn học, nhưng bạn sẽ thấy ít nhất vài nguyên tắc trong số đó là cần thiết.

Tôi thấy hữu dụng nhất là cho rằng những nguyên tắc này giống như một bảng liệt kê những mục cần kiểm tra thứ hai. Bất cứ khi nào bạn quyết định học một kỹ năng mới, chỉ cần xem lại bảng liệt kê những mục cần kiểm tra này và quyết định nguyên tắc nào có thể ứng dụng được cho dự án của bạn.

Đây là bảng liệt kê những mục cần kiểm tra để học hiệu quả:

- | Nghiên cứu kỹ năng và các chủ đề liên quan.
- | Khai mở tâm trí.
- | Xác định hình mẫu tinh thần và móc nối tinh thần.
- | Tưởng tượng điều ngược lại với điều bạn muốn.
- | Nói chuyện với người đang thực hành kỹ năng xác định được có thể kỳ vọng vào điều gì.
- | Loại bỏ những điều gây xao lãng.
- | Sử dụng sự củng cố và lặp lại cách quăng để ghi nhớ.
- | Tạo lịch trình và bảng liệt kê những mục cần kiểm tra.
- | Dự đoán và kiểm nghiệm dự đoán.
- | Trân trọng hệ thống sinh học của bạn.

Vậy đấy. Hãy áp dụng bảng liệt kê những mục cần kiểm tra này cho kỹ năng quan trọng hiện tại của bạn, và bạn sẽ học được là bạn cần phải biết điều gì để luyện tập hiệu quả và năng suất.

# Đưa lý thuyết vào thực hành

Thật sáo rỗng làm sao khi ngồi xuống để viết khi chưa từng đứng lên để sống.

–HENRY DAVID THOREAU

Đã đủ lý thuyết rồi, đến giờ thực hành thôi.

Chúng ta đã lướt qua những điều cơ bản của việc học kỹ năng nhanh, nhưng biết cách làm những việc này không quan trọng bằng việc thực sự bắt tay vào thực hiện chúng. Hãy nhớ: Không luyện tập thì không học được kỹ năng.

Thay vì nhắc đi nhắc lại về lý thuyết học kỹ năng, tôi sẽ chỉ cho bạn cách làm thực sự. Tôi sẽ sử dụng những nguyên tắc này để học một vài kỹ năng mới, và bạn sẽ được ngồi hàng ghế đầu để chứng kiến.

Dưới đây là những kỹ năng tôi dự định học:

Yoga: phát triển một bài tập tư thế (asana) ở nhà

Lập trình: tạo một ứng dụng web chức năng

Đánh máy: học lại cách đánh máy không cần nhìn vào bàn phím với cách bố trí bàn phím không theo tiêu chuẩn

Chiến lược: chơi cờ vây, trò chơi trên bàn cờ phức tạp và cổ nhất trên thế giới

Lướt ván: lướt và di chuyển trên mặt nước lặng

Tôi không có kinh nghiệm về bất kỳ kỹ năng nào trong số những kỹ năng trên. Sử dụng những kỹ thuật và phương pháp tôi vừa mới miêu tả, mục tiêu của tôi là học được mỗi kỹ năng trong số đó trong vòng 30 ngày hoặc ít hơn. Thời gian dự tính để học một kỹ năng đó là khoảng 20 giờ, trung bình 60 đến 90 phút luyện tập mỗi ngày.

## Về những ví dụ này

Những kỹ năng này hoàn toàn mang phong cách riêng. Đó đều là những việc

tôi có hứng thú học vì một vài lý do mà tôi sẽ giải thích cụ thể sau. Những kỹ năng bạn muốn học có thể hoàn toàn khác, nhưng quá trình học kỹ năng cơ bản sẽ giống nhau.

Tôi hi vọng bằng cách chỉ cho bạn thấy cách tôi đã sử dụng quá trình này để học được nhiều kỹ năng khác nhau trong các lĩnh vực khác nhau, bạn sẽ thực sự hiểu được cách sử dụng những thủ thuật này để trở nên thành thạo hơn những kỹ năng quan trọng với bạn.

Nếu bạn quan tâm tới một hoặc hai ví dụ này hơn những ví dụ khác thì cũng điều là bình thường. Hãy đọc những chương đó trước. Nếu bạn bắt đầu đọc một chương và nhận thấy kỹ năng đó chán ngắt hoặc không phù hợp với tình huống của bạn, hãy bỏ qua chương đó. Tôi chỉ sử dụng một phương pháp cốt lõi cho tất cả các ví dụ này, vì thế bạn không cần lo là sẽ bỏ mất phần quan trọng nào trong cách tiếp cận của tôi.

Tôi viết những chương này dưới dạng hướng dẫn, với giả định rằng bạn chưa bao giờ nhìn thấy các kỹ năng đó trước đây, vì thế bạn không có bất cứ kinh nghiệm hoặc hiểu biết nào từ trước. Ngoài việc minh họa cho phương pháp, tôi hi vọng bạn cũng có thể thu lượm được chút ít kiến thức quý giá về năm kỹ năng thú vị rất đáng để luyện tập này.

Nếu bạn có nhiều kinh nghiệm ở một trong năm bộ môn này, nhiều khả năng là bạn sẽ nhận ra một thiếu sót nào đó trong vốn hiểu biết của tôi, một sai lầm nào đó trong cách diễn đạt dài dòng của tôi hoặc có sự bất đồng nào đó với phương pháp của tôi. Điều đó hoàn toàn bình thường. Hãy nhớ rằng: *Tôi đang bắt đầu như một người hoàn toàn mới, và tôi không biết tôi đang làm gì.* (Chưa biết thì đúng hơn.)

Những gì bạn sẽ đọc là tổng quan về quá trình học mỗi một kỹ năng đó của tôi. Tôi đã rất cố gắng để đảm bảo những thông tin trong cuốn sách này đều chính xác và hoàn thiện, nhưng chắc chắn vẫn có sai sót. Trong tất cả các trường hợp, *phương pháp* nghiên cứu và luyện tập có mục đích, có chủ ý là điều quan trọng nhất.

Trước tiên là yoga.



# 4

## YOGA

---

### BÀI HỌC: ĐỪNG LÀM MỌI CHUYỆN KHÓ HƠN MỨC CẦN THIẾT

**Tôi không đánh giá tiến bộ của tôi trong việc học yoga qua việc tôi có thể gặp hoặc vận người như thế nào, mà thông qua việc tôi đối xử với vợ con như thế nào.**

**–T. K. V. DESIKACHAR, Giảng sư dạy yoga danh tiếng**

...

Tôi đang già đi.

Để cho dễ hiểu: Nói một cách tương đối thì tôi vẫn là một người trẻ, nhưng tôi bắt đầu để ý một vài điều về cơ thể khiến tôi lo lắng.

Khi thức dậy vào buổi sáng, tôi thường thấy đau ê ẩm ở phần giữa lưng. Sau một ngày dài ngồi trước máy tính, cổ và vai tôi căng cứng và nhức mỏi. Đó hoàn toàn không phải là cảm giác dễ chịu.

Tính tới thời điểm này, tôi vẫn chưa nghĩ nhiều về cơ thể của mình. Nó chỉ biết hoạt động thôi. Sau khi học hết trung học, tôi không còn chơi các môn thể thao nữa. Khi học cao đẳng, tôi bắt đầu đối xử với cơ thể mình như một loại phương tiện mà mục đích duy nhất của nó là giúp tôi vận chuyển bộ não của mình từ lớp học này tới lớp học khác. Từ khi tốt nghiệp, điều đó vẫn không thay đổi, và tới tận thời điểm này tôi vẫn cảm thấy ổn. Lúc này, những năm tháng bỏ bê cơ thể đã “đuôi kịp” tôi.

Tôi già và tôi biết điều đó.

“Anh nên thực sự tìm hiểu yoga...”

Kelsey, vợ tôi, đã tập yoga từ khi học cao đẳng. Ngày nào cô ấy cũng đi bộ xuống phố để tới phòng tập và dành ít nhất một tiếng ở đó để vận vẹo người với

những tư thế khác nhau trong một căn phòng rất ấm. Cô ấy thích việc đó, và luôn về nhà với vẻ mặt hạnh phúc, thư thái.

Tuần nào Kelsey cũng nói với tôi: “Anh cũng nên thử yoga đi. Tuyệt lắm đấy. Rồi anh sẽ thích nó cho xem”.

Tôi đã do dự. Yoga với tôi mà nói không phải thứ có sức hút.

Vấn đề không phải là việc kéo căng người. Tôi đã tham gia chạy ở trường cấp ba, thế nên kéo căng không phải là vấn đề lớn. Việc yêu thích của tôi là chạy vượt rào 110m. Việc này đòi hỏi phải rất linh hoạt.

Luyện tập chạy vượt rào liên quan rất nhiều tới việc kéo căng, và tôi vẫn duy trì được sự linh hoạt mà tôi đã phát triển trong suốt những năm này, đặc biệt là ở gân khoeo. Thậm chí bây giờ, đã qua hơn một thập kỷ, tôi vẫn có thể chạm lòng bàn tay xuống đất mà không cần phải gập đầu gối.

Nói thẳng ra, lý do tôi chần chừ không tập yoga là vì môn đó có vẻ kỳ cục thế nào ấy.

## “Thư giãn gương mặt”

Cuối cùng, Kelsey cũng thuyết phục được tôi thử tập yoga, coi đó như một quyết tâm của năm mới. Tôi làm việc tại nhà, nên cô ấy vin vào đó để bảo tôi: Học yoga là cách tốt để ra khỏi nhà và gặp gỡ mọi người. Tôi vẫn còn hoài nghi, nhưng vẫn quyết định làm vừa lòng cô ấy.

Chúng tôi đăng ký trở thành thành viên ở Pure Yoga, phòng tập mới toanh ở phía đông New York. Nhiều lớp học trong số đó do Marco Rojas, giáo viên yêu thích nhất của vợ tôi và cũng là một trong những người hướng dẫn hàng đầu của thành phố, giảng dạy.

Hãy tưởng tượng một phiên bản trẻ hơn kiểu Antonio Banderas trong chiếc quần yoga và bạn sẽ có Marco. Anh ta đẹp trai, lôi cuốn và có chất giọng Tây Ban Nha cuốn hút. Các lớp của Marco đều chật cứng, vì anh ta là một thầy giáo tuyệt vời, và còn vì 99% những người theo học yoga ở thành phố này đều có cảm tình với anh ta. (Và ai có thể trách họ được?)

Vì là một người mới nên tôi đã cố gắng hết sức để có thể theo được lớp học của Marco. Tôi không biết về các tư thế, cũng không biết mỗi tư thế được gọi là

gì, vì thế tôi chỉ quan sát những người khác đang làm gì và cố gắng bắt chước. Điều đó đòi hỏi phải rất tập trung. Tôi có thể cố gắng, nhưng đó thực sự là một cuộc vật lộn. Một số tư thế như trồng cây chuối quả thật nằm ngoài khả năng của tôi.

Tôi đã cố gắng đứng ở cuối phòng để có thể quan sát mọi người đang làm gì, đồng thời cũng là giảm thiệt hại gia tăng trong trường hợp tôi ngã (việc này xảy ra thường xuyên hơn mức tôi muốn thừa nhận).

Tôi thích các buổi học của Marco, nhưng tôi không thực sự hiểu điều gì đang diễn ra. Cứ thi thoảng Marco lại phải dùng bên thảm của tôi và điều chỉnh, giúp tôi thực hiện tư thế chính xác.

Trong một lần, Marco đã dùng chân của mình chống chân của tôi, để chúng vuông góc với sàn nhà. Sau đó anh ta chộp lấy tay của tôi và kéo.

“Sử dụng cơ bốn đầu... lấy hông làm trụ. Tốt. Giờ thì... thư giãn gương mặt đi nào”.

Lúc đó tôi đang nhăn nhó, và vì lý do cổ tay tôi chạm tới ngón chân mà tôi đã thả lỏng gương mặt.

## Tình trạng tiến thoái lưỡng nan kiểu “chủ gia đình”

Dù tôi thích những khóa học của Marco, nhưng thật khó có thời gian để dành cho chúng. Mỗi buổi học kéo dài một tiếng rưỡi, chưa kể thời gian đi bộ tới phòng học, thay đồ, tắm sau khi tập xong và lại đi bộ về nhà.

Dù phòng tập chỉ cách nơi tôi ở vài dãy nhà, nhưng tôi vẫn phải mất nguyên hai tiếng. Khi đó tôi đang làm việc tại nhà cho một công ty thuộc danh sách Fortune 500 với một công việc đòi hỏi khát khe, và tôi thường phải họp trùng với thời gian học. Chẳng ích gì khi các lớp học đều diễn ra vào khoảng giữa buổi sáng và buổi chiều, đồng nghĩa với việc nếu theo học tôi sẽ phải bỏ cả công việc. Nói chung, việc sắp xếp công việc của tôi khá linh động, nên chủ yếu là vấn đề tâm lý thôi: Thật khó để thuyết phục bản thân là hoàn toàn có thể vặn vẹo cơ thể trong hai giờ đồng hồ khi mà tôi vẫn còn nhiều việc phải làm.

Cuối cùng, tôi cũng đã học khoảng 15 buổi trong năm đó. Dù cảm thấy rất tuyệt sau mỗi buổi học, tôi vẫn không thấy thoải mái với việc phải dành thời gian cho nó.

Ngoài ra, tôi cũng không tập ở nhà. Tôi không biết làm thế nào. Ở lớp học, tôi bắt chước những học viên khác. Tôi có thể lăn trên tấm thảm ở nhà và duỗi thẳng người, nhưng trong suy nghĩ của tôi, đó không phải là yoga thực sự.

## Khi trò đã sẵn sàng, thầy sẽ xuất hiện

Sau khi chúng tôi chuyển tới Colorado, một tối nọ, Kelsey nói với tôi là cô ấy nhớ New York. Cô ấy nhớ những lớp học của Marco, và nhớ khóa học giải phẫu của Leslie Kaminoff mà cô ấy đã tham gia sau khi hoàn thành chứng chỉ giảng dạy yoga. Leslie là đồng tác giả của cuốn *Giải phẫu học Yoga* (Yoga Anatomy) (xuất bản lần đầu năm 2007, lần thứ hai năm 2011), một trong những cuốn sách về yoga bán chạy nhất mọi thời đại, và tài liệu của ông được sử dụng trong tất cả các chương trình đào tạo giáo viên dạy yoga trên khắp thế giới. Đáng tiếc là để theo khóa học của Leslie, bạn phải sống ở thành phố New York, vì thế Kelsey đã phải nghỉ học khi chúng tôi chuyển nhà.

“Nên có ai đó đưa khóa học của Leslie lên mạng”, cô ấy nói.

Thật chuyện cứ như châm biếm. Chỉ vài tuần trước, Kelsey còn giúp tôi tung ra khóa học trực tuyến đầu tiên của mình, vì thế cô ấy đã biết cách làm. Và cô ấy cũng đúng về khóa học của Leslie: Vì khóa học này chỉ dành cho những người sống ở thành phố New York, trong khi học sinh từ khắp mọi nơi trên thế giới đều muốn tham gia, nên sẽ ý nghĩa hơn nhiều nếu có thể đưa khóa học này lên mạng.

Tôi đã chỉ cho cô ấy một sự thật hiển nhiên là: Kelsey hoàn toàn có thể tạo ra một khóa học như thế nếu Leslie sẵn sàng để cô ấy thực hiện. Tối hôm đó, cô ấy đã gửi cho Leslie một lời đề nghị.

Hai hôm sau, Kelsey và Leslie đã đồng ý bắt tay vào thực hiện, và Kelsey đã thành lập công ty đầu tiên của cô ấy, công ty cung cấp khóa học yoga trực tuyến.

Vì cả hai chúng tôi cùng làm việc ở nhà, nên tôi bỗng nhiên được nghe *rất nhiều* về yoga. Một phần của quá trình sản xuất là quan sát hình ảnh ở mỗi lớp học, sau đó đưa ra những chú thích chi tiết cho người biên tập hình ảnh và viết lời. Kết quả là, tôi đã được nghe chi tiết về khóa học của Leslie khi Kelsey sản xuất nó.

## Thời điểm tôi quyết định học yoga nghiêm túc

Một trong những điều đầu tiên khơi gợi quan tâm, hứng thú của tôi với yoga là giọng nói của Leslie. Ông ấy không có vẻ gì của một người ẻo lả, như tôi vẫn nghĩ. Ngược lại, ông ấy có 30 năm kinh nghiệm về giải phẫu học, sinh lý học, y học thể thao, và không có khả năng chấp nhận những thuật ngữ nhẹ nhàng.

Theo quan điểm của Leslie, yoga rất có giá trị vì nó hữu ích. Khi luyện tập, yoga rất tốt cho việc tăng cường sức khỏe, cải thiện sự dẻo dai và duy trì vận động.

Yoga thậm chí còn hiệu quả hơn khi bạn tập trung vào hơi thở khi luyện tập. Hầu hết mọi người đều cho rằng yoga là sự kết hợp kỳ lạ của thể dục nhịp điệu, thể dục thể thao và vận người, nhưng không phải như vậy. Điều khiến yoga trở thành yoga là sự kết hợp giữa việc thở, chuyển động và trạng thái tinh thần lại với nhau.

Leslie cũng giải thích một số thuật ngữ yoga kỳ cục. Những người mới được thử nghiệm với cơ thể và tâm trí. Họ cố làm nghiên cứu khoa học trước khi những nguyên tắc hiện đại như giải phẫu học, sinh lý học và tâm lý học nhận thức được chính thức hóa.

Kết quả là, khi những người mới bắt đầu này thấy điều gì đó thú vị, lựa chọn duy nhất của họ là giải thích điều họ thấy bằng những câu chuyện hoặc ẩn dụ. Chẳng hạn, khi người tập yoga phát hiện ra rằng tư thế đặc biệt và kỹ thuật thở thường có xu hướng kích thích một số phản ứng tình cảm cụ thể thì họ có thể giải thích điều họ chú ý theo cách tốt nhất có thể.

Tôi *biết* là mình chưa hoạt động đủ, thế nên video này đã thu hút được sự chú ý của tôi. Liệu có phải “sự cứng và giòn” mà tôi cảm nhận về cơ thể mình chính là việc các cơ của tôi đang trở nên xơ cứng không? (Thật rùng mình!)

Đó là một cách giải thích hợp lý, nhưng không phải là khả năng duy nhất. Theo Leslie, đau lưng và đau cổ còn có thể là vì thiếu oxy chảy vào các cơ – hiện tượng vẫn được gọi là *thiếu máu cục bộ*. Vận động không đủ có thể gây thiếu máu chảy vào các cơ. Khi các cơ thiếu oxy, cơ quan thụ cảm nỗi đau bắt đầu bùng lên, tạo ra cơn đau dai dẳng, âm ỉ. Tình trạng này càng kéo dài, bạn càng bị đau.

Thường xuyên căng thẳng và nín thở cũng có thể góp phần gây thiếu máu cục bộ. Những nguyên nhân này thường đi kèm với nhau: Thường thì mọi

người sẽ nín thở khi có cảm giác căng thẳng. Nếu bạn giúp các cơ của mình nhận được nhiều oxy hơn bằng cách vận động, thở và giảm mức độ căng thẳng nói chung thì các cơn đau cơ sẽ biến mất.

Cử động thường xuyên và lượng oxy hít vào là hai yếu tố quan trọng giúp giải phóng cơn đau cơ và duy trì chuỗi hành động. Yoga kết hợp cử động, hơi thở và thiền, do đó nó đặc biệt hiệu quả trong việc làm giảm hoặc loại bỏ cơn đau cơ kinh niên.

Với tôi, thế là đủ. Tôi sẽ học cách tập yoga.

Giờ thì... tôi phải bắt đầu thế nào đây?

## Thực ra “Yoga” là gì?

Nếu tôi định tập yoga, tôi cần phải có một vài hiểu biết cơ bản về việc tôi định làm. Nếu yoga chỉ có nghĩa là kéo căng cơ thể, thì người ta hẳn đã gọi nó là “kéo căng” rồi.

Hóa ra, mấu chốt để xác định bản chất của yoga là phải hiểu được nó bắt nguồn từ đâu và làm thế nào nó trở thành “yoga” mà chúng ta gọi hiện nay?

Yoga đã tồn tại dưới nhiều dạng thức khác nhau suốt hàng nghìn năm. Bằng chứng được ghi nhận sớm nhất về yoga xuất hiện trong những vật dụng được khai quật ở di chỉ khảo cổ liên quan tới nền văn minh thung lũng Indus (nằm ở Pakistan và tây bắc Ấn Độ hiện nay) từ năm 3300 đến năm 1300 trước công nguyên.

Vừa là triết học vừa là thực hành, yoga là một bộ kỹ thuật phi tôn giáo được gắn kết với những truyền thống tôn giáo và triết lý của khu vực. Các thầy tu Veda (Vệ Đà)<sup>(1)</sup> cổ đại đã hoàn thành các nghi lễ tự nhiên phức tạp để kết nối thế giới vật chất với thế giới thần thánh nhằm tìm kiếm thần Brahma, “nơi trú ngụ tối thượng của mọi sinh vật”. Từ “yoga” bắt nguồn từ tiếng Phạn có nghĩa là *cái ách*. Theo đúng nghĩa con bò bị buộc vào cái cày để làm việc trên đồng. Thông qua các nghi lễ, các thầy tu đã cố gắng buộc thế giới tinh thần với thế giới vật chất.

Theo thời gian, truyền thống Veda (Vệ Đà) mất dần tầm quan trọng. Các thầy tu bắt đầu khám phá ý tưởng về *bản ngã* (atman), bản chất của cái tôi đích thực. Nghi thức tự nhiên bên ngoài chuyển thành thiền định nội tại, và những

người đang thực hành về mặt tinh thần) bắt đầu từ bỏ thể giới vật chất và nghĩ thức phức tạp để lang thang trong rừng sâu, thực hành thiền định và thực hiện lời thề sống nghèo khó, khổ hạnh để tìm kiếm sự thật cuối cùng.

Yājñavalkya, nhà hiền triết sống trong thời kỳ mà bây giờ chúng ta gọi là Upanishadic, đã cho rằng, chính việc hiểu được cái tôi của con người đã ngăn cản chúng ta biết được cái tôi đích thực của mình. Bằng cách bỏ đi cái tôi chúng ta có thể trở thành một người có bản ngã, “linh hồn” hoặc một cái tôi đích thực. Yājñavalkya cũng giới thiệu ý tưởng *nghiệp* (karma) – khái niệm mà ông định nghĩa là những hành động được thực hiện để được giải phóng khỏi cái tôi. Bằng cách tách biệt cơ thể với bản ngã, hoàn toàn có thể đạt được điều mà Đức Phật sau này gọi là sự khai sáng.

## Bước vào tập tư thế (asana)

Kiến thức triết học của **Yājñavalkya** đã dẫn đến sự phát triển của điều mà hầu hết những người luyện tập yoga đều coi là cốt lõi của yoga đương thời: Các tư thế kết hợp với việc hít thở và thiền định. Học giả về lịch sử tôn giáo Karen Armstrong giải thích trong cuốn *Sự biến đổi kỳ diệu* (The Great Transformation) rằng:

*Yoga là một trong những thành tựu vĩ đại nhất của Ấn Độ, và trong dạng thức phát triển nhất, ban đầu nó được (những người luyện tập) thiết kế để giải phóng purusha [cái tôi] khỏi sự khó khăn của tự nhiên.*

*Loại yoga cổ điển này rất khác so với loại yoga vẫn được dạy ở phương Tây hiện nay. Nó không phải là kiểu tập thể dục, và cũng không nhằm giúp người ta thấy tốt hơn về cuộc sống – mà hoàn toàn ngược lại. Yoga là sự tấn công có hệ thống vào cái tôi, là một chế độ ăn uống khát khe trong khoảng thời gian dài khiến người thực hành phải hủy bỏ nhận thức thông thường với những sai lầm và ảo tưởng, thay vào đó là cuộc khám phá diệu kỳ về cái tôi của bản thân.*

Yoga là công việc toàn thời gian. Nó không dành cho những người đua đòi, học kiểu tài tử, hay những người yếu tim. Yoga là sự khổ luyện tinh thần, dành riêng cho những thầy tu và những người có uy tín lớn. Những người tập yoga cổ điển không nhằm mục đích khỏe hơn hay linh hoạt hơn. Họ nỗ lực để chia rẽ mối liên kết giữa linh hồn (atman) và thể xác. Một nhiệm vụ nặng nề.

# Luật lệ của Yoga

Vào thế kỷ thứ II trước công nguyên, một học giả có tên là Patañjali đã bắt đầu tổng hợp các triết lý và bài tập yoga thời đó. Phân tóm lược những bài tập yoga tốt nhất (*raja* hay “hoàng gia”) của Patañjali đã trở thành *Yoga Sutras của Patañjali*, tập hợp những câu cách ngôn mà sau này trở thành tài liệu cơ sở cho những người tập yoga.

Trong cuốn *Yoga Sutras*, Patañjali đã phác họa tám mặt cơ bản, hay “phần” của bài tập yoga *hoàng gia*:

- | **Yama** – đạo đức
- | **Niyama** – tự thanh lọc
- | **Asana** – tư thế
- | **Pranayama** – kiểm soát hơi thở
- | **Pratyahara** – kiểm soát cảm giác
- | **Dharana** – ý định
- | **Dhyana** – thiền
- | **Samadhi** – miệt mài/trâm ngâm

Patañjali gọi hệ thống này là “*ashtanga*” hay yoga “tám chi”. Bằng cách luyện tập tất cả tám chi (nhánh) của yoga, người tập có thể trải nghiệm cảm giác *kaivalya*: Tách rời hoàn toàn linh hồn của người tập ra khỏi thế giới vật chất, đưa tới hạnh phúc mãi mãi.

Điều mà hầu hết mọi người nghĩ tới khi nghe thấy từ “yoga” – gặp người theo những tư thế lạ – là chỉ có một chi, *asana* (tư thế). Tư thế thường được luyện tập với các kỹ thuật thở đặc biệt gọi là *pranayama* (kiểm soát hơi thở). Mục đích của bài tập này là chuẩn bị cho việc thiền (*dhyana*).

Hệ thống của Patañjali coi yoga như một triết lý, chứ không phải một hệ



thông các bài tập. Tư thế thường bị hạn chế bởi một số tư thế ngồi cố định – khác xa so với những động tác mà người tập hiện đại vẫn hay tập.

Nói như vậy không có nghĩa là không có những tư thế mãnh liệt. Các tư thế yoga *Hatha* (mạnh mẽ) có từ thời Yājñavalkya, nhưng sự khỏe khoắn không phải là ưu tiên chính.

Tài liệu sưu tập nổi tiếng nhất về các bài tập yoga hatha là *Hatha Yoga Pradipika* của Maharishi Swatmarama, một nhà hiền triết thế kỷ XV. Swatmarama coi yoga hatha là cách lý tưởng để đạt được yoga *raja* (yoga tâm thức) của Patañjali. Bằng cách thanh lọc cơ thể thông qua các bài tập, luyện tập kiểm soát hơi thở và hít thở đều đặn trong lúc trồng cây chuối, Swatmarama tin rằng những người luyện tập yoga hatha có thể đạt được trạng thái ý thức cao hơn.

Chưa bao giờ các dạng thức ban đầu của yoga hatha được hiểu thực sự thấu đáo. Các bài tập tư thế mạnh mẽ vẫn là một nhánh ít người biết đến của yoga *raja* cho tới khi người Bà La Môn ở Mysore, Ấn Độ, cách mạng hóa những bài tập này bốn thế kỷ sau.

## Yoga = Thở + Chuyển động + Thiền định

Trở lại với hiện tại. Lịch sử đó có liên quan gì tới yoga hiện đại?

Đây là điều khiến tôi ngạc nhiên: Xét về mặt lịch sử, bài tập tư thế hiện đại chính là sự sáng tạo gần đây. Tất nhiên là phần triết lý của yoga đã có từ vài nghìn năm trước, nhưng bài tập thực sự với các tư thế tiêu tốn thời gian trong khi hít thở và thiền định lại hoàn toàn mới.

Việc nhấn mạnh vào sức khỏe và sự linh động trong các bài tập yoga cũng hoàn toàn mới. Rất ít người trong chúng ta chủ động thử tách linh hồn khỏi thể xác để tìm kiếm bản ngã thực sự của mình. Hầu hết những người luyện tập yoga hiện đại đều không phải những người tu hành khổ hạnh. Nhìn chung, những người luyện tập yoga hiện nay chủ yếu đều muốn tăng cường sức khỏe, cải thiện sự dẻo dai và loại bỏ căng thẳng.

Về cơ bản, rõ ràng các bài tập yoga hiện đại là sự kết hợp của chuyển động, hít thở và thiền định. Bản thân các tư thế không phải là điều kỳ diệu. Chúng đã thay đổi rất nhiều sau nhiều thế kỷ.

Và cũng rõ ràng yoga không phải là các chuyển động thu hút sự chú ý. Trong các bài tập đó có một triết lý: Bạn dành thời gian luyện tập để trở thành một người tốt hơn. Dù rằng các tư thế hiện đại đều mới, nhưng triết lý yoga đã có từ vài nghìn năm trước, và thật thú vị là triết lý đó đến giờ vẫn đúng. Yoga sâu sắc theo kiểu những cách chăm sóc sức khỏe hiện đại không có được.

Bạn không cần phải gồng mình để tập các tư thế điên rồ. Là một “yogi” (người tập yoga) thực sự không có nghĩa là phải bắt chước được các tư thế kiểu như tư thế Gumby, mà chỉ là luyện tập theo cách giúp bạn sống tốt hơn. Bạn không thể thực hiện được một tư thế vì bạn không đủ dẻo dai, không đủ khỏe hoặc vì bạn đang bị thương? Không vấn đề gì, hãy điều chỉnh tư thế đó, hoặc thực hiện một tư thế khác. Chỉ cần bạn tập trung vào hơi thở khi di chuyển, và bạn chú tâm tới điều cơ thể đang nói với bạn là bạn đã làm đúng rồi.

Về cơ bản, bài tập yoga hiện đại là sự kết hợp của việc hít thở, chuyển động và chú tâm thiền định, chứ không phải là các bài tập thể dục, nhịp điệu và cũng không nhằm mục đích trở nên siêu dẻo dai. Trở nên siêu dẻo dai hoặc điều luyện không khiến bạn trở thành một yogi giỏi hơn.

## Xóa bỏ những hiểu lầm

Dưới đây là một số hiểu lầm thường thấy về yoga:

- Bạn không cần phải biết hàng nghìn tư thế để tập yoga. Biết nhiều tư thế không khiến bạn trở thành một yogi giỏi hơn.
- Bạn không cần biết tên tiếng Phạn, thậm chí tên tiếng Anh của các tư thế. Biết tên của nhiều tư thế cũng không giúp bạn trở thành một yogi giỏi hơn.
- Một số người tập yoga theo Ấn Độ giáo, nhưng niềm tin tôn giáo không nhất thiết giúp ích cho việc tập các tư thế.
- Một số người tập yoga thực hiện chế độ ăn kiêng, ăn chay như những người giảm cân, hoặc tập Ayurveda<sup>(2)</sup>, nhưng chế độ ăn kiêng đặc biệt không phải là điều kiện tiên quyết để tập các tư thế yoga.
- Một số người tập yoga cũng quan tâm tới y học, tử vi và các chủ đề tương tự, nhưng những quan tâm này không phải là điều kiện tiên quyết để thực hành

yoga.

Tất cả những điều này đều là những hiểu lầm mà tôi có khi lần đầu tiên nghe nhắc đến yoga. Chỉ cần một chút thời gian tìm hiểu xem bài tập yoga *thực sự* là gì cũng đủ để giảm bớt những lo lắng ban đầu của tôi, cũng như giúp tôi không lãng phí thời gian cho những thứ không quan trọng.

## Yoga có nguy hiểm không?

Còn một điều cuối cùng mà tôi băn khoăn: Liệu tôi có làm mình bị thương khi tập yoga không?

Khoảng thời gian tôi nghiên cứu về các tư thế, có một bài báo mang tính kích động về việc luyện tập yoga được đăng trên tờ *New York Times*. Bài báo có tựa đề “Yoga tàn phá cơ thể bạn như thế nào” là bài viết của nhà báo thuộc chuyên mục khoa học William J. Broad. Thực ra bài báo được trích từ một cuốn sách của Broad là “*Khoa học về Yoga: Nguy cơ và phần thưởng*” (The Science of Yoga: The Risks and the Rewards) (2012), cuốn sách bị công kích mạnh mẽ sau khi được xuất bản chỉ vài tuần sau khi bài báo kia được đăng.

Bài báo gợi lại rất nhiều cuộc tranh cãi trong giới yoga. Broad đã trích dẫn một số nghiên cứu khoa học về lợi ích của việc tập các tư thế yoga, chủ yếu liên quan tới việc cân bằng tình cảm và cảm xúc. Sau đó ông chuyển sang tấn công, cho rằng yoga phải chịu trách nhiệm cho một loạt những chấn thương quan trọng, từ bong gân tới đột quỵ. Trong một số trường hợp, Broad còn cho rằng yoga phải chịu trách nhiệm cho những người luyện tập đã rơi vào trạng thái hôn mê.

Bài báo này là một ví dụ về *sự đảo ngược* có ích. Như thế nào là tập yoga không đúng, và những nguy cơ chủ yếu của việc luyện tập các tư thế yoga là gì?

Phản ứng của cộng đồng những người dạy yoga với bài báo và cuốn sách của Broad đều nhanh chóng và gay gắt: Dựa vào những chứng cứ cụ thể, có thể nói Broad đã kích động được mọi người. Yoga, giống như bất cứ hoạt động liên quan đến cơ thể nào khác, cũng có nguy cơ, nhưng những nguy cơ này không thể tổn hại nghiêm trọng tới bạn nếu bạn luyện tập theo cách an toàn.

Dù bạn thực sự có thể làm tổn thương bản thân khi tập yoga, nhưng những vấn đề về sức khỏe nghiêm trọng như đột quỵ hay hôn mê cực kỳ hiếm, và

không có bằng chứng nào cho thấy luyện tập các tư thế yoga có thể tăng nguy cơ bị thương nghiêm trọng. So với các hoạt động thể lực khác, ngay cả những môn thể thao không tiếp xúc như gôn, yoga cũng được thống kê là rất an toàn.

Đối với việc bong gân, căng cơ, rách cơ, những chấn thương kiểu này chắc chắn có thể xảy ra nếu bạn ép cơ thể phải thực hiện những tư thế khó, bất chấp cảm giác đau đớn và khó chịu. Nếu bạn tin rằng những yogi “đích thực” có thể gập người, chạm mũi vào mắt cá chân, và bạn ép cơ thể bạn phải thực hiện bằng được tư thế đó, bạn sẽ có nguy cơ bị giãn dây chằng hoặc đứt gân. Tương tự như vậy, nếu một người dạy yoga chỉnh tư thế của bạn một cách thô bạo, vượt qua giới hạn của cơ thể thì nhiều khả năng bạn sẽ bị chấn thương.

Khu vực dễ bị tổn thương nhất là đốt sống cổ. Một số tư thế như trồng cây chuối thường gây áp lực lên đầu và cổ, và trong lúc tập, nếu bạn ngã không đúng, bạn có thể bị thương nghiêm trọng ở cổ hoặc cột sống.

Dựa trên những thông tin này, chỉ cần tôi lắng nghe cơ thể mình, không ép mình phải tập gắt gao, và tránh những trọng lượng không cần thiết lên đốt sống cổ, tôi có thể thu lợi từ việc luyện tập mà không phải đối mặt với nguy cơ bị thương. Thật tuyệt khi biết điều đó.

## Tư thế khả thi tối thiểu

Tôi đã chuẩn bị tất cả để bắt đầu luyện tập... vậy tôi phải bắt đầu từ đâu?

Rõ ràng là có rất nhiều kiểu, dạng và triết lý về việc yoga “trông” như thế nào. Tất cả đều rất tốt, nhưng tôi cần phải lùi một bước và quyết định mục đích của yoga *đối với tôi* là gì.

Tại sao tôi lại làm việc này?

Tôi muốn năng động hơn, và cơ thể tôi đang thể hiện rõ ràng là tôi cần phải di chuyển nhiều hơn nữa. Bác sỹ của tôi khuyến cáo tôi nên tập các bài tập liên quan đến vận động, và nhắc đến yoga như một lựa chọn. Tôi đã biết về lợi ích của cả luyện tập và thiền định, và tôi thích yoga vì nó kết hợp được cả hai điều đó. Tôi không có thời gian để tới các phòng tập và theo học các lớp dài lê thê, vì thế tôi thích luyện tập ở nhà theo cách hiệu quả và an toàn.

Như thế là đã đủ chi tiết để xác định được mức độ thành thạo mục tiêu rồi. Dưới đây là mức độ thành thạo mục tiêu của tôi:

1. Học chuỗi tư thế mà tôi có thể luyện tập để tăng cường sự nỗ lực, sức khỏe hàng ngày và sự dẻo dai nói chung.
2. Kết hợp bài tập với việc hít thở và thiền định, thu được một số lợi ích thể chất và tinh thần quan trọng trong một khoảng thời gian ngắn.
3. Luyện tập an toàn dựa theo trí nhớ trong khoảng 20 đến 30 phút, điều chỉnh theo khoảng thời gian, sức khỏe, sự linh hoạt và mức độ năng lượng nói chung.

## Đồ dùng cần thiết để tập yoga

Trước khi bắt đầu luyện tập, tôi cần một số vật dụng cơ bản. Đầu tiên và quan trọng nhất là một tấm thảm, làm đệm lót và giúp tay, chân không bị trượt khỏi vị trí. Tập yoga có thể ra rất nhiều mồ hôi, và nếu bạn bị trượt chân hoặc tay, bạn có thể bị thương, vì thế một tấm thảm tốt là thứ rất quan trọng.

Khi tôi tập ở trung tâm Yoga Pure, họ cấp cho tôi một tấm thảm tốt. Tuy nhiên, tôi thấy tấm thảm hơi ngắn so với tôi. Tôi để ý thấy mình hay bị trượt ra mép thảm, và mỗi lần như vậy tôi lại phải dừng lại để điều chỉnh. Có lẽ sẽ tốt nhất nếu tôi có thể tìm được một tấm thảm dài hơn.

Sau khi nghiên cứu một chút, tôi đã đặt mua một tấm thảm PRO dài 2,15m của Manduka. Đó là một tấm thảm dài, bền và có màu xanh rất đẹp.

Ngoài ra, tôi còn mua một chiếc túi đựng thảm và vài chiếc khăn tay, phòng trường hợp tay tôi đổ mồ hôi. Tôi đã có quần soóc và áo thun, và vài chiếc quần dài tập yoga, vì thế tôi đã sẵn sàng về mặt trang phục. Các tư thế yoga được tập bằng chân trần, vì thế tôi không cần loại giày đặc biệt nào cả.

Theo gợi ý của Kelsey, tôi đã mua một sợi dây cotton dài và một đoạn tre, vì cô ấy nói những thứ đó có thể có ích khi điều chỉnh một số tư thế.

Đó là tất cả những trang thiết bị mà tôi cần. Bạn không cần nhiều thứ để tập yoga.

## Học các tư thế

Đây là chiến lược ban đầu của tôi: Kelsey là một giáo viên dạy yoga đã qua đào

tạo, vì thế, tôi nên bắt đầu với sự hướng dẫn của cô ấy. Chúng tôi quyết định dành ra 90 phút một buổi chiều để học những điều cơ bản.

Nếu tôi không có một giáo viên dạy yoga ngay trong gia đình, hẳn tôi sẽ phải thuê một giáo viên hướng dẫn riêng. Cách học một thầy một trò trong trường hợp này rất hữu ích, vì chuyển động là thứ khó có thể học được từ sách vở.

Nhưng điều đó không có nghĩa là tôi bỏ qua việc đọc sách. Tôi đã mua một số cuốn sách như:

, *Giải phẫu học yoga* của Leslie Kaminoff và Amy Matthews (2011)

, *Yoga vì sức khỏe* của Gary Kraftsow (1999)

, *Ashtanga Yoga: Triết lý và thực hành* của Gregor Maehle (2007)

, *Yoga tập trung vào hơi thở cùng với Leslie Kaminoff* (DVD) của Leslie Kaminoff (2010)

Tôi sử dụng những cuốn sách đó để tham khảo là chủ yếu. Trong đó có tranh ảnh và hình minh họa về các tư thế, cũng như những hướng dẫn chi tiết về cách thực hiện đúng mỗi tư thế, vì vậy tôi có thể tra cứu thông tin nếu cần. Tôi sẽ biết nhiều hơn về điều tôi cần phải học sau khi bắt đầu luyện tập.

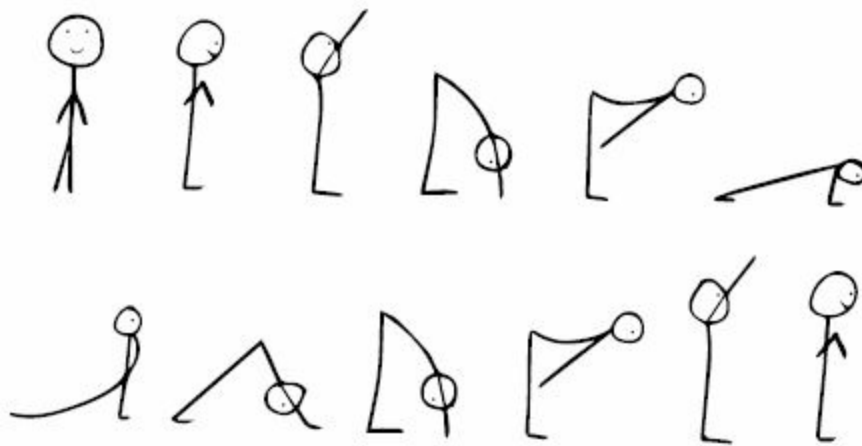
Tôi và Kelsey trải thảm ra và bắt đầu. Trước tiên là bài tập chào mặt trời (Sun Salutation).

## Tuần tự bài tập chào mặt trời

Bài tập chào mặt trời là một chuỗi các tư thế được thiết kế để giúp bạn khởi động cơ thể, vì thế hầu hết các tư thế đều bắt đầu với một số vòng tròn.

Chú thích nhanh nếu bạn quyết định thử tập những động tác này: Mỗi người mỗi khác, và cách “đúng” để thực hiện mỗi tư thế này tùy thuộc vào giới hạn riêng của từng cơ thể. Nếu bạn chú ý tới điều cơ thể đang nói với bạn, tránh tập quá mức và điều chỉnh mỗi tư thế theo giới hạn của cá nhân, bạn có thể luyện tập tư thế này một cách an toàn.

Dưới đây là hình minh họa cho chuỗi tư thế này:



1. Hít vào. Đứng thẳng ở phần trước tám thảm, tay thả lỏng theo người, đầu gối chùng.
2. Thở ra. Chắp hai tay trước ngực theo tư thế “cầu nguyện”.
3. Hít vào đồng thời vươn thẳng hai tay lên cao cho tới khi hai tay cao quá đầu. Sau đó, mắt nhìn theo hai tay. (Nếu bạn thấy không thoải mái khi nhìn như vậy, thì chỉ cần nhìn thẳng cũng được.)
4. Thở ra và gập người về phía trước, lấy hông làm trọng tâm, đặt hai tay xuống xuống sàn, chạm vào các đầu ngón chân. Vươn hết sức có thể nhưng không được vươn quá mức. Có thể gập đầu gối cũng được. Thư giãn cổ và vai.
5. Hít vào, mắt nhìn bức tường phía trước, tay vẫn chạm đất. (Ngoài ra có thể duỗi thẳng hai chân, đồng thời tiếp tục thư giãn đầu và cổ.)
6. Thở ra khi quay lại tư thế gập người về phía trước.
7. Hít vào khi đưa chân ra phía sau trong tư thế “chống đẩy”, để cơ thể được dẫn hết cỡ, trọng lượng cơ thể dồn lên hai tay và mũi bàn chân. Bạn có thể thu chân về và nhảy lên nếu muốn.
8. Thở ra, hạ thấp cơ thể xuống sàn như thể bạn đang chống đẩy. Không được tì hẳn xuống sàn, phải dừng lại khi cách thảm vài cm. Vai, khuỷu tay và cổ tay phải tạo thành các góc vuông 90 độ, để tránh bị thương. Nếu lúc đầu thấy tư thế này quá khó, bạn có thể tì đầu gối xuống sàn. (Tư thế này được gọi là Chaturanga Dandasana hay gọi tắt là Chaturanga.)

9. Hít vào khi bạn đẩy người về phía trước, cong lưng khi đẩy cho tới khi thân người bạn thẳng đứng. Đồng thời, xoay chân sao cho mũi bàn chân chạm đất, và căng chân cách thắm vài cm.
0. Thở ra và xoay chân lại, sao cho gót chân chạm thắm. Đồng thời, nâng hông lên cao cho tới khi cơ thể tạo thành hình chữ V. Thư giãn cổ và vai. Nếu bạn có cảm giác căng ở các gân khoeo, có nghĩa là có tác dụng, vì hãy tưởng tượng như bạn đang “thở” vào đó. Duy trì tư thế này trong từ ba đến năm nhịp thở.
  1. Hít vào, sau đó thở ra khi bạn đưa hoặc đẩy chân về phía trước, cho tới khi chân ở dưới vai.
  2. Lặp lại bước 5 và 6.
  3. Hít vào trong khi giơ hai tay lên cao cho tới khi hai tay cao quá đầu, giống như bước 3.
  4. Thu hai tay về trước mặt cho tới khi tay chắp lại trước ngực trong tư thế cầu nguyện, như ở bước 2.

Bạn sẽ nhận thấy là chuỗi tư thế này kết thúc với tư thế giống như lúc bắt đầu. Khi bạn đã hoàn thành chuỗi tư thế này, bạn có thể lặp lại bao nhiêu lần tùy ý, hoặc dễ dàng chuyển sang một tư thế đứng khác.

Còn nữa, bạn có để ý mỗi bước đều bắt đầu với việc “hít vào” hoặc “thở ra” không? Việc đó là có chủ đích, mỗi chuyển động đều tương ứng với một hơi thở. Tập trung vào hơi thở chính là điểm tạo nên sự khác biệt giữa yoga và thể dục nhịp điệu.

Bạn có thể coi bài tập chào mặt trời như “tư thế khả thi tối thiểu”: Nếu bạn thực sự muốn, toàn bộ bài tập của bạn có thể chỉ gồm các bài tập chào mặt trời được lặp đi lặp lại. Chuỗi tư thế này hợp với tất cả các nhóm cơ của cơ thể, đòi hỏi sự kết hợp nhuần nhuyễn giữa sức mạnh và sự dẻo dai. Đây là bài tập có tính thử thách cao nhưng không hề phức tạp.

Trong bài tập ashtanga truyền thống, chào mặt trời (hay còn gọi là *vinyasa*) được thực hiện giữa mọi tư thế riêng lẻ. Điều đó có nghĩa là bạn dành phần lớn thời gian trong hai tiếng thực hiện các bài tập chính để thực hiện chuỗi tư thế đơn giản này. Đây chính là ví dụ hoàn hảo cho quy luật sức mạnh trong thực tế: Các tư thế phụ cấu thành nên phần lớn bài tập.



Tuy nhiên, cơ thể con người có phản ứng với chấn thương, căng thẳng lặp đi lặp lại: Nếu bạn thực hiện quá nhiều bài tập chào mặt trời, có khả năng bạn sẽ tự làm bị thương chính mình. Đó chính là lý do tại sao chỉ nên tập một vài lần bài tập chào mặt trời để khởi động, sau đó chuyển sang những tư thế khác.

## Hãy nhớ thở như Darth Vader<sup>(3)</sup>

Thông thường, khi tập các tư thế sẽ cần sử dụng một thủ thuật thở đặc biệt gọi là *ujjayi*. Đây là dạng thức đơn giản nhất của kỹ thuật thở (kiểm soát hơi thở) được sử dụng trong khi luyện tập yoga, được thiết kế để có thể dễ dàng tập trung vào hơi thở hơn.

Phía sau cổ họng có một bộ phận nhỏ được gọi là thanh môn, gần như ở cùng vị trí với amidan và dây thanh âm. Nếu bạn tập trung vào khu vực đó trong một lúc, bạn có thể chủ động thu nhỏ thanh môn lại một chút, từ đó kiểm soát không khí vào và ra lá phổi của bạn.

Có thể nhìn thấy được tác động của việc hạn chế này: Bạn có thể nghe thấy (và cảm nhận được) luồng khí lạnh ulla vào lá phổi của bạn khi bạn hít vào, và luồng khí nóng tràn ra khi bạn thở ra.

Các tài liệu yoga cổ đại gán cho *ujjayi* và những kỹ thuật kiểm soát hơi thở khác những đặc điểm thần bí. Bất chấp *ujjayi* có “tạo nhiệt trong cơ thể” hoặc “khuyến khích dòng chảy của sự sống” hay không, thì chắc chắn nó vẫn khiến chúng ta dễ dàng tập trung vào hơi thở hơn.

Mục đích chính của *ujjayi* là giúp giữ hơi thở ổn định, thư thái và đều đặn. Bằng cách duy trì *ujjayi* như một bài tập, tâm trí của bạn có thể dễ dàng tập trung vào hơi thở hơn, giúp bạn tập trung chú ý. Nếu bạn chỉ quan tâm tới hơi thở, không quan tâm tới bất cứ thứ gì khác, bạn sẽ thiền được mà không cần phải cố gắng.

Tôi đã học được hầu hết các chuỗi tư thế hay được sử dụng trong yoga, cũng như những kiểu thở cơ bản trong khoảng 20 phút.

Bây giờ, hãy cùng tôi học những tư thế đứng phổ biến.

Các tư thế đứng

## CHIẾN BINH 1



Tư thế chiến binh là phần cốt lõi ở các lớp học ashtanga và vinyasa. Tư thế này thường xuất hiện sau (hoặc giữa) chuỗi tư thế chào mặt trời, và có thể thực hiện như một chuỗi tư thế riêng.

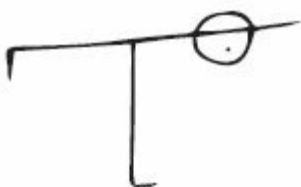
Đây là cách thực hiện tư thế chiến binh thứ nhất: Đứng thẳng. Hít vào, một chân lùi lại phía sau, mũi bàn chân lùi lại tạo thành một góc khoảng 45 độ. Hai tay giơ cao lên đầu, song song với nhau. Chân trước nên gập đầu gối, bàn chân duỗi thẳng, căng chân gàn như vuông góc với sàn. Bàn chân sau duỗi thẳng trên mặt sàn. Giữ tư thế đó trong khoảng từ ba đến năm nhịp thở.

## CHIẾN BINH 2



Từ tư thế chiến binh 1, dang hai tay và xoay hông về một phía của tấm thảm. (Nếu chân phải ở phía sau thì xoay sang phải và ngược lại.) Giữ cho hai tay song song với mặt đất, “mở rộng” thông qua ngực. Giữ tư thế này trong khoảng ba đến năm nhịp thở.

## CHIẾN BINH 3



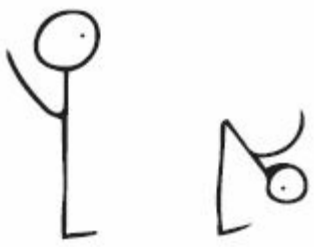
Tư thế chiến binh 3 có thể dễ dàng được gọi là tư thế siêu nhân. Mục tiêu lần này là di chuyển phần thân về phía trước trong khi vẫn giữ được thăng bằng trên một chân, với toàn bộ phần còn lại của cơ thể song song với mặt đất.

Từ vị trí đứng trung lập (vị trí đứng bình thường, hoặc chiến binh 1), giơ hai tay lên cao qua đầu, nhẹ nhàng nhấc một chân lên khỏi sàn. Di chuyển chân đó về phía sau ở trên không, đồng thời đẩy phần thân về phía trước, lấy hông làm trung tâm, cho tới khi đạt được tư thế siêu nhân. Hãy tập trung vào việc cân bằng các cơ ở bàn chân, giữ cho phần đầu gối phía dưới vững chắc nhưng không được gò cứng.

Giữ nguyên tư thế trong khoảng ba đến năm nhịp thở, sau đó quay trở lại tư thế đứng ban đầu.

## TƯ THẾ VẠN MÌNH KAMINOFF

Từ tư thế đứng thẳng, hít vào, xoay hai tay ra ngoài và đưa lên trên như lúc bắt



đầu chuôi tư thế chào mặt trời. Sau đó, thở ra và xoay hai tay ngược lại, hướng vào trong và đưa xuống dưới, đồng thời gập người xuống phía trước sao cho ngực chạm vào đầu gối. Lặp lại vài lần, hít vào khi xoay ra và thở ra khi gập người xuống.

Đây là tư thế đứng yêu thích của tôi: Tư thế này đem lại cảm giác *rất tuyệt* sau khi bước chân ra khỏi giường, hoặc sau vài tiếng ngồi máy tính. Lợi ích của tư thế này là nó không gia tăng trọng lượng lên cánh tay hay bàn tay, rất tốt nếu bạn bị hội chứng ống cổ tay, đang phục hồi sau chấn thương hoặc cần tăng cường sức mạnh, sự dẻo dai trước khi thử sức với những tư thế khó hơn.

## GẬP NGƯỜI VỀ PHÍA TRƯỚC



Từ tư thế đứng thẳng ban đầu, lấy hông làm trọng tâm và gập người về phía trước, có thể khụy đầu gối nếu cần. Thả lỏng cổ và vai, cúi đầu xuống khi cố gắng chạm các ngón tay xuống sàn.

Thử khi gập và không gập đầu gối, chú ý lúc bạn bắt đầu cảm thấy căng cơ. (Nếu gân khoeo của bạn linh hoạt, bạn có thể chạm cả lòng bàn tay xuống sàn, nhưng không được cố quá mức.) Sau khoảng ba đến năm nhịp thở, hãy quay trở lại tư thế đứng.

## GẬP NGƯỜI VỀ PHÍA TRƯỚC, CHÂN DANG RỘNG



Từ tư thế đứng thẳng, dịch chuyển hai chân về hai phía, tạo thành tư thế hình chữ V rộng. Sau đó, lấy hông làm trung tâm, gập người về phía trước. Thả lỏng cổ và vai, cúi đầu xuống khi cố gắng chạm lòng bàn tay xuống sàn. Sau khoảng ba đến năm nhịp thở, trở lại tư thế đứng thẳng.

## TƯ THẾ TAM GIÁC



Từ tư thế đứng thẳng, di chuyển hai chân về hai phía, tạo thành tư thế hình chữ V rộng, bàn chân hướng về phía trước. Sau đó, xoay hông sao cho một trong hai chân hướng về mép thảm. Gập đầu gối của chân đó và để khụy tay tì lên đó. Tay còn lại giơ cao lên trên. Cuối cùng, duỗi thẳng đầu gối vừa gập, và để cánh tay chạm mũi chân hoặc chạm sàn. Giữ tư thế này trong khoảng ba đến năm nhịp thở, trở về tư thế chân dang rộng, sau đó lặp lại các động tác với nửa người còn lại.

## TƯ THẾ GÓC BÊN MỞ RỘNG



Từ tư thế đứng thẳng, di chuyển hai chân sang hai bên, tạo thành hình chữ V rộng, chân hướng về phía trước. Sau đó, xoay hông sao cho một trong hai chân hướng về mép thảm. Gập đầu gối chân đó và để khuỷu tay tì lên đó. Tay còn lại giơ cao lên trên, sau đó (nếu thoải mái) chạm lòng bàn tay xuống sàn. Cánh tay của bạn nên vuông góc với sàn, tạo thành một

đường thẳng.

Có một biến thể của tư thế góc bên mở rộng mà tôi rất thích: Bạn có thể chấp hai tay trong tư thế “cầu nguyện” thay vì duỗi tay chạm sàn và giơ cao lên trên. Sau khi giữ tư thế này trong khoảng ba đến năm nhịp thở, hãy trở về tư thế dang rộng và lặp lại các động tác với bên còn lại.

## TƯ THẾ CÁI GHÉ



Đứng thẳng, chân chụm vào nhau, tay buông thõng theo cơ thể. Hít vào và giơ thẳng tay lên quá đầu. Sau đó thở ra và gập đầu gối, vẫn giữ hai chân chụm vào nhau cho tới khi bạn tạo thành tư thế cái ghế, nghĩa là giống như bạn đang ngồi trên một chiếc ghế tưởng tượng. (Tư thế này khó, và sẽ càng khó hơn nếu bạn giữ tư thế này lâu hơn.)

Giữ tư thế này trong khoảng ba đến năm nhịp thở, sau đó trở lại tư thế đứng thẳng.

## TƯ THẾ CÁI CÂY



Đứng thẳng, hai chân chụm vào nhau, hai tay buông thõng theo cơ thể. Nhấc một chân lên, gập đầu gối lại và dùng tay cùng chiều giữ mắt cá chân. Giữ thẳng bằng trên chân còn lại, đặt bàn chân của chân kia lên đùi chân trụ. Có lẽ bạn sẽ nhận thấy các cơ của chân trụ vận động điên cuồng cho tới khi bạn đạt được trạng thái cân bằng.

Một khi chân đã ở vào đúng vị trí, chấp hai tay trước ngực trong tư thế cầu nguyện. Giữ tư thế này trong khoảng từ ba đến năm nhịp thở, sau đó lặp lại động tác với chân kia.

Các tư thế trên sàn

## TƯ THẾ CON MÈO/CON BÒ



Quỳ gối và chống tay, tay vai và chân cùng rộng bằng vai. Thả lỏng cổ và vai, mắt nhìn xuống sàn.

Thở ra khi khom lưng, hóp bụng lại. Bạn sẽ thấy mình nhìn về phía hai chân. Đây là tư thế con mèo.

Tiếp theo, hít vào khi bạn duỗi thẳng lưng và trở về vị trí ban đầu, sau đó khom lưng theo hướng ngược lại, kéo bụng xuống sát sàn, giống như lúc bắt đầu bài tập chào mặt trời. Bạn sẽ thấy mình hướng mắt về phía trước, và lưng bạn lõm xuống. Đây là tư thế con bò.

## TƯ THẾ NGỒI GẬP NGƯỜI VỀ PHÍA TRƯỚC



Ngồi trên thảm tập, chân duỗi thẳng về phía trước. Ngón chân vuông góc với sàn. Không gập đầu gối, xoay hông khi sử dụng cơ tứ đầu (cơ ở phía trước bắp đùi), sau đó với tay về phía bàn chân. Với càng xa càng tốt mà không thấy đau hoặc khó chịu.

Giữ tư thế này trong khoảng ba đến năm nhịp thở.

## TƯ THẾ GẬP NGƯỜI VỀ PHÍA TRƯỚC, MỘT CHÂN GẬP



Hãy nghĩ về tư thế này như sự kết hợp của tư thế cái cây và tư thế gập người về phía trước. Nhấc một chân lên, gập đầu gối lại, sau đó xoay chân sang một bên. Đặt bàn chân vào phía trong đùi, sau đó lấy hông làm trọng tâm, giống như trong tư thế gập người về phía trước. Hãy với càng xa càng tốt mà bạn không thấy đau

hoặc khó chịu, sau đó giữ tư thế này trong khoảng ba đến năm nhịp thở.

## TƯ THẾ STAFF



Tư thế này khó hơn về ngoài của nó. Ngồi lên tám thảm, chân duỗi thẳng trước mặt. Bàn chân vuông góc với sàn. Chống hai tay xuống sàn, sau đó đẩy tay xuống sàn, nhẹ nhàng nâng cơ thể lên khỏi sàn, cố gắng để ngón chân vẫn hướng thẳng lên. Giữ tư thế đó trong khoảng ba đến năm nhịp thở.

## TƯ THẾ GÓC GIỚI HẠN

Ngồi xuống sàn, đầu gối tì vào ngực. Sau đó, di chuyển đầu gối về hai bên cho tới khi chúng gần chạm sàn, và lòng bàn chân của hai chân chạm vào nhau.



Dùng tay giữ bàn chân, và giữ tư thế này trong khoảng ba đến năm nhịp thở.

## TƯ THẾ NGỒI VẶN XOẢN



Ngồi trên thảm, chân duỗi thẳng trước mặt. Một chân gập đầu gối vào sát ngực cho tới khi bàn chân gần với đầu gối của chân còn lại. Sau đó, lấy tay nhắc bàn chân đó đặt lên chân đang duỗi thẳng. Chồng hai tay xuống sàn giống như trong tư thế staff.

Khi đã thực hiện được tư thế đó, vòng một tay qua cơ thể cho tới khi khuỷu tay chạm vào cạnh bên đầu gối. (Nếu chân phải đang gập, bạn sẽ phải vòng tay trái.) Gập khuỷu tay và giữ cho các ngón tay hướng lên trên. Tiếp tục vặn người bằng cách nhẹ nhàng đặt cánh tay lên chân. Hoàn thiện động tác vặn toàn bộ cột sống bằng cách xoay đầu theo cùng hướng khi bạn vặn người.

Có một điều rõ ràng ở tư thế này, đó là bạn hoàn toàn có thể nhận ra sự khác biệt khi quay các phần khác nhau của cột sống. Theo Leslie Kaminoff, đốt sống ngang lưng chỉ quay được một góc khoảng 5 độ. Đốt sống ngực quay một góc 35 độ, và đốt sống cổ xoay một góc khoảng 80 đến 90 độ, vì thế khi bạn xoay được, bạn có thể nhìn được phía sau mình.

Giữ tư thế vặn người này trong khoảng ba đến năm nhịp thở, sau đó quay trở lại tư thế ban đầu và tiếp tục với bên kia.

## TƯ THẾ CHÈO THUYỀN



Đây là một tư thế khó thực hiện. Ngồi lên tám thảm, hai chân duỗi thẳng trước mặt. Gập đầu gối sao cho bạn vẫn giữ được thẳng bằng khi ngồi, và ngón chân chạm nhẹ xuống sàn. Duỗi thẳng tay ra trước mặt, sau đó chụm hai chân lại với nhau khi nhắc chân lên khỏi sàn. Giữ hai cẳng chân thẳng và sát lại với nhau, có thể điều chỉnh nếu cần để giữ thẳng bằng. Giữ tư thế này trong khoảng ba đến năm nhịp thở, sau đó bỏ ra.

## TƯ THẾ CON QUẠ

Ngồi xồm trên thảm, chồng lòng bàn tay xuống sàn. Phía trong của khuỷu tay tiếp xúc với đầu gối tương ứng, sau đó đẩy người về phía trước cho tới khi toàn



bộ trọng lực cơ thể dồn về phía trước, và chân nhắc hẳn lên khỏi mặt đất. Giữ thẳng bằng trên hai tay càng lâu càng tốt, giữ hơi thở đều đặn.

## TƯ THẾ BÁNH XE



Nằm ngửa trên thảm, lưng sát sàn. Co hai đầu gối lên cho tới khi bàn chân đặt thẳng trên thảm, ngay dưới hông. Cùng lúc đó, đặt hai tay ở cạnh tai, lòng bàn tay úp xuống. Khi hít vào, kéo mạnh chân và đẩy mạnh tay sao cho cả cơ thể được nhắc

bồng lên khỏi thảm. Giữ tư thế này trong khoảng ba đến năm nhịp thở, sau đó từ từ hạ thân mình xuống sàn.

## Điều chỉnh độ khó

Bạn có để ý là những tư thế này thường kết thúc với câu “giữ trong khoảng ba đến năm nhịp thở” không? Đó là sự điều chỉnh có chủ đích. Thở là điều tạo nên sự độc đáo của yoga, vì thế, nó phải là một phần quan trọng của tất cả các tư thế. Thở từ ba đến năm nhịp là một khởi đầu tốt đối với những người mới bắt đầu.

Những người luyện tập các tư thế yoga thành thục không nhất thiết phải thực hiện các tư thế phức tạp hơn, hay nhiều tư thế hơn, mà họ chỉ thực hiện các tư thế đó chậm hơn, kiểm soát nhiều hơn và giữ mỗi tư thế lâu hơn. Khi tiến bộ hơn trong các bài tập, tôi cũng sẽ giữ mỗi tư thế thêm một vài nhịp thở.

## Tư thế xác chết (shavasana)



*Shavasana*, theo truyền thống yoga, là mục đích cuối cùng của tất cả những bài tập gập người và thở này. Hãy

nhớ, mục đích ban đầu của các tư thế là để chuẩn bị cho việc ngồi thiền. Truyền thống đó đã được nối tiếp trong các bài tập hiện đại vì một lý do đơn giản: Nó thực sự tốt.

Shavasana là sự giải phóng tất cả các cơ bị căng kết hợp với thiền sâu. Nó thường được gọi là tư thế xác chết, nhưng đừng để cái tên đó khiến bạn chùn bước. Tư thế này thực ra rất đơn giản: nằm thẳng trên thảm, hai chân duỗi thẳng, hai tay để dọc hai bên. Nhắm mắt và giải phóng toàn bộ căng thẳng

khỏi cơ thể. Các cơ của bạn không cần phải làm bất cứ việc gì.

Quét một lượt cơ thể bạn, từ đầu tới chân, và bất cứ khi nào bạn nhận thấy có cơ nào co rút, hãy giải phóng chúng, kể cả những nhóm cơ mà chúng ta thường không mấy quan tâm như cổ, và thậm chí là cả lưỡi. Lúc này, bạn cũng không kiểm soát hơi thở và bắt đầu thở bình thường.

Cảm giác tựa hồ bạn đang tan chảy. Sau một khoảng thời gian rất dài liên tục ép căng các cơ, shavasana thực sự là một sự thả lỏng. Cơ thể kiệt quệ có vẻ khiến bạn phải thư thái đầu óc, chính vì thế sẽ dễ thiền hơn khi bạn thực hiện tư thế shavasana.

Duy trì tư thế shavasana trong khoảng năm phút, sau đó đứng dậy. Bài tập các tư thế của bạn đã hoàn thành.

## Tóm lược phương pháp

Lúc này, tôi đã đạt được mức độ biểu hiện mục tiêu của mình. Sau tổng cộng ba tiếng hướng dẫn, tôi đã có thể tập các tư thế yoga ở nhà theo cách an toàn và hiệu quả.

Hãy cùng tóm lược lại những điểm căn bản trong phương pháp mà tôi đã sử dụng để học yoga:

• Tôi mua những vật dụng cần thiết: thảm, dây buộc...

• Tôi đã dành vài tiếng để học những điều cơ bản và điều chỉnh các hiệu làm cùng với một người hướng dẫn có kinh nghiệm.

• Tôi đã học được rằng các bài tập yoga liên quan tới chuyển động thông qua các tư thế mà vẫn tập trung vào hơi thở và duy trì trạng thái tinh thần sáng suốt.

• Tôi đã học được các chuỗi tư thế chào mặt trời, 10 tư thế đứng phổ biến và 10 tư thế ngồi/vặn mình phổ biến.

• Tôi đã học được các kỹ thuật thở (kiểm soát hơi thở) khiến cho việc tập trung vào hơi thở trong lúc tập trở nên dễ dàng hơn.



Tôi đã học được cách tránh bị thương bằng cách chú ý tới cơ thể, điều chỉnh các tư thế nếu cần thiết, tránh tập quá mức và tránh dồn trọng lượng không cần thiết lên đốt sống cổ.

Tôi đã luyện tập cho tới khi tôi có thể hoàn thành các chuỗi tư thế theo trí nhớ, mất khoảng 25 phút.

## Từ đây tôi sẽ đi đâu tiếp

Hiện tại, tôi vui với việc luyện tập các tư thế yoga của mình. Sau khoảng ba tiếng nghiên cứu và học hỏi, tôi đã có thể tập yoga bất cứ khi nào tôi muốn hoặc cần, và tôi không còn phải phụ thuộc vào các lớp học có sự chỉ dẫn của giáo viên ở trung tâm nữa. Tôi đã đặt lịch tập là 25 phút vào buổi sáng hoặc buổi tối.

Tôi thích yoga kết hợp các bài tập với việc thiền định. Sau khi luyện tập, rõ ràng là cơ thể tôi cảm thấy tốt hơn rất nhiều, kể cả cảm giác của tôi. Nếu tiếp tục luyện tập, chắc chắn tôi sẽ đạt được thành quả cả về thiền định và các bài tập chuỗi chuyển động.

Vấn đề là tôi cần phải luyện tập thường xuyên để đạt được kết quả đó. Mấu chốt là phải dành thời gian mỗi sáng và mỗi tối để luyện tập. Thay vì bắt đầu một ngày bằng việc vào mạng kiểm tra thư điện tử hoặc bắt tay ngay vào công việc, tôi sẽ thay đổi lịch trình của mình để có thể luyện tập yoga trước khi làm bất cứ việc gì khác.

Chăm sóc cơ thể cần trở thành ưu tiên hàng đầu của tôi, chứ không phải ưu tiên cuối cùng. Yoga sẽ giúp tôi khỏe mạnh trong nhiều năm tới.

Tôi không cần trở thành một thầy tu hay chuyên gia uốn dẻo mới có thể tận hưởng được những lợi ích của yoga. Tôi không cần biết mọi tư thế. Tôi chỉ cần luyện tập những điều cơ bản – những điều không khó chút nào.

Hơn thế nữa, tôi đã học được những gì tôi muốn học chỉ trong ba tiếng, tốt hơn rất nhiều so với kỳ vọng của tôi. Tôi bắt đầu dự án này với hi vọng học được một loạt những tư thế phức tạp, nhưng hóa ra không cần phải như vậy. Tôi có thể đạt được điều tôi muốn bằng cách đơn giản hóa, điều đó giúp tôi có thể đạt được kết quả mà mình muốn trong khoảng thời gian ít hơn.

Khi học một thứ mới, sẽ chẳng ích gì nếu phức tạp hóa nó lên.

# LẬP TRÌNH

---

## BÀI HỌC: VIỆC PHỨC TẠP SẼ TRỞ NÊN ĐƠN GIẢN KHI BẠN CHIA NHỎ CHÚNG RA

**Nếu gỡ lỗi là quá trình loại bỏ các lỗi thuộc về phần mềm thì lập trình lại là quá trình đặt các lỗi đó vào.**

**–EDSGER DIJKSTRA, nhà khoa học máy tính danh tiếng**

...

Tôi bắt đầu kiếm sống nhờ web từ năm 2007, khi tôi bỏ công việc toàn thời gian là trưởng phòng marketing cho công ty Procter & Gamble để thành lập công ty tư vấn và xuất bản của riêng mình.

Trang web quan trọng nhất của tôi, PersonalMBA.com, là phương kế sinh nhai của tôi: Tôi là một giáo sư kinh doanh hiệu quả, nhưng tôi không làm việc ở trường đại học. Mỗi năm, tôi lại cập nhật danh sách những cuốn sách kinh doanh tốt nhất dành cho những người muốn tự học kiến thức cơ bản nhất của kinh doanh.

Danh sách tài liệu nên đọc của *The Personal MBA* có một lượng độc giả yêu thích lâu năm kể từ khi ấn bản đầu tiên được phát hành năm 2005, và phần cập nhật cho danh sách đáp ứng được làn sóng quan tâm của các độc giả trên khắp thế giới. Kể từ năm 2005, PersonalMBA.com đã được hơn hai triệu lượt truy cập.

Cuốn sách đầu tiên của tôi, *MBA cá nhân: Làm chủ nghệ thuật kinh doanh* (The Personal MBA: Master the Art of Business) (2010), là sự mở rộng của trang web PersonalMBA.com, và đã trở thành cuốn sách bán chạy nhất. Một phần của việc viết sách là phải xác định làm thế nào để truyền bá ngôn ngữ, vì thế sau nhiều năm, tôi đã làm việc chăm chỉ để thu hút được nhiều người đọc mới.

Kết quả là cuốn sách đã được đề cao trong tờ *New York Times*, *Wall Street Journal*, *Fortune*, *Forbes* và *FastCompany*, cũng như trên nhiều trang web blog nổi tiếng khác. Mỗi lần cuốn sách hay trang web của tôi được giới thiệu lại có hàng nghìn người ghé thăm trang PersonalMBA.com chỉ trong một khoảng thời gian ngắn.

## Cái giá của sự tiến bộ

Cùng lúc có hàng nghìn vị khách ghé thăm trang web của bạn là một điều tuyệt vời, với điều kiện là những vị khách đó có thể thực sự truy cập được vào trang web của bạn. Đó chính là vấn đề của tôi: bất cứ khi nào trang web của tôi được một lượng lớn khách ghé thăm, nó lại bị nghẽn mạng, chỉ để lại cho những vị khách đó một mẩu thông tin báo lỗi khó hiểu.

Và đây là một ví dụ điển hình: Lifestyler.com, một trang blog nổi tiếng, đã giới thiệu cuốn *MBA cá nhân* ba lần trong vòng bảy năm qua. Mỗi lần lại có hàng nghìn người cố gắng truy cập vào trang PersonalMBA.com cùng một lúc, làm nghẽn máy chủ (máy tính cung cấp trang web theo yêu cầu của khách ghé thăm) của trang web. Thay vì cung cấp thông tin được yêu cầu, máy chủ của tôi lại đưa ra thông điệp “Lỗi thiết lập kết nối với cơ sở dữ liệu” hoặc “Lỗi 503”, thông điệp số tương đương với việc máy chủ nài xin lòng khoan dung.

Mỗi lần máy chủ của tôi sập vì quá tải, một phần linh hồn của tôi cũng chết theo. Toàn bộ thời gian dành cho việc quảng bá trang web của tôi đã thành công cốc. Hàng nghìn độc giả tò mò cuối cùng đã chịu quan tâm tới điều tôi giới thiệu, nhưng vì máy chủ của tôi bị sập, họ lại phải quay trở ra cùng với tâm trạng thất vọng và trống rỗng. Công tác marketing của tôi cũng rất tốt, nhưng hệ thống của tôi lại không thể xử lý được nhu cầu bất ngờ.

## Kiểm tra vấn đề

Lúc đầu, giải pháp của tôi là tăng cường cho máy chủ bằng cách tăng năng lực xử lý và thêm bộ nhớ. Điều đó có tác dụng, nhưng chỉ tại một thời điểm nào đó. Ngoài thời điểm đó ra, trang web của tôi có thể bị nghẽn và sập bất cứ lúc nào – chuyện thường xảy ra cùng lúc với những lần marketing thắng lợi.

Khi đó, PersonalMBA.com đang chạy trên hệ thống quản lý trang web nổi tiếng là WordPress. WordPress được tối ưu hóa để dễ sử dụng và cài đặt, không

chạy dưới tải nặng. Theo cấu hình mặc định của WordPress, tất cả các trang web đều yêu cầu khởi động một tầng hoạt động của máy chủ, tạo ra hàng trăm yêu cầu ẩn nhằm cung cấp trang web tới người đọc.

Điều đó khiến cho yêu cầu đối với mỗi trang web trở nên “đắt đỏ”. Nghĩa là, mỗi yêu cầu lại đòi hỏi một lượng lớn bộ nhớ và năng lực xử lý mới hoàn thành được. Nếu một khách xem năm trang trên web, người khách đó đã tạo ra năm yêu cầu đắt. Nếu hàng nghìn khách cùng đưa ra yêu cầu vào cùng một thời điểm, máy chủ sẽ phải cố gắng thực hiện hàng nghìn quá trình xử lý cùng một lúc.<sup>101</sup>

## Hệ thống hỏng

Trong tình huống này, chiếc máy chủ tội nghiệp bị bao vây sẽ cố gắng đáp trả tất cả các yêu cầu, nhưng vì mỗi yêu cầu lại đòi hỏi rất nhiều nguồn khác nhau, máy chủ sẽ hết bộ nhớ trước khi các yêu cầu được thực hiện. Lúc đó, máy chủ giờ cò trống đầu hàng, còn khách là người kém may mắn.

Để giải quyết vấn đề này, tôi đã đổi công ty quản trị web năm lần, và đã dành hàng trăm giờ nghiên cứu làm thế nào để các máy chủ chạy trên nền WordPress có thể giữ mạng được trong tình trạng tải nặng. Mỗi cấu hình máy chủ mới lại tăng mức độ phức tạp, và mỗi lần cài đặt mới lại đòi hỏi phải liên tục bảo dưỡng nhiều hơn.

Cuối cùng tôi đã xây dựng cấu hình máy chủ riêng cho mình, viện dẫn một loạt các yêu cầu hệ thống phức tạp để cài đặt, định cấu hình và thay đổi các ứng dụng máy chủ phức tạp mà tôi chỉ hiểu vừa đủ. Mỗi một lỗi hoặc một vấn đề mà tôi gặp phải trong quá trình hình thành hoặc duy trì đều ngôn của tôi hàng giờ nghiên cứu và tìm cách xử lý.

Thế vẫn chưa hết, tính phổ biến và bảo mật kém của WordPress khiến nó trở thành món mồi ngon cho các tin tặc và những kẻ gửi thư rác. Cứ khoảng một tuần, một lập trình viên ẩn danh lại tìm thấy một vài lỗ hỏng mới, sau đó sử dụng chúng để kiểm soát tài khoản của người sử dụng hoặc gửi đầy thư rác vào hàng triệu blog ở WordPress.

Chỉ riêng việc bảo mật cho quá trình cài đặt WordPress và duy trì các cập nhật phần mềm hàng đầu cũng đủ là một việc làm toàn thời gian rồi, đặc biệt khi bạn duy trì nhiều trang web. (Lúc đó tôi duy trì cùng lúc 12 trang web và

đã thất bại một cách thảm hại.)

Đến một lúc, tôi chợt nhận ra mình đã dành nhiều thời gian hơn để giữ cho các trang web này luôn ở trạng thái trực tuyến thay vì nghiên cứu và viết cho các độc giả của mình đọc. Điều đó chẳng có ý nghĩa gì. Tôi không chỉ làm lãng phí khả năng của mình, mà còn không thực sự học cách lập trình. Tôi chỉ đang học một mớ những thứ liên quan tới một tình huống cụ thể và chỉ áp dụng cho việc chạy WordPress. Chẳng hay ho gì.

Tôi đã quyết định tìm cách khác để duy trì các trang web của mình, và không tốn nhiều thời gian lắm để tìm được một giải pháp thay thế đầy hứa hẹn.

## Giải pháp tiềm năng

Một hôm, tôi tình cờ đọc được một bài luận về Jekyll, chương trình quản lý web của Tom Preston-Werner (nổi tiếng với vai trò người sáng lập kho mã nguồn mở GitHub). Jekyll được thiết kế để thay thế cho những hệ thống như WordPress bằng cách khiến cho việc chạy web mà không cần phụ thuộc vào những yêu cầu đắt đỏ trở nên dễ dàng hơn.

Hãy tưởng tượng bạn có hàng trăm tài liệu xử lý chữ chứa thông tin quan trọng, và bạn cần phải biến tất cả chúng thành giống nhau – cùng phong chữ, cùng kiểu đề mục... Nếu bạn viết một chương trình có thể áp dụng một thiết kế trang web mà bạn chọn cho tất cả các tập tin một cách tự động (thay vì cập nhật từng tập tin bằng tay) thì bạn sẽ tiết kiệm được rất nhiều thời gian.

Đó chính xác là điều mà Jekyll làm được cho các trang web. Hãy chạy một lệnh đơn, và Jekyll sẽ tạo ra một trang web sử dụng các tập tin trong máy tính có chứa thông tin trang web và mẫu thiết kế. Nếu bạn thay đổi thiết kế hoặc nội dung trang, bạn chỉ cần chạy lại Jekyll, toàn bộ trang web sẽ được cập nhật các thay đổi một cách tự động, giúp bạn tiết kiệm được rất nhiều sức lực.

Jekyll đem lại một cơ hội đầy hứa hẹn. Về mặt lý thuyết, tôi có thể thay thế WordPress chỉ với một thư mục gồm các tập tin văn bản đơn giản trong máy tính của mình. Trang web của tôi có thể mở rất nhanh, ổn định và tôi sẽ tiết kiệm được cho mình hàng trăm giờ bảo dưỡng máy chủ mỗi năm.

Tuy nhiên, vẫn có một vấn đề: Jekyll được viết bằng Ruby, ngôn ngữ lập trình mà tôi không biết. Tôi không có chút ý niệm nào về việc viết mã bằng Ruby hay chạy các ứng dụng Ruby phục vụ những người dùng thực sự. Một số

việc tôi cần phải làm để chạy PersonalMBA.com đòi hỏi nhiều hơn là việc định dạng đơn giản.

Để chạy các trang web sử dụng Jekyll của mình, tôi phải học cách lập trình và triển khai các ứng dụng web chạy bằng Ruby.

## Học cách viết mã

Tôi muốn dành một khoảng thời gian để học cách lập trình, nhưng những dự án khác luôn chiếm vị trí ưu tiên. Nếu tôi có thể học cách viết mã, cơ hội kinh doanh và xuất bản của tôi sẽ được mở rộng đáng kể, vì mọi thứ tôi làm để xây dựng doanh nghiệp đều được chạy trên web.

Có một điểm quan trọng cần phải lưu ý là tất cả mọi việc tôi làm cho tới thời điểm này đều không phải là lập trình. HTML và CSS, hai ngôn ngữ tôi sử dụng để xây dựng các trang web đều được gọi là ngôn ngữ “đánh dấu” (markup). Mã HTML và CSS không được thông minh cho lắm, nó chỉ yêu cầu máy tính thể hiện cho người sử dụng một tập tin văn bản theo một cách cụ thể nào đó (ví dụ như “bôi đen đoạn văn bản này” hay “đề mục của phần này được để ở cỡ chữ 24”).

Việc hình thành máy chủ của tôi cũng như vậy. Dù tôi sắp xếp các phần mềm lại với nhau, thì cũng không phải là tôi đang lập trình thực sự. Thay vào đó, tôi chỉ cài đặt các chương trình được viết sẵn, sau đó thay đổi một vài cài đặt. Lập trình viên viết ra các ứng dụng mà tôi đang sử dụng, nhưng tôi không cần phải có kiến thức về lập trình để sử dụng được chúng. Lập cấu hình máy chủ và quản lý là kỹ năng hữu dụng, nhưng không phải lập trình.

## Vậy “lập trình” là gì?

Suy nghĩ đầu tiên của tôi: Lập trình là bắt máy tính làm việc, nhưng suy nghĩ đó chưa cụ thể và có ích. Làm gì? “Việc” ở đây là việc gì?

Mười năm trước, tôi đã theo học hai khóa lập trình cơ bản ở trường cao đẳng, thế nên việc không thể đưa ra được một định nghĩa hữu dụng khiến tôi thấy hơi xấu hổ. Thành thật là thế. Tôi nhớ được một vài thuật ngữ cơ bản như *biến số (variable)*, *vòng lặp (loop)*, *dữ liệu vào (input)*, *dữ liệu ra (output)*, *chức năng (function)*, *hướng đối tượng (object orientation)* và một khái niệm

gì đó gọi là *sắp xếp nổi bọt* (*bubble sort*) nhưng không biết gì hơn nữa.

Yêu cầu của khóa học là học được ngôn ngữ lập trình C++, và tôi nhớ là mình đã rất chán nản khi mất hàng tiếng đồng hồ để tìm một dấu chấm phẩy bị thiếu khiến cả chương trình của tôi bị tê liệt. Tôi còn nhớ thầy giáo của tôi đã nói câu gì đó đại khái là: “Bạn sẽ không bao giờ sử dụng tới sắp xếp nổi bọt cho một ứng dụng thực sự nào cả, nhưng dù sao chúng ta vẫn cứ phải học”.

Tôi đã viết một số chương trình cơ bản cho khóa học vì tôi buộc phải viết: mục tiêu là phải vượt qua khóa học với xếp loại giỏi, và tôi đã làm được. Đáng tiếc là những chương trình mà chúng tôi viết khi đó không có giá trị sử dụng bên ngoài lớp học, và tôi chưa từng sử dụng những khái niệm đó trong một thời gian dài. Dù tôi vẫn nhớ được một số từ, nhưng tôi vẫn phải bắt đầu lại với việc tìm hiểu các ý tưởng chính.

Vì tôi không thể định nghĩa được việc tôi đang làm là gì khi nói tôi muốn “học lập trình”, nên tôi gặp khó khăn trong việc xác định mức độ thành thạo mục tiêu. “Tạo ra một chương trình máy tính” có vẻ cụ thể hơn, nhưng vẫn chưa hữu ích lắm.

Đã đến lúc để giới thiệu tôi đã biết gì về lập trình:

Tôi biết các lập trình viên là người “viết” các chương trình, điều đó có nghĩa là đó là một công việc sáng tạo, có thể được hoàn thành bằng nhiều cách khác nhau.

Các chương trình thường được gọi là “ứng dụng”, và hai từ này có thể dùng để thay đổi cho nhau được.

Khi các chương trình máy tính được “chạy” hoặc “thực hiện”, chúng sẽ tiến hành những việc mà chúng được viết để thực hiện, bất kể việc đó là gì.

“Dữ liệu vào” và “dữ liệu ra” là hai khái niệm khá dễ nhớ, vì chúng hay được sử dụng. Dữ liệu vào là thông tin hoặc dữ liệu mà chương trình sử dụng, và dữ liệu ra là điều bạn nhận được khi chương trình chạy xong.

“Biến số” về cơ bản là trình giữ chỗ cho một thứ gì đó thay đổi. Bạn có thể tạo ra bao nhiêu biến số tùy thích, và khiến những biến số này tượng trưng cho bất cứ điều gì bạn muốn.



Bản thân “chương trình” là một bộ chỉ lệnh chi tiết và các quy tắc, nói cho máy tính biết chính xác phải làm gì với dữ liệu vào. Khi chương trình chạy xong, nó sẽ cho bạn dữ liệu ra.

Chương trình “hỏng” hoặc hiển thị thông báo lỗi khi có điều gì đó không ổn hoặc máy tính không thể xác định được cần phải làm gì tiếp theo.

Giờ chúng ta bắt đầu đến một nơi nào đó. Và đây là một bước “tháo dỡ” rất cơ bản: Thay vì “lập trình”, giờ chúng ta có ba khái niệm nhỏ hơn cần phải xử lý:

Dữ liệu vào – thông tin bạn sử dụng để thực hiện một quá trình.

Quá trình – một loạt các bước mà chương trình thực hiện với dữ liệu vào.

Dữ liệu ra – kết quả cuối cùng của chương trình.

Sự phân chia này hữu ích hơn nhiều. “Viết một chương trình máy tính” có nghĩa là xác định bạn bắt đầu với thông tin gì, các bước miêu tả chính xác điều mà máy tính sẽ thực hiện với những dữ liệu đầu vào đó, và xác định đầu ra mà máy tính sẽ trả lại khi chương trình hoàn thành quá trình thực hiện.

Hãy nghĩ về biểu đồ tiến trình – thứ có vẻ là hình ảnh tinh thần hữu dụng cho thấy cách các chương trình vận hành. Bạn bắt đầu quá trình với một vài dữ liệu đầu vào cụ thể. Suốt quá trình, bạn thực hiện những hành động cụ thể khi các điều kiện cụ thể đúng hoặc sai. Quá trình kết thúc khi bạn đến được điểm cuối của biểu đồ tiến trình, và bạn có được dữ liệu đầu ra: Kết quả cuối cùng của quá trình hoàn thiện mà biểu đồ tiến trình miêu tả.

Tạo ra một chương trình máy tính dường như là một cách khác để thực hiện quá trình suy nghĩ mà bạn vẫn làm khi tạo ra một biểu đồ tiến trình. Bạn sẽ hỏi những câu hỏi giống nhau và giống như thế này:

Mình bắt đầu với cái gì?

Chuyện gì sẽ xảy ra lúc bắt đầu quá trình?

Chuyện gì sẽ xảy ra sau đó? Và sau đó nữa?

Khi nào quá trình kết thúc?

Mình có được gì khi quá trình kết thúc?

Biểu đồ tiến trình miêu tả câu trả lời cho các câu hỏi này dưới dạng hình ảnh, còn các chương trình miêu tả chúng bằng cách sử dụng câu chữ, nhưng quá trình suy nghĩ thì giống hệt nhau.

Sự tương đồng về biểu đồ tiến trình có ích vì nó đưa ra một vài gợi ý về những khái niệm có thể quan trọng khác.

*Điều kiện* là những mệnh đề kiểu như:

“Nếu X đúng/sai thì thực hiện Y”

“Nếu X là/không phải là Y thì thực hiện Z”

“Khi X đúng/sai thì thực hiện Y”

“Khi X là Y thì thực hiện Z”

“Trong khi X đúng/sai thì thực hiện Y”

“Trong khi X là Y thì thực hiện Z”

X, Y và Z trong trường hợp này là các *biến số*, nghĩa là nó có thể tượng trưng cho bất cứ thứ gì. Biến số có thể tượng trưng cho các con số, giống như trong đại số cơ bản, hoặc chúng có thể tượng trưng cho các từ. Đôi khi biến số chỉ là các chữ cái hoặc biểu tượng, và đôi khi chúng lại là các từ. Dù là gì thì chúng cũng thể hiện bất cứ điều gì bạn đang làm việc cùng.

Mệnh đề điều kiện (phần NẾU... THÌ..., KHI... THÌ...) giống như các mũi tên có câu hỏi trong biểu đồ tiến trình. Hãy nghĩ về việc lái ô tô: NẾU đèn giao thông chuyển sang màu đỏ, THÌ dừng lại. KHI đèn giao thông chuyển sang màu xanh, THÌ đi. KHI đèn giao thông có màu vàng, giảm tốc độ và chuẩn bị

dừng lại.

Những mệnh đề điều kiện này đáng được đào sâu hơn chút nữa, vì có một số mô hình phổ biến. Đúng/sai có vẻ xuất hiện nhiều, và TRONG KHI dường như ám chỉ cho chúng ta biết chúng ta nên tiếp tục làm điều gì đó thay vì chỉ thực hiện một hành động.

Trong tất cả các trường hợp, mệnh đề điều kiện có chứa tuyên bố xác định nên hay không nên thực hiện hành động. Tuyên bố này được gọi là một *điều kiện*, và nó có thể tồn tại dưới nhiều dạng. Đôi khi điều kiện có dạng cơ bản là so sánh đúng/sai (Đèn đang đỏ à?), và đôi khi nó lại là phép so sánh toán học (X lớn hơn 100 à?), và đôi khi nó hàm chứa cả sự suy luận logic (Đèn báo hiệu dừng KHÔNG đỏ à?).

Mục đích của điều kiện là xác định liệu quá trình liên quan có nên xảy ra hay không. Nếu điều kiện đúng hoặc hợp lý, chương trình sẽ thực hiện chỉ lệnh. Nếu không, chỉ lệnh có liên quan sẽ bị bỏ qua, và chương trình sẽ chuyển sang chỉ lệnh tiếp theo.

Biến số đúng/sai được gọi là *biến số Boolean*, thuật ngữ khá phổ biến dùng để chỉ một việc gì đó chỉ có hai lựa chọn. Có/không và bật/tắt cũng là các biến số Boolean. Biến số Boolean rất quan trọng trong các chương trình máy tính vì chúng là đơn vị cơ bản của cả những quá trình đơn giản (như đúng/sai trong biểu đồ tiến trình) và việc bật/tắt nút công tắc điện nhỏ xíu ở góc máy tính.

TRONG KHI trong trường hợp này là một kiểu điều kiện đặc biệt được gọi là *vòng lặp*. Vòng lặp khiến cho quá trình đang được yêu cầu lặp lại cho tới khi điều kiện được đáp ứng. Trở lại với ví dụ lái xe: TRONG KHI đèn giao thông đang đỏ, không được đi.

Rất đơn giản, đúng không? Nếu chúng ta nghĩ về việc viết một chương trình máy tính giống như vẽ một biểu đồ tiến trình thì quá trình cơ bản sẽ dễ hình dung hơn rất nhiều.

Chỉ còn một vấn đề nhỏ cuối cùng: Chuyện gì sẽ xảy ra nếu máy tính bị kẹt, không biết phải làm gì, hoặc chỉ lệnh mà máy tính đang cố gắng thực hiện không có nghĩa hoặc không thực hiện được? Chuyện gì sẽ xảy ra?

Chương trình “tê liệt” – chương trình dừng hoàn toàn và thường đưa ra một thông điệp báo lỗi thay vì kết quả như mong muốn. Tất cả chúng ta đều đã chứng kiến lỗi Windows đáng sợ “màn hình xanh lè” hoặc “Lỗi 404: không tìm

thấy trang web” trong khi duyệt web. Có điều gì đó bất ngờ xảy ra, máy tính “bối rối” và chương trình tê liệt.

Là một lập trình viên máy tính, công việc của bạn là ngăn chặn tình trạng tê liệt và lỗi sai, không để chúng xảy ra. Cách tốt nhất để thực hiện điều đó là đảm bảo chương trình luôn có được thông tin nó cần để hoàn thành quá trình như dự kiến, nhưng điều đó không phải lúc nào cũng là điều có thể. Trong những trường hợp mà sự không chắc chắn là điều khó tránh khỏi, sẽ có ích khi có một cách để khôi phục khi chương trình thất bại trong khi cố gắng hoàn thành một quá trình nào đó.

Những câu lệnh phục hồi lỗi này được gọi là *ngoại lệ*, và chúng rất hữu ích. Bạn có thể nghĩ về chúng như những câu điều kiện lỗi-đặc biệt: NẾU chương trình chuẩn bị hỏng theo cách X, THÌ thực hiện Y thay vì để (nó) hỏng.

Ngoại lệ gần giống như thêm máy phát dự phòng cho tòa nhà bệnh viện. Hầu hết thời gian, máy phát chỉ để đó không sử dụng. Tuy nhiên, khi mất điện, máy phát khởi động, và bệnh viện sử dụng điện từ máy phát thay vì chìm trong bóng tối. Đó là một thiết bị hữu ích nếu trong bệnh viện đang có những bệnh nhân phải sử dụng những máy móc hỗ trợ sự sống chạy bằng điện. Thất bại hoàn toàn rất nguy hiểm, do đó, kế hoạch dự phòng là điều cần thiết.

Đó chính là lập trình cơ bản. Xác định dữ liệu đầu vào. Đặt biến số. Tạo ra những quá trình dẫn tới đầu ra mong muốn. Hãy nghĩ về các quá trình này như một biểu đồ tiến trình, thêm điều kiện và ngoại lệ nếu cần thiết. Nếu mọi chuyện tốt đẹp, bạn cung cấp dữ liệu đầu vào, chạy chương trình và nhận được kết quả mong muốn.

Đây là sự đơn giản hóa tối đa một hoạt động phức tạp, nhưng nó đủ chi tiết để trở nên hữu ích cho những người mới có thể lập trình. Bằng cách chia nhỏ lập trình theo cách này, sẽ dễ hơn để bạn biết phải bắt đầu từ đâu.

## Ngôn ngữ lập trình

Đây là phần thủ thuật: Máy tính không nói cùng thứ ngôn ngữ với con người. Về cơ bản, máy tính làm việc bằng cách bật và tắt công tắc điện nhỏ xíu theo những cách vô cùng đặc biệt. Nếu máy tính không có cách dịch yêu cầu theo ngôn ngữ của con người thành việc bật công tắc điện, máy tính không thể thực hiện được điều chúng ta yêu cầu.

Đó chính là điều mà ngôn ngữ lập trình thực hiện: Chúng đưa cho người lập trình một cách cụ thể để nói cho máy tính biết khi nào bắt đầu, cần phải làm gì và khi nào thì dừng lại. Chúng cũng cho phép người lập trình xác định thế nào là dữ liệu đầu vào, quá trình, dữ liệu đầu ra, và cần phải làm gì khi chương trình chạy xong.

Mỗi ngôn ngữ lập trình có một cách riêng để viết yêu cầu, gọi là *cú pháp* (syntax). Cú pháp của ngôn ngữ có chứa các quy tắc mà máy tính sử dụng để dịch các yêu cầu thành các tín hiệu bật-tắt điện tử nhỏ xíu.

Mỗi ngôn ngữ lại có một cách để xác định biến số, điều kiện và ngoại lệ riêng. Chi tiết có thể khác nhau, nhưng khái niệm cốt lõi thì giống nhau.

## Nghĩ như một lập trình viên

Lập trình viên thường nghĩ về vấn đề theo cách được gọi là *giải mã* (pseudocode): ngôn ngữ giống như mật mã, nhưng lại không đủ cụ thể để máy tính có thể thực hiện được. Hãy nghĩ về nó như một dạng phác họa. Giải mã giúp bạn nghĩ thông qua quá trình giải quyết vấn đề.

Ví dụ về việc lái xe mà tôi đã sử dụng chính là một dẫn chứng về giải mã. Nếu tôi bước vào xe và nói to : “NẾU tôi xoay chìa khóa THÌ động cơ sẽ khởi động”, thì sẽ chẳng có chuyện gì xảy ra cả. Điều đó không có nghĩa là câu nói đó vô dụng. Mệnh đề đó chỉ là một cách nghĩ thông qua các bước cần thiết để giải quyết một vấn đề hoặc nhận được một kết quả mong muốn.

Bạn có thể sử dụng những khái niệm lập trình cơ bản mà chúng ta đã thảo luận để phác thảo một vài chương trình đơn giản cho những nhiệm vụ phổ thông.

Có một cách thú vị để tự mình thử nghiệm điều đó: Hãy tìm một người bạn, và nhờ người đó giúp bạn hoàn thành một nhiệm vụ đơn giản như “làm bánh sandwich”. Quy tắc duy nhất là người bạn đó của bạn chỉ được làm chính xác những gì bạn bảo anh ta làm, không hơn, không kém. Anh ta không được học bất cứ thứ gì, và chỉ được thực hiện mọi yêu cầu theo đúng nghĩa đen của nó.

Chỉ vài phút, bạn sẽ thấy mình có một cuộc hội thoại như thế này:

*Bạn:* Nhắc bánh mì lên.

*Người bạn:* Mình không hiểu từ “nhấc lên”.

*Bạn:* Di chuyển tay bạn tới chỗ mà mình nói và nắm lấy nó.

*Người bạn:* Mình không hiểu từ “tay”.

*Bạn:* (thở dài) Nó đây này. [Bạn chỉ vào tay của người bạn.]

*Người bạn:* Hiểu rồi.

*Bạn:* Di chuyển tay bạn tới chỗ mà mình nói và nắm lấy nó.

*Người bạn:* Mình không hiểu từ “nắm”.

*Bạn:* Gập ngón tay của bạn như thế này này. [Bạn làm mẫu, gập và thả các ngón tay ra.]

*Người bạn:* Hiểu rồi.

*Bạn:* Di chuyển tay tới chiếc bánh mì và nắm lấy nó.

*Người bạn:* Mình không hiểu từ “bánh mì”.

*Bạn:* Chính là thứ này này! [Bạn chỉ vào chiếc bánh mì.]

*Người bạn:* Hiểu rồi.

*Bạn:* Di chuyển tay tới chiếc bánh mì và nắm lấy nó.

*Người bạn:* [Di chuyển tay tới chiếc bánh mì, gập ngón tay và nhả ra. Chiếc bánh mì không xô dịch.]

*Bạn:* ĐÂY LÀ MỘT TRÒ CHƠI NGÓ NGÂN!

Đó là một ví dụ ngó ngân, nhưng lại gần giống với việc lập trình như thế nào và có cảm giác ra sao, đặc biệt là ở giai đoạn đầu.

Chiếc máy tính giống như người bạn của bạn, không hiểu được bất cứ điều gì bạn yêu cầu một cách dứt khoát. Bất cứ quá trình phức tạp nào mà bạn cố gắng định nghĩa cần phải được chỉ rõ, không mập mờ.

Đây chính là điều khiến lập trình trở nên khó khăn: Chỉ một sự mơ hồ nhỏ hoặc một yêu cầu không chính xác cũng có thể khiến toàn bộ chương trình bị hỏng. Lập trình là một nghề thủ công không rộng lượng xét ở góc độ đó: Đoạn

mã của bạn hoặc là đúng hoặc sai, và cần phải được thể hiện bằng chi tiết chính xác.

Máy tính không ấn tượng với sự quyến rũ hay hóm hỉnh. Nếu đoạn mã của bạn chưa hoàn thiện, hoặc viết không tốt, thì ứng dụng của bạn sẽ bị hỏng, bạn sẽ mất dữ liệu, hoặc bạn sẽ tạo ra một *lỗi* (bug): Một mẫu mã độc tạo ra những hệ quả không mong đợi hoặc không dự đoán được. Giống như toán học, logic hoặc phát huy tác dụng hoặc không. Không có điểm A nào cho nỗ lực cả.

Như đã nói, trong lập trình không có một giải pháp duy nhất cho một vấn đề cụ thể nào đó giống như trong toán học. Có hàng nghìn cách để tạo ra kết quả mong muốn với lượng dữ liệu đầu vào cụ thể. Là một lập trình viên, bạn có thể chọn cách tiếp cận dựa trên những công cụ mà bạn có.

Khi bạn đã quen với việc máy tính không thể đọc được ý nghĩ của bạn, bạn sẽ học cách bắt đầu đưa ra những yêu cầu mà hệ thống có thể hiểu được, theo đúng cách mà bạn đã học để định nghĩa các thuật ngữ và làm mẫu các hành động cơ bản cho người bạn của bạn trong ví dụ “làm bánh sandwich” trước khi đưa ra những yêu cầu phức tạp.

## Điều gì khiến lập trình ứng dụng web khác biệt?

Lúc này chúng ta đã có một định nghĩa về lập trình. Nó đã được đơn giản hóa, nhưng vẫn đủ để hiểu điều chúng ta đang cố thực hiện lúc này.

Tôi không quan tâm tới tất cả các loại lập trình, tôi chỉ muốn viết các chương trình có thể chạy trên các trang web. Nếu bạn đã từng sử dụng những chương trình gửi thư dựa trên web như Gmail, Hotmail, Yahoo Mail hoặc đại loại như vậy, bạn sẽ biết tôi đang nói về điều gì. Những chương trình này chạy trên trình duyệt internet của bạn. Bạn không cần phải tải chương trình phần mềm về máy tính để sử dụng. Bạn chỉ cần chỉ trình duyệt của bạn tới trang web đó, đăng nhập và bạn đã có thể sử dụng.

Sự khác biệt giữa phần mềm chạy *tại chỗ* và phần mềm chạy *từ xa* trên máy chủ web là một sự khác biệt quan trọng. Để phát triển một ứng dụng web, trước tiên bạn cần viết một ứng dụng, sau đó thử nghiệm để xem nó có hoạt động không. Tất cả việc phát triển và thử nghiệm đó đều diễn ra trên máy tính của riêng bạn.

Khi chương trình đã hoạt động, bạn có thể gửi nó tới một máy chủ web “sản

xuất” để những người khác cũng có thể truy cập được. Mọi người trên internet không thể truy cập vào máy tính cá nhân của bạn từ trang web, nên việc đăng tải phần mềm lên một máy chủ web được truy cập thoải mái là bước cần thiết để mọi người có thể sử dụng được chương trình của bạn.

Điều đó có nghĩa là quá trình phát triển gồm hai giai đoạn quan trọng: lập trình và thử nghiệm, sau đó đẩy chương trình hoàn thiện lên máy chủ sản xuất từ xa để sử dụng thực sự. Tôi cần phải xác định hai giai đoạn này hoạt động như thế nào.

Từ kinh nghiệm với HTML và CSS, tôi còn biết thêm một điều: cả hai đều “ngu ngốc”, với chúng, bạn không thể yêu cầu một trang web cơ bản chứa được bất cứ thông tin gì.

Giả sử bạn có một tập tin trang web với nội dung “Xin chào, thế giới!” và bạn muốn thay từ “thế giới” bằng tên của một người truy cập trang web. Đó là một suy nghĩ thú vị, nhưng các trang web cơ bản không có cách nào để phân loại thông tin để sau này sử dụng lại. Chúng chỉ hiển thị văn bản trong tập tin, và tập tin cũng không được phép tự cập nhật.

Thuật ngữ kỹ thuật chỉ hiện tượng này là *trạng thái* (state). Các trang web để cơ bản được xây dựng bằng cách sử dụng HTML và CSS không có trạng thái, vì thế chúng được coi là các nguồn “không có trạng thái”. Bạn có thể thêm một mẫu nhỏ cho trang web hỏi tên người sử dụng với một nút Lưu ở bên cạnh, nhưng cái nút này không thực sự lưu bất cứ thông tin nào trừ khi bạn tạo được một chỗ riêng để chứa các thông tin đó.

Đó chính là lý do tại sao các ứng dụng web lại sử dụng hai cách tiếp cận phổ biến để lưu trữ dữ liệu cho sau này: *Cơ sở dữ liệu* (database) và *bộ nhớ nhỏ* (cookie<sup>(1)</sup>).

Cách tốt nhất để hiểu cơ sở dữ liệu làm gì là tưởng tượng một chồng thẻ ghi chú. Giả sử bạn đang cố gắng tạo ra một cuốn sổ địa chỉ, và bạn muốn ghi tên, số điện thoại, địa chỉ e-mail, giới tính và tuổi của mỗi người bạn của bạn.

Mỗi người bạn có một chiếc thẻ ghi chú riêng, và thông tin của họ được ghi trên tấm thẻ đó. Nếu một trong số họ thay đổi địa chỉ e-mail, giả sử như vậy, bạn có thể xóa thông tin cũ đi và cập nhật tấm thẻ với thông tin mới. Khi bạn nhìn vào tấm thẻ của một người bạn nào đó, bạn có thể nhìn thấy tất cả các thông tin của họ cùng một lúc.



Bạn có thể nghĩ về cơ sở dữ liệu như một chồng các tấm thẻ ghi chú đó. Mỗi tấm thẻ trong chồng đó được gọi là *một bản ghi* (record<sup>(2)</sup>). Bạn có thể có bao nhiêu bản ghi trong cơ sở dữ liệu tùy thích, nhưng đến một lúc nào đó, chồng thẻ sẽ trở nên khó quản lý. Thường sẽ hiệu quả hơn nếu chia chồng thẻ đó thành những chồng nhỏ hơn, ít hơn: gia đình và bạn bè vào một chồng, đồng nghiệp ở một chồng khác, chẳng hạn như vậy.

Điều đó có rõ ràng không? Và đây mới là điều thú vị: Hãy tưởng tượng chồng thẻ ghi nhớ của bạn lúc này là một chồng thẻ ma thuật. Bạn có thể nói chuyện với thẻ ma thuật và yêu cầu nó cho bạn xem những tấm thẻ phù hợp với một số điều kiện cụ thể, chẳng hạn như:

“Hãy cho ta xem thẻ của John Smith”.

“Hãy cho ta xem tất cả các thẻ có giới tính là nữ”.

“Hãy cho ta xem tất cả các thẻ có tuổi lớn hơn 50”.

Khá hữu ích, đúng không? Về cơ bản, đó chính là điều mà cơ sở dữ liệu thực hiện: Chúng cung cấp cho bạn một cách để lưu trữ các thông tin đã được cơ cấu, cũng như cách để lấy lại thông tin đó theo bất cứ cách nào mà bạn muốn.

Mỗi một mẫu dữ liệu mà bạn đặt vào một tấm thẻ được gọi là *trường dữ liệu* (field). Trường dữ liệu trong cơ sở dữ liệu của bạn càng nhiều thì bạn càng có nhiều cách để lấy lại dữ liệu khi bạn muốn.

Cơ sở dữ liệu là cách phổ biến nhất để lưu trữ dữ liệu trên ứng dụng web. Nếu bạn muốn lưu giữ thông tin như tên người sử dụng, địa chỉ e-mail và các thông tin khác thì sử dụng cơ sở dữ liệu là hợp lý nhất. Một khi John Smith đăng nhập vào ứng dụng của bạn, bạn có thể để ứng dụng của bạn lấy được tên qua bản ghi cơ sở dữ liệu của John, sau đó hiển thị “Xin chào, John Smith!”

Một cách phổ biến khác để lưu giữ thông tin trên ứng dụng web là sử dụng cookie: Một đoạn văn bản rất nhỏ được lưu trên máy tính của người sử dụng. Cookie đặc biệt hữu dụng khi lưu giữ một lượng nhỏ dữ liệu vốn không cần tồn tại trong một thời gian dài.

Trong trường hợp chương trình sở địa chỉ, sẽ thích hợp khi lưu giữ một cookie khi John Smith đăng nhập. Tập tin cookie về John sẽ chứa các thông tin

như tên đăng nhập (username) = johnsmith và đăng nhập (loggedin) = đúng. Nếu John Smith thoát ra khỏi ứng dụng, nhưng sau đó quay trở lại, thì ứng dụng có thể nhận ra đoạn cookie và cho phép anh ta truy cập mà không yêu cầu phải đăng nhập lại. Có thể đặt để cookie hết hạn trong một khoảng thời gian cụ thể nào đó, khiến nó trở nên có ích cho kiểu lập trình này. (Nếu bạn đã từng nhìn thấy dòng chữ “Nhớ mật khẩu này” trên một trang web thì đó chính là cách cookie hoạt động.)

Điều mà chúng ta đang thực hiện lúc này chính là phân chia. Đây không phải danh sách toàn diện những điều khiến lập trình web trở nên độc đáo, nhưng nó lại đủ hoàn chỉnh để cung cấp một bộ khung đơn giản về điều tôi cần phải học: biến số, điều kiện, ngoại lệ, môi trường cục bộ/sản xuất, cơ sở dữ liệu và cookie.

Bạn có thấy việc chia nhỏ này có ích không? Tôi đã bắt đầu với một khái niệm mơ hồ về điều tôi muốn làm, và giờ tôi đã có một danh sách những kỹ năng nhỏ quan trọng cần phải học.

Nhưng tôi vẫn chưa sẵn sàng để học đâu. Bạn có nhớ tôi đã đề cập đến việc máy tính không hiểu được ngôn ngữ của con người không? Tôi cần phải chọn được một ngôn ngữ lập trình để viết các chỉ lệnh cho chương trình của mình, mà việc này đòi hỏi phải nghiên cứu thêm một chút.

## Chọn ngôn ngữ để lập trình ứng dụng web

Có hàng nghìn ngôn ngữ lập trình khác nhau, và ngôn ngữ mới vẫn được sáng tạo hàng ngày. Cú pháp của mỗi ngôn ngữ lại khác nhau và chịu ảnh hưởng lớn của việc ngôn ngữ đó được thiết kế để hoàn thành. Có một số ngôn ngữ tối ưu cho một số nhiệm vụ cụ thể hơn so với một số ngôn ngữ khác.

Trước khi bắt tay vào thực hiện, tôi đã quyết định dành một tiếng mò mẫm các trang web lập trình lớn để xem ngôn ngữ nào được các nhà phát triển ứng dụng web khuyến khích. Công việc nghiên cứu ban đầu này sẽ giúp tôi quyết định cần phải học ngôn ngữ nào và luyện tập kỹ năng ban đầu nào.

Hai trong số những điểm đến phổ biến nhất của các nhà lập trình là Stack Overflow và Hacker News, vì thế tôi bắt đầu bằng cách tìm kiếm lời khuyên về ngôn ngữ tốt nhất để học.

Stack Overflow là một trang web hỏi-và-đáp. Đó là nơi để hỏi những câu

kiểu như: “Làm thế nào tôi có thể thực hiện được X?” Những người lập mã có kiến thức và kinh nghiệm hơn sẽ trả lời những câu hỏi đó bằng những gợi ý, cách tiếp cận hoặc sửa lỗi cụ thể, khiến cho Stack Overflow trở thành điểm đến tốt nhất khi bạn muốn tìm kiếm sự giúp đỡ để xử lý những vấn đề lập trình.

Hacker News là trang web tin tức xã hội: Một bộ sưu tập các đường dẫn (link) đến các cuộc thảo luận có liên quan. Các chủ đề trên Hacker News thay đổi theo từng phút, nhưng thường xoay quanh những phát triển mới trong lập trình, công nghệ và kinh doanh, khiến cho Hacker News trở thành một địa chỉ lý tưởng để tìm kiếm các ý kiến, ít nhất là bán chuyên về những phát triển mới trong lập trình.

Ngôn ngữ, thư viện và kỹ thuật mới được các nhà lập trình trên khắp thế giới tạo ra hàng ngày. Một vài sự kết hợp giữa công nghệ và cách tiếp cận khá hữu ích đối với một số vấn đề, nhưng có những sự kết hợp lại không được như vậy. Thường thì bạn không thể biết được cho tới khi bạn thử.

“Tốt nhất”, theo thuật ngữ lập trình, có liên quan tới vấn đề mà bạn đang cố gắng xử lý và tới những ưu tiên cụ thể của bạn. Nhìn chung, lời khuyên là (1) chọn công cụ giúp bạn xử lý vấn đề hiệu quả và (2) nếu bạn được lựa chọn, hãy lựa chọn công cụ mà bạn thấy thích sử dụng. Chỉ thế là đủ.

Việc duyệt dữ liệu của Stack Overflow và Hacker News khiến tôi rơi vào tình trạng quá tải thông tin: Có quá nhiều thông tin để xử lý cùng một lúc, đặc biệt là khi bạn không quen với công nghệ. Tôi cần phải giảm bớt các tạp nhiễu nếu muốn tìm thấy lời khuyên cụ thể hơn.

Và đây là một thủ thuật nghiên cứu mà hầu hết mọi người không biết: Những cỗ máy tìm kiếm thông thường cho phép bạn giới hạn sự tìm kiếm của bạn trong một trang web cụ thể thay vì dàn trải ở tất cả các trang. “Mật mã” để làm được việc này trong Google như sau:

“cụm từ cần tìm” trang: (vidu).com

Hãy thay thế “cụm từ cần tìm” bằng cụm từ mà bạn đang tìm kiếm, và “vidu.com” bằng trang web mà bạn muốn tìm kiếm. Dấu ngoặc kép có nghĩa là tìm kiếm chính xác cụm từ đó. Không có dấu ngoặc kép, Google sẽ cho ra các kết quả là các trang có chứa tất cả các từ trong cụm từ đó, nhưng không hẳn theo đúng thứ tự bạn muốn.

Bằng cách sử dụng thủ thuật này, tôi đã tìm được một vài biến thể của cụm

từ “lập trình ứng dụng web”, “học cách viết mã” và “lập trình cho người mới bắt đầu”, sau đó dành thêm một giờ nữa hoặc để đọc các kết quả.

Đây là điều tôi đã tìm thấy: Những nhà phát triển web dày dạn kinh nghiệm gợi ý nên bắt đầu với một trong hai ngôn ngữ phổ biến là Ruby hoặc Python. Ruby và Python khá dễ học, và có thể cho bạn một nền tảng vững chắc về các khái niệm lập trình quan trọng. Ruby phổ biến với các nhà lập trình quan tâm tới các ứng dụng web, còn Python lại phổ biến hơn với các nhà khoa học và toán học, do nó sở hữu nhiều thư viện toán học, khoa học và đồ họa.

Ruby và Python, mỗi một ngôn ngữ lại có những cộng đồng nhà phát triển năng động, những nguồn tài liệu và sách miễn phí, những chương trình và công cụ có sẵn khiến cho các tính năng quan trọng dễ thực hiện hơn. Chọn ngôn ngữ nào để sử dụng xem ra chủ yếu lại là vấn đề tham khảo.

Sau khi đọc một vài ví dụ về mã được viết bằng mỗi loại ngôn ngữ này, tôi đã quyết định sẽ học Ruby. Dưới con mắt chưa qua đào tạo của tôi, mã Ruby có vẻ sạch, dễ đọc và tương đối dễ hiểu. Vì phần lớn những ý tưởng và thủ thuật mà tôi học sử dụng Ruby sẽ chuyển sang những ngôn ngữ khác mà có thể sau này tôi sẽ học, nên tôi cần phải làm sao để quá trình học thú vị ngay từ ban đầu.

Ngoài ra, có rất nhiều chương trình và công cụ mà tôi muốn sử dụng để hỗ trợ hoặc để học được ngôn ngữ Ruby. Đặc biệt, Jekyll được viết bằng Ruby, nên học Ruby sẽ giúp tôi giải quyết được một vấn đề nan giải. Python cũng có những công cụ tương tự, nhưng có vẻ phức tạp, khó sử dụng.

## Chọn một bộ khung

Ngoài việc đưa ra lời khuyên về ngôn ngữ lập trình, các nhà lập trình ứng dụng web còn có ý kiến gay gắt về *bộ khung (frameworks)*: Thư viện về mã giúp cho việc thực hiện những việc mà hầu hết các ứng dụng đều cần trở nên dễ dàng hơn.

Thư viện rất quan trọng vì máy tính thực hiện chính xác những gì bạn yêu cầu chúng làm. Không hơn, không kém.

Đó là một thủ thuật, vì đoạn mã mà bạn cung cấp là tất cả những gì mà máy tính quan tâm. Điều đó cũng giống như câu nói của Carl Sagan, nhà vật lý học nổi tiếng: “Nếu bạn muốn có một chiếc bánh táo từ con số không, thì trước tiên bạn cần phải tạo ra một vũ trụ”.

“Vũ trụ” trong chương trình của bạn được xác định bởi (1) chỉ lệnh và yêu cầu trong cơ sở mã, (2) thư viện mà chương trình của bạn nhập về, và (3) hệ thống mà chương trình đang chạy trên đó. Nếu đoạn mã đó là cần thiết để hoàn thành một hoạt động cụ thể nhưng lại không tồn tại trong hệ thống, thì chương trình của bạn sẽ không chạy hoặc báo lỗi.

Hầu hết các ngôn ngữ lập trình đều chứa những thư viện phổ biến mà hầu hết các chương trình đều cần, nhưng chỉ có một số công cụ chuyên dụng. Đó chính là lúc cần phải có bộ khung. Thay vì mã hóa mọi thứ ngay từ đầu – việc này sẽ tốn rất nhiều thời gian, việc sử dụng bộ khung giúp bạn nhập và sử dụng những thư viện đáng tin, đã qua kiểm tra dành cho những nhiệm vụ cụ thể, từ đó giúp bạn tập trung vào phần cốt lõi của ứng dụng thay vì tái tạo lại cả vũ trụ.

Hiện tại, Ruby tự hào với một vài bộ khung phát triển ứng dụng web chính. Trong số đó có hai bộ khung phổ biến nhất là Ruby on Rails và Sinatra.

Ruby on Rails (thường được viết tắt là Rails) là một trong những bộ khung ứng dụng web quan trọng hàng đầu do Ruby phát triển. Được David Heinemeier Hansson tạo ra năm 2004, Rails dễ dàng trở thành bộ khung Ruby phổ biến nhất, và được sử dụng để phát triển một vài ứng dụng thành công ở công ty 37signals – công ty ứng dụng web thuộc sở hữu tư nhân mà Hansson là đối tác. Tính tới nay, đã có hàng nghìn doanh nghiệp phát triển ứng dụng web quan trọng với công việc kinh doanh bằng cách sử dụng Rails.

Rails phụ thuộc nhiều vào “người tạo”: Các chương trình được cài đặt sẵn tạo ra một lượng lớn mã văn bản thảo sẵn với một yêu cầu đơn nhất. Văn bản thảo sẵn sau đó được điều chỉnh dựa trên những yêu cầu độc đáo của người lập trình. Thay vì dành hàng giờ tạo ra một ứng dụng ngay từ đầu, Rails giúp các nhà phát triển tạo ra một ứng dụng chức năng mà không tốn nhiều công sức, chỉ cần họ biết họ đang làm gì.

Ngược lại, Sinatra là bộ khung rất nhỏ được Blake Mizerany thiết kế và phát triển. Thay vì phụ thuộc vào người tạo ra, Sinatra tập trung vào việc cung cấp cho người phát triển một số chức năng đơn giản phổ biến mà hầu hết các ứng dụng đều cần, sau đó là hết nhiệm vụ.

Ứng dụng Sinatra tạo cảm giác đơn giản hơn so với ứng dụng Rails. Một yêu cầu đơn nhất trong Rails cũng có thể tạo ra ít nhất 10 thư mục và 20 tập tin. Ngược lại, ứng dụng Sinatra thường không được chứa đầy đủ trong một tập tin đơn lẻ. Thay vì tạo ra hàng tấn mã có thể cần phải loại bỏ, Sinatra lại ưu tiên

việc giữ cho dự án đơn giản và chỉ thêm đủ mã để hoàn thành công việc.

Giống như chọn ngôn ngữ, chọn bộ khung chủ yếu cũng là vấn đề tham khảo và chọn công cụ tốt nhất cho công việc. Rails thường được sử dụng cho những dự án lớn với nhiều người lập trình, còn Sinatra phù hợp với những dự án nhỏ hơn. Hai ứng dụng này có nhiều tính năng giống nhau, vì thế, một bài phân tích mới đây trên trang RubySource.com đã kết luận rằng, cuối cùng đó là sự lựa chọn mang tính chủ quan.

GitHub là kho mã nguồn mở mà nhiều người lập trình sử dụng để tung ra và duy trì các dự án của họ. Thật dễ tìm thấy ví dụ về các ứng dụng được viết bằng Rails và Sinatra để có cảm nhận về mỗi loại khung, vì thế tôi đã dành thêm một giờ nữa để duyệt các dự án công khai.

Ở đây có một nguy cơ không hề nhỏ: Để đạt được tiến bộ trong lập trình, bạn cần phải cam kết một việc. Một khi bạn đã chọn được ngôn ngữ và bộ khung, sẽ dễ hơn nhiều để bắt đầu học mọi thứ bạn cần biết để viết được một chương trình. Tuy nhiên, nếu bạn chần chừ trong việc đưa ra lựa chọn, bạn có thể mất hàng năm chỉ để cố gắng tìm kiếm môi trường lập trình “hoàn hảo”.

Sẽ tốt hơn khi chọn một ngôn ngữ và bộ khung cuốn hút bạn, cam kết khám phá nó trong một khoảng thời gian, và chấp nhận những đánh đổi không thể tránh được thay vì mất hàng năm “tìm kiếm” và không tiến bộ được chút nào. Duyệt Stack Overflow và Hacker News hàng ngày *không* phải là lập trình.

Cuối cùng, tôi đã quyết định là sẽ bắt đầu với Sinatra. Vì dù rằng Rails có thể tiết kiệm rất nhiều thời gian nếu bạn biết bạn đang làm gì, nhưng tôi lại không biết tôi đang làm gì.

Trực giác của tôi nói cho tôi biết Sinatra là lựa chọn tốt nhất tại thời điểm đó. Cú pháp rõ ràng, đơn giản và dễ hiểu. Mã duy nhất tồn tại trong ứng dụng Sinatra là mã mà người phát triển thêm vào. Có nhiều tài liệu viết về bộ khung và dễ tìm thấy những ví dụ về các ứng dụng đang hoạt động trên GitHub, cũng như sự giúp đỡ trên Stack Overflow.

Có thể một lúc nào đó trong tương lai, tôi sẽ thử Rails. Nhưng lúc này, tôi sẽ bắt đầu với Sinatra.

Chia nhỏ kết quả cuối cùng

Khoảng 5 giờ đồng hồ tìm kiếm sơ bộ đã đem lại tất cả những gì tôi cần để bắt đầu: chia nhỏ công việc lập trình web, ngôn ngữ, bộ khung và một dự án cụ thể. Đã đến lúc bắt tay vào làm việc.

“Viết mã một ứng dụng Sinatra phục vụ cho trang web Jekyll” là mức độ thành thạo mục tiêu của tôi, nhưng tôi cần phải chia nhỏ những gì ẩn trong mệnh đề này để xác định được điều tiếp theo cần thực hiện. Tôi cần phải làm được việc gì?

Thêm một giờ nghiên cứu, tôi đã đi đến các quyết định sau:

1. Chạy Jekyll tạo ra trang web hoàn thiện từ các tập tin thông kê cục bộ. Tôi sẽ cần tạo ra một mẫu HTML cho trang web đó với các thẻ định dạng đặc biệt, và xuất khẩu kho lưu trữ các bài đăng tải từ trang PersonalMBA.com – việc này được miêu tả trong một bài hướng dẫn của Paul Stamatiou.
2. Ứng dụng Sinatra xử lý các yêu cầu từ khách ghé thăm trang web, chuyển giao các tập tin được yêu cầu. Tôi cần phải viết ứng dụng này ngay từ đầu.
3. Cả trang web Jekyll hoàn thiện và ứng dụng máy chủ Sinatra đều cần phải được đăng tải trên một máy chủ web.
4. Để hoàn thiện tất cả nhiệm vụ này, tôi cần xác định cách cài đặt phiên bản mới nhất của Ruby, cũng như Sinatra và các chương trình khác mà tôi cần, vào máy của tôi.

Yêu cầu cuối cùng này là một ví dụ hay ho về việc tìm kiếm các công cụ cần thiết. Nếu tôi không thể xác định được cách để cài đặt Ruby vào máy của mình thì tôi không thể làm được bất cứ việc gì, vậy nên đó sẽ là điểm khởi đầu tốt nhất.

Quan trọng là phải lưu ý rằng công nghệ web thay đổi hàng ngày. Nhiều khả năng là những yêu cầu cụ thể trong phần này sẽ trở nên lỗi thời khi bạn đọc cuốn sách này. Đừng lo lắng, vì phương pháp mới là điều quan trọng, chứ không phải yêu cầu.

Tương tự như vậy, bạn sẽ bị thôi thúc là phải che đậy phần mã trong chương này. Đó là một thôi thúc tự nhiên: Nó có vẻ phức tạp và bạn sẽ không thể ngay lập tức biết được nó có nghĩa là gì.

Tôi khuyến khích bạn đấu tranh chống lại sự thôi thúc này. Những cái tên, yêu cầu và biểu tượng này cũng không quen thuộc với tôi, giống như với bạn. Chương này nói về quá trình xác định điều này có nghĩa là gì, và làm sao để sử dụng nó. Nếu bạn cố gắng đọc mã, bạn sẽ nhận được nhiều hơn từ chương này.

Tiến lên nào!

## Nâng cấp Ruby

Tôi đã có máy tính, đó là điểm khởi đầu, vì bạn không thể lập trình được nếu không có máy tính.

Hiện tại, tôi đang sử dụng một chiếc MacBook Air của Apple, hệ điều hành Mac OS X 10.6. Chỉ cần thực hiện một thao tác tìm kiếm nhanh là biết hệ điều hành đó là Ruby phiên bản 1.8.7 cài đặt sẵn. Đó là một tin tốt, vì về mặt lý thuyết tôi có thể bắt đầu chạy các chương trình Ruby trên chính máy tính của mình.

Vấn đề là Ruby 1.8.7 không phải là phiên bản mới nhất của Ruby. Khi tôi có cài đặt Jekyll, hệ thống báo cho tôi biết chương trình cần phiên bản mới nhất 1.9.1, điều đó có nghĩa là tôi sẽ phải xác định cách nâng cấp. Tôi đành quay lại với Google thôi.

Một số thao tác tìm kiếm cơ bản đưa ra kết quả là hai chương trình được thiết kế để việc quản lý công đoạn cài đặt Ruby được dễ dàng hơn: *rbenv* và *ruby-build*. Cả hai chương trình đều được duy trì bởi Sam Stephenson, người đã phát triển Ruby ở 37signals. Cùng với đó, những chương trình này có thể giúp bạn cài đặt những phiên bản mới của Ruby và nói cho máy tính của bạn biết cần phải sử dụng phiên bản Ruby nào.

Có một bản hướng dẫn về trang tư liệu *rbenv* cho biết cách cài đặt chương trình vào máy của bạn. Các yêu cầu cài đặt như sau:

```
$ cd ~
```

```
$ git clone git://github.com/sstephenson/ruby-build.git
```

```
$ cd ruby-build
```



```
$ sudo ./install.sh
$ cd ..
$ git clone git://github.com/sstephenson/rbenv.git .rbenv
$ mkdir -p ~/.rbenv/plugins
$ cd ~/.rbenv/plugins
$ git clone git://github.com/sstephenson/ruby-build.git
$ echo 'export PATH="$HOME/.rbenv/bin:$PATH"' >> ~/.bashprofile
$ echo 'eval "$(rbenv init -)"' >> ~/.bashprofile
$ exec $SHELL
$ rbenv install 1.9.3-p125
$ rbenv rehash
$ rbenv global 1.9.3-p125
```

Điều này trông có vẻ đáng sợ, nhưng đó chỉ là một danh sách các yêu cầu mà thôi. Hãy chia nhỏ nó ra.

Những yêu cầu này được đưa vào một chương trình có tên là *Terminal*, được cài đặt sẵn trong các máy tính của Apple. Nếu bạn đã từng nhìn thấy các hacker trên phim gõ điên cuồng vào máy tính đang hiển thị những dòng văn bản dài ngoằng, có nghĩa là chiếc máy tính đó đang chạy chương trình *Terminal*.

Tôi mở *Terminal* và nhập yêu cầu đầu tiên:

```
$ cd ~
```

Yêu cầu này rất dễ hiểu. “\$” là ký hiệu *Terminal* hiển thị khi nó đã sẵn sàng cho một yêu cầu mới. “cd” là viết tắt của “change directory” (chuyển thư mục), một thuật ngữ khác để chỉ thư mục. Tìm kiếm nhanh cho tôi biết “~” là viết tắt của “thư mục gốc của người sử dụng”, thư mục trong máy tính của tôi, nơi lưu giữ hồ sơ.

Tôi gõ yêu cầu và nhấn nút Enter. Lúc này, *Terminal* hiển thị như sau:

joshkaufman \$

Đó là điều đáng mừng, tôi đang ở thư mục gốc của mình. Tôi gõ yêu cầu thứ hai:

```
$ git clone git://github.com/sstephenson/ruby-build.git
```

Máy tính trả lại:

```
git: command not found (không tìm thấy yêu cầu)
```

Có vẻ như chương trình git không được cài đặt trong máy của tôi. Tôi cần phải xác định cách cài đặt nó.

## “Git” là gì?

Sau khi tìm kiếm hướng dẫn về cách cài đặt git vào Mac OS X, tôi thấy Heroku Toolbelt. Heroku là công ty máy chủ ứng dụng web nên họ có hứng thú với việc giúp cho các nhà phát triển có thể dễ dàng tạo ra các ứng dụng web.

Heroku Toolbelt là chương trình cài đặt một vài công cụ phát triển phần mềm phổ biến mà các nhà lập trình cần có để phát triển ứng dụng trên Heroku. git là một trong những chương trình đó.

Tôi tải gói cài đặt về, chạy và được xác nhận là mọi thứ được cài đặt chính xác. Đã đến lúc thử lại yêu cầu trước:

```
$ git clone git://github.com/sstephenson/ruby-build.git
```

Tôi nhận được kết quả là:

```
Cloning into ruby-build...
```

```
remote: Counting objects: 1004, done.
```

```
remote: Compressing objects: 100% (453/453), done.
```

```
remote: Total 1004 (delta 490), reused 937 (delta 431)
```

```
Receiving objects: 100% (1004/1004), 108,77 KiB, done.
```

```
Resolving deltas: 100% (490/490), done.
```

Thành công! “*Done*” (hoàn thành) là một dấu hiệu tốt, và tôi không nhận được thông báo báo lỗi. Tiến lên nào!

Tôi tiếp tục với phần còn lại của các yêu cầu. Dựa trên thông tin trong phần hướng dẫn cài đặt, tất cả những gì tôi phải làm là tải các tập tin cần thiết, sử dụng một yêu cầu được gọi là *echo* để tự động thêm đoạn văn bản vào tập tin xác định cấu hình máy tính của tôi, sau đó khởi động lại chương trình có tên là SHELL trên máy tính để lưu lại các thay đổi. Một khi SHELL khởi động lại, *rbenv* và *ruby-build* đã được cài đặt. Oa!

Giờ đã đến lúc để cài đặt phiên bản mới nhất của Ruby:

```
$ rbenv install 1.9.3-p125
```

Chương trình tự động tải mã nguồn Ruby và xây dựng nó, đẩy ra một lượng ấn tượng các thông tin chạy trong chương trình *Terminal* đang được xử lý. (Lúc này tôi cảm thấy mình giống như một người lập trình đích thực trong phim của Hollywood.)

Yêu cầu này giúp máy tính nhận ra có một phiên bản mới của Ruby đã được cài đặt.

```
$ rbenv global 1.9.3-p125
```

Yêu cầu này đặt phiên bản 1.9.3-p125 là phiên bản mặc định của Ruby trên chiếc máy tính này. Tài liệu hướng dẫn cho tôi biết chạy yêu cầu này có thể đảm bảo chắc chắn máy tính của tôi đang sử dụng phiên bản mới:

```
$ ruby -v
```

Và tôi nhận được:

```
ruby 1.9.3p125 (2012-02-16 revision 34643) [x8664-darwin11.3.0]
```

Thành công rồi! Đó chính là điều phải đạt được, dựa trên bản hướng dẫn.

Theo tài liệu, nếu tôi muốn cài đặt một phiên bản mới của Ruby vào chiếc máy tính này, tất cả những gì tôi phải làm là chạy lại *rbenv install*, *rbenv rehash* và *rbenv global*. Quá dễ.

Dù rằng lúc đầu các yêu cầu nhìn có vẻ đáng sợ, nhưng thực ra chúng lại rất đơn giản. Những thứ trông không khác gì thảm họa ngôn ngữ thực ra lại chỉ là những ký hiệu. Một khi bạn biết ký hiệu đó tượng trưng cho cái gì thì các yêu

câu đó tự nhiên cũng trở nên dễ hiểu.

Hãy nhớ, chẳng ai sinh ra đã biết điều này (lập trình). Hầu hết thời gian, tất cả những gì bạn cần phải làm là dành vài phút để đọc tài liệu, sau đó cố gắng thực hiện theo tài liệu đó.

## Cài đặt thư viện Ruby (Gems)

Giờ thì phiên bản mới nhất của Ruby đã được cài đặt. Đã đến lúc xác định cách cài đặt các thư viện mà tôi cần, bao gồm cả Sinatra.

Các thư viện của Ruby được gọi là *gems*, và hóa ra là chúng rất dễ cài đặt. Dưới đây là các yêu cầu để cài đặt gem Sinatra:

```
$ gem install sinatra
```

Để cập nhật Sinatra, đưa ra yêu cầu như sau:

```
$ gem update sinatra
```

Chẳng có gì dễ dàng hơn thế!

Tuy nhiên, trước khi tôi cài đặt quá nhiều thư viện, tôi muốn chắc chắn là chương trình gem được cập nhật. Vì máy tính của tôi sử dụng phiên bản Ruby cũ, nên có vẻ như các phần mềm liên quan cũng cần phải cập nhật theo.

Sau một thời gian ngắn tìm kiếm, tôi tìm thấy yêu cầu cập nhật chương trình gem của Ruby:

```
$ gem update--system
```

Đơn giản.

Khi tôi chạy yêu cầu *gem install* (cài đặt gem), tôi để ý thấy yêu cầu cũng cài đặt các gem bổ sung, như *rack*, *rack-protection* và *tilt*. Những thư viện phụ này được gọi là (thư viện) phụ thuộc (*dependencies*). Sinatra phụ thuộc vào những thư viện đó để chạy, vì thế yêu cầu *gem install* cài đặt chúng một cách tự động.

Lúc này tôi đã sẵn sàng chạy ứng dụng Ruby, tôi quyết định nhắc hai cuốn sách tham khảo về Ruby được nhiều người trên Stack Overflow khuyên đọc: *Người thành thạo Ruby* (The well- Grounded Rubyist) (2009) của David A. Black và *Ruby hùng biện* (Eloquent Ruby) (2011) của Russ Olsen. Cả hai cuốn đều là sách vỡ lòng được thiết kế để giới thiệu cho độc giả những khái niệm và thủ thuật Ruby phổ biến, cũng như những đoạn văn bản tham khảo cơ bản nhất.

Tôi còn mua cả cuốn *Sinatra: Cài và chạy* (Sinatra: Up and Running) (2011) của Alan Harris và Konstantin Hasse. Cuốn sách này được thiết kế như phần giới thiệu cơ bản về bộ khung Sinatra. Dù có rất nhiều tài liệu về Sinatra trên mạng, nhưng cuốn sách này cũng có rất nhiều ví dụ – điều này giúp bạn dễ dàng xác định được cách sử dụng Sinatra cho những nhiệm vụ phổ biến.

Như vậy là có rất nhiều tài liệu tham khảo để bắt đầu, vì thế tôi đã dành ra 90 phút để xem nhanh tất cả những gì tôi thu lượm được từ trước đến giờ.

Tôi ngồi xuống cùng với đồng sách, và với mỗi cuốn tôi lại đọc lướt bảng đề mục và phụ lục, chú ý tới những thuật ngữ và ý tưởng có vẻ quan trọng. Tôi cũng nắm bắt các khái niệm xuất hiện đi xuất hiện lại, cũng như thứ tự của phần mở đầu. Tôi đọc các đề mục chính và các cột nội dung bên lề. Khi đã xử lý xong đồng sách, tôi cũng làm tương tự với các trang web.

Và đây là điều tôi đã học được. Ngoài biến số, điều kiện, ngoại lệ và những điều cơ bản khác của lập trình, Ruby được xây dựng quanh hai ý tưởng cốt lõi là *đối tượng* và *phương pháp*.

Đối tượng là danh từ của thế giới lập trình. Đó là những thứ chúng ta có thể làm gì đó (về hoặc với chúng). Giả sử tôi muốn tạo ra một biến số mới trong Ruby được gọi là `firstname`, và tôi muốn nó có tên tôi. Trong Ruby, yêu cầu đó như sau:

```
firstname = "Josh"
```

Rất đơn giản. Bằng cách đặt "Josh" trong dấu ngoặc kép, tôi đang nói cho Ruby biết `firstname` là một chuỗi ký tự (*string*): Dãy vừa có chữ vừa có số. Điều đó khiến `firstname` trở thành một đối tượng trong loại "chuỗi ký tự". (Loại là một kiểu đối tượng cụ thể với những đặc điểm nhất định.)

Chuỗi ký tự không nhất thiết chỉ là loại các đối tượng. Đây là đối tượng thuộc loại số nguyên:

million = 1000000

Nếu đối tượng là danh từ trong lập trình thì phương pháp lại là động từ. Chúng là những việc mà chúng ta có thể làm về (hoặc với một đối tượng).

Giả sử tôi có hai chuỗi đối tượng có chứa tên và họ của tôi:

firstname = “Josh”

lastname = “Kaufman”

Tôi có thể sử dụng dấu cộng (+) để *ghép* những chuỗi ký tự này lại với nhau, thuật ngữ để chỉ việc xếp chúng lại gần nhau:

fullname = firstname + lastname

Một câu đố vui, *fullname* (tên đầy đủ) có chứa gì? Nếu bạn đoán là “Josh Kaufman” thì bạn đã nhầm.

Hãy nhớ, máy tính chỉ làm chính xác những gì bạn bảo nó làm. Chúng ta chưa bảo nó thêm khoảng cách giữa “Josh” và “Kaufman”, vì thế nó sẽ không thêm đâu. Tên đầy đủ trong trường hợp này là “JoshKaufman”.

Nếu chúng ta muốn sửa lỗi sai nho nhỏ này, chúng ta cần phải thay đổi đoạn mã để thêm khoảng cách vào:

fullname = firstname + “ ” + lastname

Dấu + là một phương pháp, và cách phương pháp hoạt động phụ thuộc vào các đối tượng mà chúng ta đang xử lý. Nếu chúng ta sử dụng số nguyên thay vì chuỗi ký tự, hệ thống sẽ thực hiện phép cộng thay vì phép ghép:

sum = million + million

Tổng (sum) đó bằng bao nhiêu? “2000000”

Các phương pháp được cài đặt sẵn của Ruby có thể giúp bạn thực hiện ngay được rất nhiều việc thú vị. Giả sử tôi muốn thấy tên đầy đủ của mình được viết ngược. Thay vì phải viết bằng tay hoặc viết một chương trình nho nhỏ để đảo ngược các chữ cái, tôi có thể sử dụng phương pháp *reverse* (đảo lộn) có sẵn dành cho mọi đối tượng chuỗi ký tự:

fullname.reverse

Và đây là kết quả tôi nhận được: `namfuaKhsoJ`

Tôi cũng có thể sử dụng nhiều hơn một phương pháp cùng một lúc. Nếu tôi muốn đảo lộn các chữ cái trong tên của mình và tất cả các chữ cái đều được viết thường, thì tôi có thể chạy đoạn mã này:

```
fullname.reverse.downcase
```

Kết quả nhận được là: `namfuakhsoj`. Chính xác luôn!

Học viết mã Ruby chủ yếu là học cách sử dụng, sáng tạo và điều khiển các đối tượng, các lớp và các phương pháp. Ngôn ngữ này có rất nhiều thứ được cài đặt sẵn, và Ruby cho phép bạn tạo ra, điều chỉnh hoặc loại bỏ các đối tượng, các lớp và các phương pháp tùy theo ý thích của bạn. Điều đó khiến ngôn ngữ này hữu dụng và linh động hơn.

Tài liệu hướng dẫn chính thức của Ruby có chứa danh sách kiểu mẫu của tất cả những đối tượng và phương pháp có thể sử dụng ngay. Xem nhanh một lượt có vẻ là hơi quá, nhưng điều đó lại giúp bạn nhận ra bạn không cần phải sử dụng tất cả những thứ đó. Chúng mang tính lựa chọn, và bạn có thể sử dụng chúng khi cần.

Tài liệu hướng dẫn còn có một mục đích khác, đó là khi bạn cố gắng làm điều gì đó mà Ruby không hiểu được, thông tin báo lỗi có thể nói cho bạn biết chương trình đã bị hỏng ở đâu.

Giả sử chúng ta đang cố gắng chạy một chương trình như sau:

```
animal = "Zebra" (con vật = "ngựa vằn")
```

```
number = 7 (số lượng = 7)
```

```
puts animal + number (in con vật + số lượng)
```

Lệnh `put` là cách nói khác của lệnh `in` (`print`). Chúng ta muốn chương trình hiển thị kiểu chữ `animal + number` có nghĩa là gì.

Và đây là kết quả tôi nhận được khi cố gắng chạy chương trình:

```
TypeError: can't convert Fixnum into String from program.
```

```
rb:3:in '+'
```

(Lỗi đánh máy: Chương trình không thể chuyển số đã nhập thành chuỗi ký tự

```
rb:3:in '+')
```

Nếu không sử dụng ngôn ngữ máy tính thì: bạn không thể sử dụng số học để cộng một số vào một từ mà vẫn có nghĩa được, vì thế máy tính đã hiển thị thông báo báo lỗi. Thế chẳng khác nào cố gắng thực hiện phép chia cho không (0), nên chương trình đã dừng lại.

Để sửa chương trình, chúng ta hoặc là cần phải đổi số thành một chuỗi ký tự để phương pháp + có thể ghép hai biến số lại với nhau, thay vì cố gắng sử dụng số học; hoặc là phải điều chỉnh chương trình để thực hiện một việc gì đó khác.

Và đây là chương trình đã được sửa đổi:

```
animal = "Zebra"
```

```
number = 7.to_s
```

```
puts animal + number
```

(con vật = "ngựa vằn"

số lượng = 7.to\_s

In con vật + số lượng)

Khi chạy chương trình, chúng ta sẽ nhận được kết quả là "Zebra7" ("Ngựa vằn 7"). Phương pháp được cài đặt sẵn .to\_s đã biến số 7 thành chuỗi ký tự, vì thế Ruby có thể thực hiện phép ghép.

Chúng ta cũng có thể làm điều gì đó hoàn toàn khác biệt, kiểu như thế này:

```
animal = "Zebra"
```

```
number = 7
```

```
number.times {puts "#{animal}"}
```

(con vật = "ngựa vằn"

số lượng = 7



số lượng.lần {in “#{con vật}”})

Và đây là kết quả:

Zebra

Zebra

Zebra

Zebra

Zebra

Zebra

Zebra

(Ngựa vằn)

Chúng ta vừa sử dụng vòng lặp điều kiện cơ bản được tích hợp sẵn trong Ruby: *number.times* có nghĩa là “thực hiện việc này X lần, với X là giá trị của biến là số lượng”. Nếu chúng ta thay đổi giá trị của biến động vật hay số lượng, chúng ta sẽ thay đổi kết quả. (Đúng vậy, bạn có thể điều chỉnh chương trình này để in “gấu túi” hàng tỉ lần nếu bạn thực sự muốn.)

## Nhận xét và gỡ lỗi

Khi đọc, tôi còn để ý một đặc điểm cơ bản nữa trong Ruby: *nhận xét*. Bất cứ khi nào bạn bắt đầu một dòng chương trình với # (thường được gọi là “dấu thăng”), Ruby sẽ dịch dòng đó như một lời nhận xét và sẽ bỏ qua nó.

Thêm nhận xét cho một chương trình khiến cho việc theo dõi chương trình đó trở nên dễ dàng hơn, vì bạn có thể giải thích bằng ngôn ngữ đơn giản điều bạn đang cố gắng thực hiện. Và nhận xét trong chương trình “Animal Print” trông sẽ như thế này:

```
# Assign variables
```

```
animal = “Wombat”
```

```
number = 1000000000
```

```
# Print loop
number.times {puts “#{animal}” }

(# Gán biến số
con vật = “gấu túi”
số lượng = 1000000000

# vòng lặp in
số lượng.lần {in “#{con vật}” }
```

Nhận xét cũng là một thủ thuật khắc phục sự cố cơ bản: bạn có thể chú thích một vài dòng mã cùng một lúc để cô lập vấn đề hoặc lỗi sai. Kết hợp với các câu lệnh in (print hoặc put) được đặt đúng chỗ, bạn có thể thực hiện chương trình từng bước một để đảm bảo mọi thứ đều hoạt động đúng như mong đợi.

Sau tổng cộng 8 tiếng nghiên cứu và cài đặt, giờ tôi đang chạy phiên bản mới nhất của Ruby. Tôi đã có thể cài đặt bất cứ thư viện nào mà tôi muốn, và tôi đã có kiến thức cơ bản về cách thức hoạt động của các chương trình Ruby.

Quan trọng là cần phải chú ý rằng, tới tận lúc này tôi vẫn chưa thực sự lập trình được bất cứ thứ gì. Toàn bộ quãng thời gian vừa rồi được dùng vào việc nghiên cứu, cài đặt Ruby, và xem viết một chương trình Ruby sẽ có cảm giác như thế nào.

Giờ hãy cùng khám phá những chương trình phức tạp hơn.

Đã đến lúc phải bắt tay vào làm rồi đây...

## Ứng dụng #1: Trang web tĩnh bằng Sinatra

Tôi đã có ý tưởng cho ứng dụng web đầu tiên của mình: một ứng dụng Sinatra phục vụ cho trang web HTML cơ bản. Mục độ thành thạo mục tiêu của tôi đối với ứng dụng này là:

1. Tạo ra một ứng dụng Sinatra cơ bản có thể đưa một trang web đơn giản đến người sử dụng cuối cùng (người đọc).
2. Kiểm tra ứng dụng trên máy tính của tôi để đảm bảo nó hoạt động.

3. Triển khai ứng dụng đó vào sản xuất trên Heroku, khiến nó “sống” để những người đọc thực sự có thể sử dụng.

Chính xác là vậy. Không có gì đặc biệt, chỉ là một chương trình Sinatra đơn giản chạy trên một máy chủ công cộng.

Vậy thì tôi nên bắt đầu từ đâu? Hãy cùng điểm lại bản liệt kê những mục cần kiểm tra của tôi:

1. Chọn một dự án đáng yêu.
2. Tập trung năng lượng của bạn vào chỉ một kỹ năng.
3. Xác định mức độ thành thạo mà bạn mong muốn.
4. Phân chia kỹ năng thành những kỹ năng nhỏ.
5. Tìm được những công cụ then chốt.
6. Loại bỏ những rào cản để luyện tập.
7. Dành thời gian chuyên tâm luyện tập.
8. Tạo vòng phản hồi nhanh.
9. Luyện tập tính giờ.
0. Nhân mạnh số lượng và tốc độ.

Tôi có một dự án được xác định rõ ràng. Tôi đã chia nhỏ kỹ năng và tôi biết chương trình này trông như thế nào khi được hoàn thành. Điều đó đưa tôi đến với những công cụ cần thiết: Có thứ gì cần để hoàn thành dự án này mà tôi chưa có không?

Hóa ra là còn: Tôi chưa có tài khoản Heroku. Điều đó dễ giải quyết: Tôi truy cập vào trang Heroku.com, click vào ô Đăng ký, xác nhận địa chỉ e-mail và tạo mật khẩu.

Vì tôi đã tải Heroku Toolbelt (chương trình tôi đã sử dụng để cài đặt git), thư viện (gem) Heroku cũng đã có trên máy của tôi, nên tôi đã sẵn sàng.

Dựa trên tài liệu hướng dẫn, còn một điều cuối cùng tôi cần phải làm để máy tính của tôi có thể “nói chuyện” được với Heroku, đó là tạo ra “chìa khóa SSH”

– một tập tin đặc biệt đóng vai trò như mật khẩu. Một khi đã có chìa khóa, tôi sẽ phải đăng tải lên Heroku để hệ thống có thể nhận biết được máy tính của tôi và cho phép nó truy cập.

May mắn là Heroku có tài liệu hướng dẫn về cách làm việc này. Tôi đã chạy yêu cầu này để tạo chìa khóa:

```
$ ssh-keygen -t rsa
```

... và yêu cầu này để đăng nhập vào Heroku:

```
$ heroku login
```

... và yêu cầu này để thêm chìa khóa vào tài khoản Heroku của tôi:

```
$ heroku keys:add
```

Tuyệt vời. Tôi đã vào được. Giờ tôi bắt đầu viết ứng dụng như thế nào?

## Tạo ra ứng dụng cơ bản

Đã đến lúc duyệt qua tài liệu của Heroku. Điều tuyệt vời là có tới hai tài liệu hướng dẫn có vẻ hữu dụng:

1 “Bắt đầu với Ruby trên Heroku”

2 “Triển khai các ứng dụng dựa trên Rack (Rack-based Apps)”

Dựa trên những hướng dẫn này, có vẻ như tôi cần phải:

1. Tạo ra các tập tin chương trình trên máy tính của tôi.
2. Thêm chúng vào “kho git” (bất kể đó là gì...)
3. Sử dụng yêu cầu git push heroku master để gửi ứng dụng hoàn thiện lên Heroku.

May mắn là tài liệu hướng dẫn có một ví dụ, và đó là một ứng dụng Sinatra! Việc này sẽ dễ dàng hơn tôi tưởng.

Tôi tạo một thư mục trong máy tính của mình. Thư mục này được gọi là thư

mục “gốc”, và tất cả các tập tin trong chương trình sẽ được lưu giữ ở đây.

Tiếp theo, tôi mở phần mềm biên tập văn bản (tôi đang sử dụng TextMate) và tạo ba tập tin theo đúng hướng dẫn:

```
application.rb
```

```
config.ru
```

```
Gemfile
```

Phần cốt lõi của chương trình sẽ nằm ở application.rb. Các ứng dụng của Ruby luôn có phần đuôi là .rb.

config.ru là nơi cài đặt cấu hình Rack. Hãy nhớ là Sinatra được xây dựng trên nền Rack, nên hoàn toàn dễ hiểu khi nó có một tập tin xác định cấu hình riêng. Các tập tin “Rackup” có phần đuôi là .ru.

Gemfile là nơi xác định gem nào sẽ được chương trình sử dụng. Chương trình của bạn sẽ chỉ có một Gemfile, nên nó vẫn luôn được gọi là “Gemfile”. Có vẻ cũng đơn giản.

Sau khi tạo các tập tin, tài liệu hướng dẫn Heroku gợi ý nên viết một chương trình đơn giản “Xin chào, thế giới!” để kiểm tra việc thiết lập. Và đây là điều trong application.rb:

```
require 'sinatra'
```

```
get '/' do
```

```
  “Hello World!”
```

```
end
```

```
(yêu cầu “sinatra”
```

```
nhận '/' làm
```

```
“Xin chào Thế giới!”
```

```
kết thúc)
```

Còn đây là những gì diễn ra trong config.ru:

```
require './application.rb'
```

```
run Sinatra::Application
```

```
(yêu cầu './ứng dụng.rb'
```

```
chạy Sinatra::Ứng dụng)
```

Và đây là những gì diễn ra trong Gemfile:

```
source 'http://rubygems.org'
```

```
gem 'sinatra'
```

```
(nguồn 'http://rubygems.org'
```

```
gem 'sinatra')
```

Như vậy là cũng không nhiều mã lắm và cũng tương đối dễ hiểu, đúng không?

1. Gemfile nói cho máy chủ biết phải thêm gem Sinatra vào, có như vậy mới chạy được ứng dụng. Gem sẽ được tải về từ RubyGems.org.

2. Tập tin config.ru thiết lập ứng dụng chính, sau đó thực hiện chương trình.

3. Khi chương trình được chạy, nó sẽ hiển thị dòng chữ “Xin chào Thế giới!” bất cứ khi nào có ai đó truy cập “/” (ký hiệu để chỉ trang chủ của một trang web).

Có thực sự dễ dàng thế không nhỉ?

Tài liệu hướng dẫn cho tôi biết còn một việc nữa cần phải làm, đó là lưu giữ các tập tin trong kho git (đôi khi được viết tắt thành “kho”). Thực ra tôi không chắc điều đó có nghĩa là gì, nhưng tôi biết là git đã được cài đặt, và họ cung cấp các yêu cầu:

```
$ git init
```

```
$ git add -A
```

```
$ git commit -m “Initial Commit”
```

Yêu cầu đầu tiên tạo ra một kho git mới trong thư mục gốc hiện có. Yêu cầu add -A thêm tất cả các tập tin trong thư mục vào kho. Yêu cầu commit -m” chuyển các tập tin tới kho, cùng với tin nhắn từ người lập trình cho biết điều gì đang được chuyển. (Tôi vẫn không hiểu sự khác nhau giữa “thêm” và “chuyển”, nhưng tôi đã ghi chú để tìm hiểu thêm sau.)

Sau khi nhập những yêu cầu này, máy tính thông báo với tôi:

```
[master (root-commit) 8ed1099] Initial commit
```

```
3 files changed, 9 insertions(+), 0 deletions(-)
```

```
create mode 100644 Gemfile
```

```
create mode 100644 application.rb
```

```
create mode 100644 config.ru
```

```
([thực hiện (chuyên-gốc) 8ed1099] chuyên chữ đầu
```

```
3 tập tin đã đổi, 9 lồng ghép(+), 0 xóa(-)
```

```
tạo trạng thái 100644 Gemfile
```

```
tạo trạng thái 100644 ứng dụng.rb
```

```
tạo trạng thái 100644 config.ru)
```

Có vẻ như đã được! Chỉ còn một việc cần làm nữa thôi, đó là tạo một máy chủ mới còn trống trên Heroku, sau đó “đẩy” chương trình của tôi lên máy chủ đó.

Tôi chạy đoạn yêu cầu này để tạo máy chủ:

```
$ heroku create
```

```
($ tạo heroku)
```

Và tôi nhận được:

```
Creating shielded-springs-2049 ... done, stack is stack is bamboo-ree-1.8.7
```

```
http://shielded-springs-2049.herokuapp.com/|git@heroku.com:shielded-springs-2049.git
```

Git remote heroku added

Đã thành công! “Ngăn xếp” (stack) cho biết phần mềm mà máy chủ đang chạy, và thông điệp cho tôi biết URL của máy chủ.

Còn đây là yêu cầu cuối cùng:

```
$ git push heroku master
```

Nếu tất cả đều ổn, tôi sẽ chính thức đẩy ứng dụng đầu tiên của tôi lên Heroku.

Và đây là kết quả tôi nhận được:

Heroku receiving push

Ruby/Sinatra app detected

Gemfile detected, running Bundler version 1.0.7

Unresolved dependencies detected; Installing...

Using-without development:test

! Gemfile.lock will soon be required

!Check Gemfile.lock into git with ‘git add Gemfile.lock’

! See <http://devcenter.heroku.com/articles/bundler>

Fetching source index for <http://rubygems.org/>

Installing rack (1.4.1)

Installing rack-protection (1.2.0)

Installing tilt (1.3.3)

Installing sinatra (1.3.3)

Using bundler (1.0.7)

Your bundler is complete! It was installed into `./.bundle/gems/`

Compiled slug size: 500K



Launching ... done, v4

<http://shielded-springs-2049.herokuapp.com> deployed to Heroku

Giờ là lúc kiểm tra thành quả đây... Tôi mở trình duyệt web, điều hướng tới <http://shielded-springs-2049.herokuapp.com> và đây là điều tôi nhìn thấy:

“Xin chào, Thế giới!”

ĐÃ THÀNH CÔNG!

## Đến thời kỳ của Sinatra

Giờ thì ứng dụng đơn giản của tôi đã được hình thành và đang chạy, cuối cùng tôi cũng có thể bắt đầu học cách vận hành của Sinatra. Tài liệu hướng dẫn về Sinatra rất dễ hiểu, và có rất nhiều ví dụ, chính vì thế mà tôi đã quyết định là sẽ bắt đầu từ đó.

Cốt lõi của các ứng dụng Sinatra là tuyến/hướng. Cách tốt nhất để hiểu được khái niệm này là xem ví dụ.

Các ứng dụng Sinatra cơ bản của chúng ta đều có một tuyến đơn, tuyến này chứa “gốc” trang web nho nhỏ của chúng ta. Những người sử dụng internet thường coi gốc trang web là trang chủ của trang web.

Nếu bạn truy cập [google.com](http://google.com) hay [yahoo.com](http://yahoo.com), trình duyệt web của bạn sẽ gửi một yêu cầu đến máy chủ của Google hoặc Yahoo. Yêu cầu này được gọi là yêu cầu NHẬN (GET), và nó buộc máy chủ phải cho bạn biết bất cứ thứ gì trong thư mục gốc của trang web. Giao thức (*protocol*), hay dạng thức máy tính sử dụng để gửi yêu cầu, được gọi là HTTP (chữ viết tắt của “hypertext transfer protocol” có nghĩa là giao thức truyền tải siêu văn bản). “<http://>” mà bạn vẫn nhìn thấy ở đầu mỗi trang web có nghĩa như thế.

GET là dạng yêu cầu HTTP phổ biến nhất, nhưng đó không phải là dạng duy nhất. Còn có thêm ba “động từ” HTTP nữa, đó là:

, POST – gửi một nguồn lên máy chủ

, PUT – cập nhật nguồn trên máy chủ

DELETE – xóa nguồn trên máy chủ

Nếu bạn đã từng đăng một nhận xét công khai trên một trang web, nhận xét của bạn được gửi tới máy chủ bằng cách sử dụng một lệnh POST. Nếu bạn viết sai và chỉnh sửa lại lời nhận xét đó thì lời nhận xét đã được chỉnh sửa của bạn được gửi tới máy chủ thông qua một lệnh PUT. Nếu bạn nghĩ nhận xét của mình thật ngớ ngẩn và chọn xóa nó đi, trình duyệt sẽ gửi một lệnh DELETE.

Các tuyến bao hàm các lệnh GET, POST, PUT và DELETE chính là cốt lõi cho thấy cách vận hành của các ứng dụng Sinatra. Mỗi một tuyến bạn tạo ra là một điều kiện: “Thực hiện X nếu tuyến Y nhận được một yêu cầu GET/POST/PUT/DELETE”.

Các tuyến của Sinatra còn có thể chứa các biến được gọi là *tham số* (parameter). Các ứng dụng Sinatra thường sử dụng tham số như dữ liệu đầu vào cho quá trình có trong mỗi tuyến.

Hãy cùng thay đổi ứng dụng Sinatra đơn giản “Xin chào, Thế giới!” lúc trước của chúng ta để chào người sử dụng bằng tên của họ. Và đây là tuyến sẽ thực hiện điều đó:

```
get '/hello/:name' do
```

```
  “Hello, #{params[:name]}!”
```

```
end
```

Ứng dụng này hoạt động bằng cách cho phép máy chủ nhìn vào bất cứ điều gì trong phần “tên” của tuyến, sau đó sử dụng nó vào ứng dụng này. Yêu cầu trong tuyến là một chỉ lệnh đơn giản để hiển thị tham số “tên” tới người sử dụng.

Sinatra cho phép bạn đặt tên cho các tham số (ví dụ như : name), nhưng nó cũng có tham số đại diện (hay còn gọi là “splat”) có thể chứa bất cứ thứ gì.

Và đây là cách chúng ta sử dụng nó trong ứng dụng “Hello” đã được điều chỉnh của chúng ta:

```
get '/hello/*' do
```

```
  “Hello, #{params[:splat]}!”
```

end

Có vẻ khá thú vị. Giữa các tham số được đặt tên và các tham số đại diện, bạn có thể tạo ra một số tuyến rất thông minh. Các tuyến bạn tạo ra sẽ quyết định cách thức vận hành ứng dụng Sinatra của bạn.

Như vậy là đã đủ chi tiết cho việc xác định cách viết một chương trình đáp ứng được mức độ thành thạo mục tiêu của tôi. Vì Jekyll đang hình thành những tập tin mà ứng dụng sẽ truyền tải để đáp lại yêu cầu GET của người sử dụng, nên tất cả những gì tôi cần phải làm là viết một vài tuyến chấp nhận những yêu cầu đó, tìm kiếm tập tin chính xác trong hệ thống và chuyển nó đến người đọc.

Dựa trên tài liệu hướng dẫn Jekyll, chương trình đặt các trang web hoàn thiện vào một thư mục được gọi là “trang” trong kho gốc. Tuyến tới trang đó được Jekyll tự động tạo ra. Nếu chúng ta muốn chuyên trang About (về trang web) có trên <http://example.com/about>, chúng ta cần phải đặt một tuyến về About trong các tập tin Jekyll của chúng ta, và chương trình sẽ tạo tập tin `in_site/about/index.html` trong thư mục gốc của trang web.

Điều đó có nghĩa là tôi phải tạo ra một tuyến mới trong Sinatra để đọc một tập tin nào đó theo yêu cầu GET của người sử dụng. Và việc tạo ra đó như sau:

```
# Index handler
```

```
get '/' do
```

```
## File.read("_site/index.html")
```

```
end
```

```
# Post handler
```

```
get '/*/' do
```

```
## File.read("_site/#{params[:splat]}/index.html")
```

```
end
```

`File.read("")` là yêu cầu được cài vào Ruby. `File` là một đối tượng, và `.read` là một phương pháp. Cách sử dụng rất đơn giản: Điều gì xuất hiện trong phần (`""`) chính là vị trí của tập tin mà bạn muốn chương trình đọc, có liên quan tới thư mục gốc của ứng dụng. Đơn giản thôi.

Chuyện gì sẽ xảy ra nếu tập tin đó không tồn tại? Nếu vậy thì cần phải có ngoại lệ, và Sinatra có hai tuyến lỗi cơ bản được cài đặt sẵn: `not_found` (không tìm thấy) và `error` (lỗi). Hãy làm cho cả hai tuyến đều trở lại cùng một trang lỗi:

```
not_found do
  ## File.read("_site/error/index.html")
end

error do
  File.read("_site/error/index.html")
end
```

Tất cả những thứ khác đều giữ nguyên. Tôi sẽ không thay đổi bất cứ thứ gì ở `config.ru` hoặc `Gemfile`. Tôi sẽ chỉ thêm các tuyến mới cho kho git của chương trình, xác nhận thay đổi, sau đó đẩy chương trình đã được cập nhật lên Heroku. Xong!

Ứng dụng web hoạt động đầu tiên của tôi đã hoàn thành. Phải mất cả tiếng để tôi xác định cần phải thực hiện những bước đó như thế nào, và thêm một tiếng nữa để xác định làm cách nào để chuyển thông tin và thiết kế trang web của tôi từ WordPress sang Jekyll.

Tổng thời gian để hoàn thiện là khoảng 10 tiếng, bao gồm cả thời gian nghiên cứu và xem lại khái niệm lập trình. Cũng không tồi!

## Ứng dụng #2: Codex, cơ sở dữ liệu ghi chú cá nhân

Ứng dụng đầu tiên của tôi đã hoạt động, và nó hoạt động rất tốt. Nó đơn giản, nhưng trong trường hợp này, đó lại là một lợi thế lớn. Càng ít phần động có nghĩa là chương trình càng ít bị hỏng.

Giờ hãy cùng tìm hiểu một thứ phức tạp hơn một chút.

Bạn có nhớ cuộc thảo luận về cơ sở dữ liệu của chúng ta lúc trước không? Các trang web cơ bản không thể tự cập nhật, vì thế chúng không thể lưu giữ thông tin được. Ứng dụng đầu tiên chỉ hoạt động vì các tập tin tĩnh: chúng không

thay đổi. Bất cứ thay đổi nào đối với các tập tin đều được thực hiện thông qua Jekyll, một chương trình tách biệt. Ứng dụng này nhanh và ổn định vì nó không phụ thuộc vào cơ sở dữ liệu.

Vậy còn những ứng dụng sử dụng cơ sở dữ liệu thì sao? Cơ sở dữ liệu là một phần lớn của các ứng dụng web nói chung, vì thế tôi cần phải hiểu cách thức hoạt động của chúng. Để học về cách chúng hoạt động, tôi cần phải bắt đầu làm việc với một dự án phụ thuộc vào cơ sở dữ liệu.

Một trong những ứng dụng tôi sử dụng hàng ngày là Backpack, một ứng dụng được 37signals phát triển. Lợi ích lớn nhất của Backpack là tạo ra các “trang” chứa được bất cứ thứ gì (văn bản, danh sách, hình ảnh, tập tin...) Khi bạn lưu thông tin trong một trang vào Backpack, sau này bạn có thể truy cập vào thông tin đó từ bất cứ máy tính nào, vì tất cả các thông tin đều được lưu giữ trong cơ sở dữ liệu của ứng dụng.

Tôi tự hỏi là: Liệu tôi có thể tự mình tạo ra thứ gì đó tương tự như vậy không? Cũng đáng để thử đây... nhưng tôi phải bắt đầu như thế nào?

Trong khi nghiên cứu Jekyll, tôi đã đọc được một bài viết của Tom Preston-Werner, nhà lập trình đã tạo ra Jekyll. Bài viết đó có tên là “*Phát triển phần mềm dựa trên README*” (README <sup>(3)</sup> Driven Development).

README là tập tin được người lập trình đưa vào thư mục gốc của ứng dụng trong quá trình viết mã. Tập tin này chứa các thông tin về cách thiết lập, xác định cấu hình và sử dụng chương trình.

Tom lập luận rằng, tốt nhất là nên viết tập tin README cho chương trình đầu tiên, trước khi bạn bắt đầu viết mã. Hầu hết những người lập trình đều viết mã trước, sau đó mới (có thể) viết README. Như thế là đã bỏ lỡ một cơ hội: viết README trước giúp bạn xác định chính xác chương trình sẽ làm việc như thế nào. README có thể là một công cụ thiết kế mà cũng có thể là một tài liệu hướng dẫn.

Tôi lấy một cuốn sổ ghi chép và ghi lại những việc tôi muốn ứng dụng của mình làm được, cũng như những đặc điểm mà tôi muốn ứng dụng của mình có:

, Chương trình là một ứng dụng ghi nhớ và tham khảo đơn giản.

, Ứng dụng được thiết kế cho một người sử dụng.

- Ứng dụng sử dụng Sinatra và cơ sở dữ liệu để tạo ra, lưu giữ, cập nhật và xóa các bản ghi dữ liệu.
- Ứng dụng cho phép người sử dụng tạo ra các trang có các định dạng thú vị như bôi đen, nghiêng, có đề mục...
- Ứng dụng yêu cầu phải nhập mật khẩu để truy cập, và giữ thông tin trong cơ sở dữ liệu an toàn nhất có thể.
- Ứng dụng có giao diện đẹp.
- Ứng dụng có thể dễ dàng được triển khai trên Heroku hoặc một trang chủ tương tự khác.

Tôi sẽ gọi ứng dụng này là Codex, một từ cổ dùng để chỉ “sách”, vì ứng dụng này sẽ chủ yếu được sử dụng như thông tin hay danh sách tham khảo, và những thứ tương tự như vậy.

Thuật ngữ lập trình web dùng để chỉ loại ứng dụng này là “CRUD”, là từ viết tắt của Create (tạo ra), Read (đọc), Update (cập nhật) và Delete (xóa). Cần phải đề cập rằng những chức năng này về cơ bản đều giống như GET, POST, PUT và DELETE, vì vậy hoàn toàn có thể sử dụng các tuyến của Sinatra để xây dựng ứng dụng này. Khác biệt lớn nhất là việc đưa cơ sở dữ liệu vào.

Có những kiểu lựa chọn cơ sở dữ liệu nào trên Heroku? Tôi không biết, phải xem lại tài liệu hướng dẫn thôi.

Theo mặc định, Heroku sử dụng cơ sở dữ liệu được gọi là Postgres. Mỗi ứng dụng mới được phân bổ một cơ sở dữ liệu phát triển mặc định. Điều đó có tác dụng với tôi, nhưng làm thế nào tôi có thể sử dụng được, và tôi cần phải sử dụng cái gì để thử nghiệm chương trình trên máy tính của mình?

## Nhập DataMapper

Để trả lời những câu hỏi này, tôi quyết định tìm kiếm trên Stack Overflow. Tất cả các ý kiến trên trang web này đều đồng nhất ở một điểm, đó là sử dụng thư viện có tên là DataMapper giúp cho việc phát triển này trở nên đơn giản hơn.

DataMapper là kiểu chương trình được gọi là “công cụ vẽ bản đồ mối quan hệ đối tượng” (object relational mapper), thường được viết tắt là ORM. ORM giải quyết được một vấn đề cấp thiết đối với những người lập trình: Cơ sở dữ liệu thường sử dụng ngôn ngữ riêng, loại ngôn ngữ khác với ngôn ngữ mà các nhà lập trình sử dụng để tạo ra ứng dụng. Ngôn ngữ cơ sở dữ liệu phổ biến nhất là SQL, ngoài ra còn có hàng trăm thứ ngôn ngữ khác.

Giả sử chúng ta là người lập trình cho trang web Amazon.com, và chúng ta muốn hiển thị một danh sách các cuốn sách của nhà văn J. K. Rowling, tác giả của loạt truyện Harry Potter nổi tiếng. Khi đó, yêu cầu SQL sẽ như sau:

```
SELECT * FROM Book WHERE author = "J.K. Rowling" ORDER BY title;
```

Lệnh này sẽ lấy ra tất cả các bản ghi có trường “author” chứa tên “J.K. Rowling” từ cơ sở dữ liệu tên Book và sắp xếp chúng theo thứ tự bảng chữ cái của tên sách.

Đáng tiếc là việc làm cho SQL hay bất cứ ngôn ngữ truy vấn cơ sở dữ liệu nào khác chạy mượt mà với những ngôn ngữ lập trình như Ruby là cả một vấn đề. Lập trình bằng một loại ngôn ngữ đã đủ khó rồi, huống hồ là bằng vài loại ngôn ngữ cùng một lúc.

Nhưng đó lại chính là nơi ORM phát huy tác dụng: Chúng cho phép người lập trình viết mã bằng một ngôn ngữ, sau đó ORM sẽ dịch ngôn ngữ đó sang ngôn ngữ của cơ sở dữ liệu. Dễ dàng hơn nhiều.

Như vậy, DataMapper là một thư viện, giúp cho việc liên lạc với cơ sở dữ liệu bằng cách sử dụng Ruby trở nên dễ dàng hơn. Theo mặc định, DataMapper cung cấp một loạt những đặc tính hữu ích cho việc tạo ra, đọc, cập nhật và xóa bản ghi truy cập cơ sở dữ liệu. Vì DataMapper được hình thành một thời gian, và đã được thử nghiệm cẩn thận, nên trong hầu hết các trường hợp, sử dụng nó sẽ đáng tin hơn nhiều so với việc tự tạo ra một loại mã cơ sở dữ liệu riêng.

DataMapper có thể sử dụng như một loại gem, và được cài đặt như sau:

```
$ gem install data_mapper
```

DataMapper là một thư viện lớn, vì thế nên cài đặt từng bước một. Đây là khái niệm được gọi là “hệ mô-đun”, và nó là dấu hiệu xác nhận việc lập trình tốt. Dưới đây là yêu cầu cài đặt toàn bộ các gem riêng lẻ:

```
$ gem install dm-core dm-aggregates dm-constraints dm-migrations dm-
```

transactions dm-serializer dm-timestamps dm-validations dm-types

Thay vì cài đặt toàn bộ thư viện, bạn có thể chỉ cài đặt những phần mà chương trình của bạn sẽ sử dụng, như vậy sẽ hiệu quả hơn nhiều.

## Sử dụng DataMapper

Giờ thì DataMapper đã được cài đặt, tôi cần phải xác định cách sử dụng nó để (1) “nói chuyện” với cơ sở dữ liệu và (2) thiết lập cơ sở dữ liệu để lưu trữ và lấy lại các thông tin mà tôi muốn.

Dựa trên tài liệu hướng dẫn về Postgres của Heroku, yêu cầu dưới đây sẽ cho phép ứng dụng Sinatra của tôi “nói chuyện” được với cơ sở dữ liệu:

```
DataMapper.setup(:default, ENV['DATABASE_URL'] ||  
“sqlite3://#{Dir.pwd}/database.db”)
```

Trong trường hợp này, ký hiệu `||` được sử dụng như một cách khác để nói “hoặc”. `ENV['DATABASEURL']` là một biến mà Heroku sử dụng để tượng trưng cho cơ sở dữ liệu trong ứng dụng của bạn. Nếu cơ sở dữ liệu này không có thì nó sẽ sử dụng lựa chọn thứ hai – cơ sở dữ liệu được gọi là SQLite.

SQLite được mặc định cài đặt trên các dòng máy tính của Mac, nên nó đã sẵn sàng để sử dụng. DataMapper có thể “nói chuyện” được với cả Postgres và SQLite nếu tôi cài đặt hai gem này:

```
$ gem install dm-sqlite-adapter dm-postgres-adapter
```

Điều này có nghĩa là ứng dụng của tôi sẽ sử dụng Postgres khi nó được chạy trên Heroku, và sử dụng SQLite khi nó chạy trên máy tính của tôi. Trong trường hợp nào thì đoạn mã của tôi cũng vẫn như vậy, ngay cả khi cơ sở dữ liệu sử dụng loại ngôn ngữ khác. Điều đó thực sự rất tuyệt.

Nhắc tới việc chạy ứng dụng này trên máy tính của tôi... Làm sao để làm được điều đó?

POW!

Tôi tìm kiếm thông tin về cách chạy kiểu chương trình này trên máy của tôi từ



hai trang Stack Overflow và Hacker News. May mắn là có một số lựa chọn. Có vẻ như tôi có thể cài đặt các loại thư viện (như Foreman hoặc Shotgun) để chạy ứng dụng này khi tôi nhập một yêu cầu vào Terminal, hoặc tôi có thể cài đặt một chương trình để giữ cho chương trình luôn chạy.

Chương trình trong cách thứ hai nêu trên là Pow, một “máy chủ Rack cấu hình số 0 cho Mac OS X”. Trang web cam kết giúp cho việc thiết lập duy trì phát triển nội bộ trên máy tính của tôi trở nên đơn giản và chỉ mất không quá một phút. Nghe có vẻ tuyệt vời với tôi!

Việc cài đặt Pow chỉ mất khoảng 10 giây: Nó đòi hỏi phải có một yêu cầu Terminal để tải về và cài đặt ứng dụng. Một khi đã được cài đặt, bạn phải chạy một yêu cầu để nối chương trình của bạn với chương trình Pow, và Pow sẽ cho phép bạn được chạy nó trên máy tính của bạn.

Có một gem của Ruby được gọi là Powder có tác dụng giúp cho quá trình này thậm chí còn dễ dàng hơn:

```
$ gem install powder
```

Khi gem đó đã được cài đặt, bạn chạy đoạn mã này để cài đặt Pow:

```
$ powder install
```

Sau đó, bạn vào kho dữ liệu gốc ứng dụng của bạn và nhập yêu cầu này vào:

```
$ powder link
```

Vậy thôi. Kho gốc của tôi được gọi là “codex”, vì thế ứng dụng của tôi lúc này đã có thể chạy trên máy tính cá nhân của tôi tại <http://codex.dev>, và tôi có thể thử nghiệm “tác phẩm” của mình.

Nếu tôi thay đổi yêu cầu này sẽ khởi động lại chương trình:

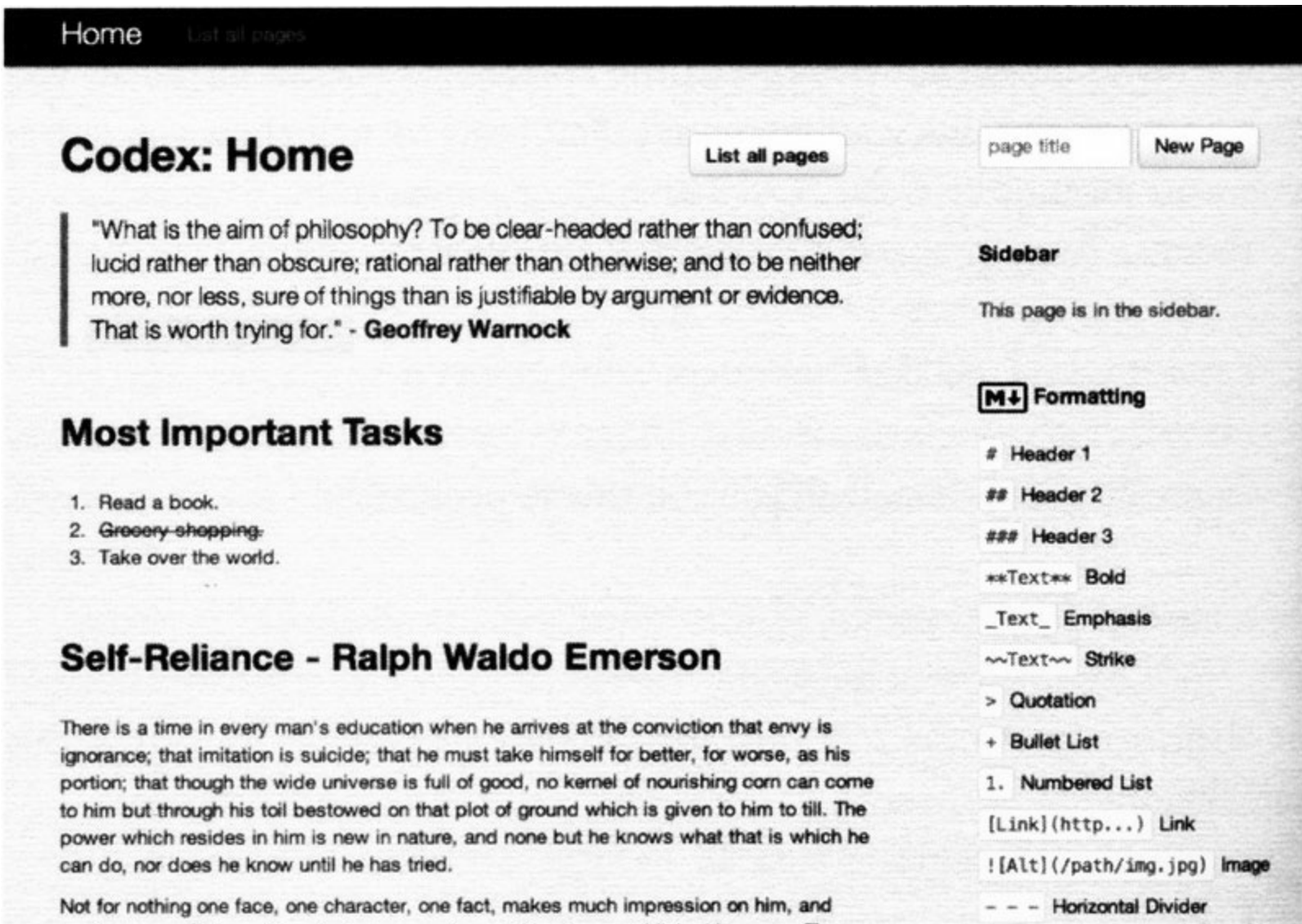
```
$ powder restart
```

Thật đơn giản. Giờ tôi đã sẵn sàng bắt tay vào xây dựng. Tôi dành ra một tiếng rưỡi mỗi tối để viết mã, và tôi sẽ tiếp tục cho tới khi hoàn thành.

Viết mã, kiểm tra, sửa lại

Lúc này, tôi sẽ miêu tả điều tôi sẽ thực hiện thay vì cách tôi thực hiện.

Đây là hình ảnh ứng dụng mà tôi muốn có khi hoàn thành:



Bạn sẽ để ý thấy là thiết kế này có ba phần: thanh điều hướng trên cùng, khu vực nội dung chính và thanh bên. Tôi đặt chúng lại với nhau bằng cách sử dụng Bootstrap do Mark Otto và Jacob Thornton tạo ra (cả hai cùng là lập trình viên ở Twitter).

Thay vì tạo ra một thiết kế trang web ngay từ ban đầu, Bootstrap là một thư viện HTML và CSS viết sẵn, cho sử dụng hoàn toàn miễn phí. Sử dụng Bootstrap sẽ tiết kiệm rất nhiều thời gian: bạn có thể đặt một mẫu cơ bản ban đầu cho ứng dụng của bạn chỉ trong vài phút thay vì vài ngày.

Đơn vị cơ bản của ứng dụng này là “trang” (page), nơi hiển thị bản ghi được lưu giữ trong cơ sở dữ liệu. Thông tin trang thuộc khu vực nội dung chính. Có một nút dẫn tới màn hình, nơi có danh sách tất cả các trang trong cơ sở dữ liệu, và hai nút ở cuối. Nút đầu tiên giúp bạn biên tập, chỉnh sửa trang hiện tại, và nút thứ hai cho phép bạn thực hiện thao tác xóa.

Thanh bên có ba tính năng chính. Ở trên cùng, có một khung mẫu cho phép bạn tạo một trang mới, việc này đòi hỏi phải nhập tiêu đề. Thứ hai, có một danh sách các trang mà người sử dụng đã thêm vào thanh bên, được coi như khu vực tham khảo nhanh. Thứ ba, có một phần tham khảo liên quan tới việc định hình, giúp người sử dụng nhớ được cách sử dụng những tính năng định hình phổ biến.

Thanh điều hướng ở trên cùng có vẻ khá đơn giản. Nó có chứa một đường dẫn (link) tới trang chủ, và một đường dẫn thứ hai tới màn hình “danh sách tất cả các trang”. Tôi có thể thêm các mục khác cho thanh điều hướng sau đó nếu tôi thích, nhưng đó là tất cả những gì tôi thực sự cần lúc này.

Mỗi ứng dụng web đều có một trang chủ, vì thế tôi cần phải quyết định tôi muốn có những gì trên trang chủ đó. Trong trường hợp này, tôi chỉ muốn hiển thị bản ghi trang chủ trong cơ sở dữ liệu.

Vậy thì có gì trong một trang? Vì mỗi trang là một bản ghi cơ sở dữ liệu, và các bản ghi có các trường chứa thông tin thực, nên tôi cần phải nói với DataMapper là cần phải thiết lập các trường nào. Và đoạn mã để làm được việc đó là như thế này:

```
class Page
  include DataMapper::Resource
  property :id, Serial
  property :title, String
  property :content, Text
  property :lastupdated, DateTime
end

DataMapper.finalize
```

Đoạn mã này sử dụng DataMapper để tạo ra một loại đối tượng mới được gọi là trang (page). Lúc này Ruby có thể sử dụng trang như bất cứ một đối tượng nào khác, và tôi có thể tạo ra và sử dụng các phương pháp xây dựng, điều chỉnh và xóa các trang. Khi tôi tạo ra một sự thay đổi trên một trang, sự thay đổi đó sẽ được lưu giữ trong cơ sở dữ liệu thông qua DataMapper.

Lệnh `DataMapper.finalize` yêu cầu ứng dụng thiết lập những trường này trong cơ sở dữ liệu thực nếu chúng chưa tồn tại.

Như vậy là cơ sở dữ liệu của tôi đã được thiết lập, giờ là lúc xác định tuyến nào Sinatra có thể kỳ vọng phục vụ. Dưới đây là danh sách của tôi, dựa trên những gì tôi biết từ trước đến giờ:

```
# Show home page
```

```
get '/'
```

```
# Creates new note from "new page" form
```

```
post '/'
```

```
# Displays requested note
```

```
get '/:url/'
```

```
# Edits requested note
```

```
get '/:url/edit'
```

```
# Saves user edits to note
```

```
post '/:url/edit'
```

```
# Deletes specified note
```

```
delete '/:url/'
```

```
# List all pages in database
```

```
get '/all/'
```

```
# Error handling
```

```
not-found
```

```
error
```

Đó là một danh sách hay ho. Ứng dụng của tôi sẽ xoay quanh những yêu cầu mà tôi tạo ra cho mỗi tuyến.

# Slug, Slug mọi nơi!

Bạn còn nhớ khi tôi nói rằng, cơ sở dữ liệu giống như một chồng thẻ ghi nhớ ma thuật, và bạn có thể tìm chúng theo bất cứ cách nào mà bạn muốn không? Chúng ta cần một cách để tìm kiếm bản ghi các trang cụ thể, đó chính là lý do tại sao bạn thấy: url trong những tuyến này. Nội dung của tham số url sẽ nói cho cơ sở dữ liệu biết phải lấy lại bản ghi nào.

Chúng ta có thể sử dụng tiêu đề trang như một tham số, nhưng có một vấn đề, đó là trình duyệt web không thích những thứ như chỗ trống, chữ cái viết hoa và các ký tự đặc biệt (kiểu như \$ và %) xuất hiện trong địa chỉ web. Trong khi đó, tiêu đề trang rất có thể sẽ có những thứ đó, vì vậy bạn cần có một cách để loại bỏ chúng.

Chuỗi ký tự duy nhất xác định trang web được gọi là *slug*. Phần slug trang web của tôi được thực hiện dựa trên tiêu đề trang với những nguyên tắc sau để đảm bảo địa chỉ web thân thiện:

1. Tất cả các chữ cái đều được viết thường.
2. Không có ký tự đặc biệt – chỉ có chữ cái và số.
3. Không có khoảng trống – tất cả các khoảng trống nên được thay thế bằng dấu gạch ngang.

Phương pháp để thực hiện việc này là tạo ra một cách chấp nhận tiêu đề trang, sau đó chuyển nó sang định dạng slug. Đảm nhận việc chăm sóc những sự thay thế đó là một chương trình có tên gọi là *regular expressions*, nó biến đổi hoặc tìm kiếm văn bản trong chuỗi ký tự dựa trên các nguyên tắc được định sẵn.

Giờ tôi đã có thể sử dụng phương pháp *slug hóa* để biến đổi một chuỗi ký tự kiểu như “Tiêu đề Trang” thành “tiêu đề-trang”, để phù hợp với việc sử dụng trong địa chỉ trang web.

Ngoài ra, nếu chúng ta lưu trữ slug cùng với tiêu đề của trang, chúng ta có thể sử dụng nó để lấy lại bản ghi trang bằng cách sử dụng tham số: url.

Tạo trang

Tôi bắt đầu bằng việc viết ra danh sách các tuyến cần thiết. Tuyến “trang chủ” rất dễ: tôi điều hướng nó tới /home/slug vì tôi muốn trang chủ thể hiện bản ghi chính.

Tuyến “tạo trang” được gắn kết với một mẫu nhỏ ở phần trên cùng của thanh phụ. Người sử dụng nhập tiêu đề trang vào mẫu này, sau đó kích vào nút. Hệ thống nắm bắt tiêu đề trang, slug hóa nó, sau đó lưu tiêu đề, slug, và thời gian tạo ra cơ sở dữ liệu. Sau đó, hệ thống sẽ gửi yêu cầu GET có chứa slug, hiển thị trang mới.

Có một chi tiết nhỏ rất quan trọng trong tuyến “tạo trang”: chuyện gì sẽ xảy ra nếu trang đó đã có rồi? Tôi không muốn viết đề lên một trang đã có chứa dữ liệu. Vô tình đánh mất dữ liệu là điều không thể chấp nhận được.

May mắn là DataMapper giải quyết được vấn đề đặc biệt này bằng một phương pháp được tích hợp sẵn .first-or-create (lần đầu-hay-tạo ra). Trước khi tạo trang, DataMapper sẽ kiểm tra xem trang đó đã có hay chưa. Nếu trang đó đã tồn tại, DataMapper sẽ không viết đề lên nó, và thay vào đó, Sinatra sẽ điều hướng trình duyệt tới trang web đã có đó. Vấn đề đã được giải quyết.

## Chỉnh sửa trang

Chỉnh sửa trang liên quan tới hai tuyến riêng biệt. Tuyến đầu tiên nhận (GET) trang người sử dụng muốn, chỉnh sửa, sau đó hiển thị nội dung trong bản ghi của trang dưới dạng mà người sử dụng có thể chỉnh sửa được.

Để hiển thị những trang này, ứng dụng dựa vào một cú pháp mẫu được gọi là ERB – yêu cầu cơ bản của HTML, và Ruby. ERB cho phép người lập trình viết HTML có chứa các thành tố có thể thay đổi được. Vì ERB xử lý trang trước khi nó được hiển thị tới người sử dụng, nên nó có thể thay đổi văn bản trên trang mỗi khi trang được tải, dựa trên các yêu cầu trong mẫu của Ruby.

Nút lưu (save) trên màn hình chỉnh sửa có nhiệm vụ gửi yêu cầu đăng (POST) tới ứng dụng để cập nhật bản ghi của trang.

## Xóa trang

Việc xóa trang đòi hỏi phải lưu ý đến một điểm: Cần phải nhớ, vô tình làm mất dữ liệu là không thể chấp nhận được. Nếu bạn chuẩn bị xóa một trang, bạn cần

phải chắc chắn rằng người sử dụng thực sự có ý định xóa trang đó.

Sẽ là sai nếu liên kết trực tiếp nút xóa với tuyến xóa (DELETE) trong ứng dụng. Cách đó sẽ xóa trang ngay cả khi người sử dụng vô tình kích vào nút.

Chiến lược tốt hơn là sử dụng quá trình gồm hai giai đoạn. Việc nhấn vào nút xóa trên trang đưa người sử dụng tới một màn hình xác nhận tiêu đề của trang mà người sử dụng muốn xóa. Nếu người sử dụng muốn thực hiện, họ có thể kích vào nút xác nhận để gửi yêu cầu XÓA (DELETE). Nếu người sử dụng vô tình nhấn vào nút xóa, họ có thể nhận được nút quay lại (back) hoặc hủy (cancel) trên trình duyệt web.

## Lần đầu khởi chạy ứng dụng

Các tính năng cơ bản đã được đặt vào đúng vị trí, nhưng tôi lại gặp phải một vấn đề: Khi tôi kiểm tra ứng dụng bằng cách truy cập vào trang chủ, ngay lập tức tôi nhận được thông báo báo lỗi. Chương trình đang cố gắng tìm kiếm bản ghi trang chủ trong cơ sở dữ liệu, nhưng nó không tồn tại vì tôi chỉ vừa mới bắt đầu ứng dụng!

Giải pháp cho vấn đề này là tạo ra một “quá trình quản lý một lần”, sử dụng chương trình có tên gọi là Rake. Các chương trình Rake được lưu giữ trong tập tin Rake (Rakefile), thuộc thư mục gốc của ứng dụng.

Rakefile hoạt động như những ứng dụng Ruby thông thường, chỉ khác một điều: Chúng tồn tại bên ngoài chương trình chính của bạn, và bạn cần phải thao tác bằng tay để chạy các yêu cầu đó.

Đó chính là điều khiến Rake trở nên hữu dụng khi thực hiện những việc như thêm các thông tin mặc định cho cơ sở dữ liệu trước khi chúng ta chính thức chạy chương trình thực. Tôi sao chép phần quan trọng của application.rb vào Rakefile, sau đó đưa ra yêu cầu tạo một trang “chủ” mới trong cơ sở dữ liệu. Sau đó, tất cả những gì tôi cần phải làm là chạy yêu cầu này một lần:

```
$ rake setup
```

Rake tạo ra bản ghi trang “chủ”, và ứng dụng của tôi không còn báo lỗi khi khởi động nữa. Khi tôi đẩy ứng dụng này lên Heroku, tôi sẽ chạy yêu cầu Rake từ xa để cài đặt cơ sở dữ liệu trước khi tôi cố gắng sử dụng ứng dụng đó.

Đến lúc này, chúng ta đã đặt được tất cả các đặc tính quan trọng vào đúng vị trí của chúng. Giờ là lúc để thêm một số điều thú vị.

## Thêm phần hỗ trợ thanh bên

Tôi vừa mới nhận ra là tôi đã không đưa cách tạo thêm trang vào thanh bên như ý định ban đầu của mình. Đặc tính này đòi hỏi phải có Boolean, vì nó chỉ có hai giá trị: Trang này được cho là sẽ hiển thị ở thanh bên, hoặc là không.

Tôi đã thêm đoạn mã này vào trường phân loại của DataMapper:

```
Property : sidebar, Boolean, :default => false
```

Tôi khởi động lại ứng dụng, biên tập bản ghi, và toàn bộ chương trình bị phá hủy. Oái!

Tôi cố gắng xác định xem sai ở đâu, nhưng tôi không gặp may. Sau khi bôi tung tài liệu hướng dẫn về DataMapper và tìm kiếm trên Stack Overflow, tôi đã phát hiện ra là sử dụng các biến Boolean theo cách này không phù hợp với dạng HTML. Cần phải có một cách tiếp cận khác:

```
property :sidebar, Enum[ :yes, :no ], :default => no
```

Đây là một cách khác để làm điều tương tự. Enum, viết tắt của từ “enumerate” (đếm, liệt kê), tạo ra một danh sách các lựa chọn, và định dạng sẽ quyết định cần phải lưu lựa chọn nào trong cơ sở dữ liệu.

## Thêm phần bảo an

Thế còn thông tin đăng nhập thì sao? Nếu tôi đẩy ứng dụng này lên Heroku mà không đòi hỏi tên đăng nhập và mật khẩu thì bất cứ ai cũng có thể nhìn thấy những gì tôi lưu giữ trong cơ sở dữ liệu.

Hóa ra là các trình duyệt web hiện đại đều hỗ trợ một giao thức bảo an được gọi là Thẩm định quyền cơ bản HTTP. Đây là một cách đơn giản để yêu cầu người sử dụng nhập tên đăng nhập và mật khẩu trước khi họ có thể sử dụng chương trình. Nếu người sử dụng không thể cung cấp được thông tin chứng tỏ quyền được truy cập thì họ sẽ được điều hướng sang trang báo lỗi.

Dưới đây là đoạn mã cho phép thẩm định quyền cơ bản trong Sinatra:



```
user Rack::Auth::Basic, "Restricted Area" do |username, password|  
  [username, password] == [ENV['ADMIN_USER'], ENV['ADMIN_PASS']]  
end
```

Trong trường hợp này, tôi đang lưu giữ tên đăng nhập và mật khẩu thực sự trên Heroku như các *biến số môi trường* mà tôi có thể đặt bằng một lệnh Terminal. Điều đó cho phép tôi sử dụng cùng một đoạn mã cho nhiều ứng dụng khác nhau, cũng như cho phép bạn nhìn thấy đoạn mã này nhưng không cho bạn thấy mật khẩu của tôi!

Đây cũng là một ví dụ cho thấy tại sao việc biết Sinatra được xây dựng trên nền Rack lại là điều quan trọng. Có rất nhiều thư viện như Rack::Auth::Basic, và tôi có thể sử dụng bất cứ thư viện nào trong số đó với Sinatra.

## Mã hoàn thiện

Dưới đây là phần mở đầu trong tập tin Readme của Codex:

*Codex là một ứng dụng web tham khảo dành cho một người sử dụng được viết bằng Ruby. Codex sử dụng Sinatra và DataMapper để tạo ra, lưu giữ, cập nhật và xóa các bản ghi trang từ cơ sở dữ liệu đơn giản có tên là Postgres. Ứng dụng đã sẵn sàng để triển khai ngay lập tức trên Heroku.*

*Định dạng Markdown được sử dụng cho tất cả các trang, khiến cho việc viết các trang phức tạp trở nên dễ dàng hơn nhờ công cụ đánh dấu đơn giản. Thảm định quyền HTTP và SSL bắt buộc dành cho tất cả các lưu lượng (traffic), giữ cho thông tin của bạn được bảo mật. Giao thức tự môi (bootstrap) giúp cho các trang của bạn trông đẹp và cuốn hút hơn.*

Tài liệu Readme còn tiếp tục với những hướng dẫn chi tiết về cách cài đặt ứng dụng trên Heroku. “Phát triển phần mềm dựa trên Readme” là một cách tiếp cận rất tốt.

Tổng cộng, Codex khiến tôi mất 10 tiếng để xây dựng. Chính vì vậy mà tổng thời gian đầu tư cho việc học lập trình của tôi đã được nâng lên con số 20 giờ. (Tôi mất nhiều thời gian để viết chương này hơn so với việc viết một ứng dụng thực sự.)

Sau khi tạo ra Codex, tôi đã tham dự một buổi gặp mặt những người lập trình sử dụng Ruby ở địa phương, và đã tình nguyện giải thích cách chương trình hoạt động. Phản hồi rất tích cực, và tôi đã được khen ngợi vì phần mã viết gọn gàng, cô đọng và dễ hiểu. Một trong số những người tham gia còn nhận xét rằng, chất lượng phần mã còn tốt hơn các dự án được những lập trình viên chuyên nghiệp viết mà ông ấy đã thấy.

Nhiệm vụ đã hoàn thành.

## Tóm lược lại phương pháp

Hãy cùng tổng kết lại những điều cốt lõi trong phương pháp mà tôi đã sử dụng để học cách lập trình:

- Tôi đã dành một chút thời gian để học lập trình và ứng dụng web nói chung là gì, sau đó chia nhỏ những kỹ năng này thành những kỹ năng nhỏ hơn, dễ hiểu và dễ thực hành hơn.
- Tôi đã đặt ra mức độ thành thạo mục tiêu bằng cách chọn hai dự án cụ thể mà tôi muốn tạo ra, sau đó xác định những dự án này sẽ như thế nào khi được hoàn thành.
- Tôi đã chia nhỏ những dự án này thành những bước nhỏ hơn, sau đó xác định những bước nhỏ nào có vẻ quan trọng nhất.
- Tôi đảm bảo rằng tôi có những công cụ cần thiết để làm việc (ví dụ như phiên bản mới nhất của Ruby), và tôi có thể tìm và sử dụng được bất cứ công cụ phụ trợ nào mà tôi cần.
- Tôi đã tìm được một số nguồn đáng tin về lập trình, nhưng tôi chỉ đọc lướt qua phần hướng dẫn để bắt tay vào viết các chương trình thực sự.
- Trước tiên tôi làm việc với những bước nhỏ quan trọng nhất, như xác định cách để kiểm tra các chương trình trên máy tính, đây ứng dụng đã hoàn thành lên máy chủ sản xuất,...
- Tôi đã sử dụng các ví dụ tham khảo để bắt đầu và gây dựng sự tự tin, sau đó thử nghiệm nhiều cách tiếp cận khác nhau để xác định làm cách nào để lập

trình được những tính năng mà tôi muốn.

Khi tôi mắc lỗi, chương trình sẽ bị hỏng và gửi cho tôi một thông điệp báo lỗi, tạo ra một vòng lặp phản hồi nhanh.

Sau khi mắc lỗi, tôi đã thử nghiệm một số cách để sửa lỗi. Nếu tôi không thể tự mình sửa được lỗi, tôi sẽ tìm kiếm sự giúp đỡ.

Tôi tiếp tục sử dụng cách tiếp cận xây dựng/ thử nghiệm/ sửa sai cho tới khi các chương trình được hoàn thành.

Tổng thời gian sử dụng: khoảng 20 tiếng. 10 tiếng dùng để nghiên cứu, và 10 tiếng còn lại là để lập trình hai ứng dụng mà hiện tại đã được hoàn thành và đưa vào sử dụng.

## Từ đây tôi sẽ đi đâu tiếp

Tôi vẫn đang tiếp tục lập trình các ứng dụng web kể từ khi hoàn thành các dự án cơ bản này, tập trung vào những chương trình giúp cho việc điều hành doanh nghiệp của tôi trở nên dễ dàng hơn.

Tôi tự hào nói rằng toàn bộ công việc kinh doanh của tôi hiện tại đều chạy trên phần mềm mà tôi tự mình tạo ra. Các ứng dụng của tôi có thể thực hiện các giao dịch bằng thẻ tín dụng, gửi thư điện tử và quản lý việc truy cập trang web cho các khách hàng của tôi. Bằng cách học viết mã, giờ tôi đã có một đội quân robot nho nhỏ được lập trình để thực hiện mệnh lệnh của mình.

Mất bao lâu để viết được tất cả những chương trình đó? Tổng cộng là khoảng 90 giờ, bao gồm cả 20 giờ mà tôi đã trình bày chi tiết trong chương này.

Và đây, thêm một lợi ích nữa: Bất cứ khi nào tôi thấy một lĩnh vực kinh doanh của mình có vẻ nhàm chán và lặp đi lặp lại, tôi sẽ bắt đầu nghĩ tới việc viết mã. Chương trình có thể xử lý vấn đề này sẽ hoạt động như thế nào? Thường thì sẽ có cách để hệ thống hóa quá trình đó theo cách khiến cho cuộc sống thường nhật của tôi trở nên dễ dàng hơn.

Tôi cũng sẽ học thêm những thủ thuật mới, kiểu như tùy chỉnh máy tính để việc lập trình được nhanh hơn. Tôi sẽ học phím tắt trên bàn phím khi chỉnh

sửa văn bản để tiết kiệm thời gian, và tôi vừa mới nâng cấp Terminal lên iTerom2 và Z-Shell để việc viết mã trở nên dễ dàng hơn.

Tôi vẫn đang luyện tập và chưa phải là chuyên gia, dù theo bất cứ tiêu chuẩn nào. Tôi phải nghiên cứu mọi thứ, và tôi vẫn phải mất một lúc mới giải quyết được vấn đề, lỗi sai và lỗi. Thật sự rất khó chịu.

Tuy nhiên, tôi vẫn đang tạo ra những chương trình có thể giải quyết các vấn đề theo cách trực diện và đáng tin. Đó thực sự là điều có ý nghĩa.

Tôi chiến đấu với chiếc máy tính, và tôi thắng.

# 6

## GỖ 10 NGÓN

---

**BÀI HỌC: THÓI QUEN CŨ KHÔNG CẦN PHẢI CHẾT THẢM**  
**Bất cứ việc nào đáng làm ban đầu cũng có vẻ không đáng làm.**

**–RAY CONGDON**

...

Tính tới thời điểm này, tôi đã học được nhiều kỹ năng mới ở những lĩnh vực mà tôi ít có kinh nghiệm từ trước. Lúc đầu, thiếu kinh nghiệm chắc chắn là một rào cản, nhưng ít nhất não bộ của tôi cũng không chủ động can thiệp vào quá trình học.

Chuyện gì sẽ xảy ra khi bạn luyện tập một thứ gì đó hoàn toàn mới và cùng với đó, bộ não của bạn còn phản kháng nữa?

Tốc độ của công nghệ hiện đại khiến cho việc học lại trở nên rất phổ biến: Chuyện gì sẽ xảy ra nếu chương trình phần mềm bạn sử dụng để thực hiện công việc thay đổi, hay một chương trình mới trở thành cách hiệu quả nhất để hoàn thành một nhiệm vụ? Chuyện gì sẽ xảy ra nếu bạn nhận được một công việc mới đòi hỏi bạn phải làm việc với những công cụ mới? Học lại kỹ năng một cách nhanh chóng cũng quan trọng không kém việc học một thứ mới.

Tôi tò mò không biết cảm giác mình học lại một thứ gì đó thực sự quan trọng sẽ như thế nào?

Tôi bắt đầu liệt kê những việc mà tôi đã biết cách làm, sau đó đào xới danh sách để tìm những kỹ năng mà (1) tôi đã giỏi, và (2) có một số phương pháp để thu được kết quả như nhau. Không tốn quá nhiều thời gian để tôi tìm ra một ứng viên đầy hứa hẹn.

Kỹ năng nào ư? Gỗ 10 ngón.

## Cuộc sống phía sau bàn phím

Theo David Allen, tác giả của cuốn sách *Để hoàn thành công việc* (Getting Things Done) (2002), nếu công việc của bạn đòi hỏi phải sử dụng bất cứ loại máy tính nào, thì việc học cách gõ bàn phím bằng 10 ngón là điều quan trọng nhất bạn có thể làm để cải thiện năng suất của mình.

Tốc độ gõ thông thường là khoảng 70 từ trên một phút. Nếu bạn không thể gõ được ít nhất 60 từ trên một phút với tỉ lệ gõ sai thấp, có nghĩa là bạn đã hạn chế khả năng làm việc hiệu quả của chính bản thân. Nỗ lực bạn dành cho việc gõ phím càng ít, thì thời gian và năng lượng bạn có để thực hiện những nhiệm vụ có giá trị cao hơn càng nhiều.

Tôi đã gõ 10 ngón được khoảng 17 năm, có lẽ còn lâu hơn. Khi còn đi học, học sinh nào cũng được yêu cầu tham gia một khóa học gõ 10 ngón, và tôi nhớ là mình đã cảm thấy rất buồn chán. Tôi đã biết cách gõ nhanh, vì thế tôi hoàn thành các bài tập chỉ trong vài phút, sau đó dành thời gian còn lại trên lớp để xác định làm cách nào máy xử lý văn bản nhả ra những ký tự trông có vẻ là lạ, kiểu như ký hiệu (§), đánh dấu phần (§) và dấu chấm than (!).

Tôi học cách gõ nhanh 10 ngón chủ yếu là nhờ dành rất nhiều thời gian làm việc với máy tính vào thời gian rảnh rỗi. Việc luyện gõ nhanh của tôi bắt nguồn từ yếu tố khách quan chứ không phải do tôi chủ tâm học: Tôi hoàn toàn không có ý định cải thiện tốc độ hay độ chính xác. Chỉ là tôi hay sử dụng máy tính, và trong quá trình đó, tôi đã học cách đánh máy.

Điều đó không có nghĩa là tôi nói rằng mình đã giỏi. Các ngón tay của tôi lướt khắp bàn phím thay vì dành hầu hết thời gian ở *hàng phím cơ sở*, những phím ở giữa bàn phím. Phương pháp của tôi không phải là gõ chính xác, mà hoàn thành công việc mới là tất cả những gì tôi quan tâm.

Dù rằng kiểu gõ của tôi không phải tuyệt vời, nhưng nó lại thiết thực. Trong công việc, tôi sử dụng máy tính rất nhiều, và tốc độ cũng như độ chính xác khi đánh máy của tôi giúp tôi hoàn thành công việc.



Tuy nhiên, thỉnh thoảng, tôi lại vô tình đọc được một bài báo về các cách bố trí bàn phím thay thế: Sắp xếp các phím khác so với QWERTY – cách bố trí mặc định phổ biến của phần lớn bàn phím ngôn ngữ tiếng Anh được sản xuất mỗi năm.

Có người cho rằng QWERTY được thiết kế cực kỳ không hiệu quả và là một thiết kế kinh khủng. Có nhiều cách khác để sắp xếp các phím trên bàn phím sao cho chúng có thể hỗ trợ người sử dụng gõ nhanh hơn, chính xác hơn và ít tốn công sức hơn.

Nỗ lực tích lũy là một vấn đề lớn: chấn thương căng thẳng lặp đi lặp lại (RSI) và hội chứng ống cổ tay là những rối loạn ở tay và cổ tay phổ biến, có thể do hoặc có thể bị nặng thêm vì đánh máy. Dù tôi chưa gặp phải những triệu chứng đó, nhưng tôi có vài người bạn thân đã bị, và điều đó chẳng thú vị chút nào. Tôi muốn tránh những rối loạn đó nếu có thể.

Tôi dự định sẽ viết và lập trình nhiều năm sau này nữa, và trừ khi nhận dạng giọng nói suy nghĩ-thành-văn bản trở thành phương pháp chính được máy tính sử dụng, nếu không tôi vẫn còn đánh máy trong tương lai. Có lẽ quan tâm lớn nhất của tôi là học đánh máy theo cách hiệu quả nhất có thể, ngay cả khi điều đó gắn liền với sự bối rối và không thoải mái trong một thời gian ngắn.

Tạm biệt QWERTY, tôi chuẩn bị học lại cách gõ 10 ngón đây.

Cách bố trí bàn phím kiểu QWERTY trở thành tiêu chuẩn “toàn cầu” như thế nào

Ngược với những hiểu biết thông thường, định dạng bàn phím kiểu QWERTY không phải được thiết kế để làm giảm tốc độ của người đánh máy, mà là một

giải pháp cho vấn đề kỹ thuật cơ khí.

C. L. Sholes, người được tin là đã phát minh ra cách bố trí bàn phím kiểu QWERTY cho máy đánh chữ, đã tạo ra mẫu dùng thử vào năm 1868. Sholes không phải là người đầu tiên tạo ra máy chữ, vì có ít nhất 51 nhà sáng chế khác đã từng thử trước ông, và Sholes đã nghiên cứu các công trình của họ, tích hợp nhiều tính năng trong thiết kế của họ vào thiết kế của ông.

Các phím trong bản mẫu đầu tiên của Sholes được sắp xếp theo thứ tự bảng chữ cái. Thời điểm đó, không ai tưởng tượng rằng người ta sẽ (hoặc có thể) học cách đánh nhanh bằng 10 ngón tay. Xếp các phím theo thứ tự bảng chữ cái giúp những người sử dụng chưa qua đào tạo có thể tìm thấy chữ cái thích hợp khi “mỏ cò” bằng hai ngón tay trở.

Tuy nhiên, có một vấn đề lớn với bản mẫu này, đó là: Cần gõ chữ sẽ bị dính lại với nhau khi các chữ cái liền kề được gõ quá nhanh. Những chữ cái thường được sử dụng nhiều nhất trong tiếng Anh là các nguyên âm AOEUI và các phụ âm DHTNS. Trên bàn phím, chữ S và chữ T nằm ngay cạnh nhau. Nếu một cần gõ chữ đi lên, còn một cần gõ chữ lại đi xuống thì máy sẽ bị kẹt, buộc người đánh máy phải dừng lại và gỡ cần gõ chữ bằng tay.

Để xử lý vấn đề khó chịu lặp đi lặp lại này, Sholes đã nhờ tới sự giúp đỡ của một thầy giáo có tên là Amos Densmore. Densmore đã tiến hành nghiên cứu nhanh các chữ cái hay được sử dụng trong tiếng Anh, sau đó, Sholes đã sử dụng nghiên cứu này để đặt các cặp chữ cái hay đi với nhau, kiểu như TH, sang hai phía của cần gõ chữ để tránh trường hợp bị va chạm.

Chiến lược này không giải quyết được vấn đề triệt để, nhưng nó đã cải thiện được đáng kể (vấn đề), tới mức Sholes đã đăng ký bản quyền cho thiết kế của mình năm 1872. E. Remington & Sons, một công ty nổi tiếng với việc sản xuất vũ khí khi đó đã mua lại bản quyền của Sholes năm 1873.

Sau khi thêm vào một số cải tiến cơ khí, như thêm phím Shift để người đánh máy có thể chuyển giữa viết hoa và viết thường các chữ cái, Remington bắt đầu sản xuất hàng loạt máy đánh chữ với bàn phím được bố trí kiểu QWERTY, với ý định bán ra thị trường kinh doanh vào năm 1874.

Remington không phải là công ty duy nhất bán máy đánh chữ. Những công ty khác, như Hammond và Blickensderfer, chào bán các thiết bị cạnh tranh, mỗi công ty lại có một cách bố trí bàn phím riêng của họ.



Thời điểm đó, các công ty phụ thuộc chủ yếu vào các bản ghi chép và thư từ dạng văn bản. Máy đánh chữ có thể tiết kiệm rất nhiều sức lực, nhưng chỉ khi người vận hành biết cách sử dụng máy. Để bán được hàng, các công ty phải giải quyết được nhu cầu đào tạo người đánh máy sao cho họ có thể sử dụng được cái máy “kỳ cục” đó.

Điều đó dẫn tới một động lực thị trường thú vị: Các công ty sản xuất máy đánh chữ tự tuyển và đào tạo người đánh máy, vận hành hiệu quả các trung tâm sắp xếp việc làm. Nếu chủ doanh nghiệp muốn thuê một người có thể đánh máy, họ sẽ gọi cho Remington – công ty có thể cung cấp cả máy đánh chữ và người có thể sử dụng thiết bị đó.

Các công ty cần người đánh máy và máy đánh chữ, còn Remington có thể cung cấp cả hai. Và khi một công ty nào đó cần một máy đánh chữ mới, hay một người đánh máy mới, cách hiệu quả nhất là mua một máy đánh chữ QWERTY mới và thuê một người đánh máy biết sử dụng QWERTY. Sau khoảng 60 năm, QWERTY đã trở thành tiêu chuẩn được công nhận trên thực tế, và các cách bố trí cạnh tranh với cách bố trí QWERTY đã dần biến mất. Sau nhiều thập kỷ, QWERTY đã thống lĩnh toàn thế giới.

## Cạnh tranh xuất hiện: Dvorak

Năm 1932, August Dvorak, giáo sư trường Đại học Washington đã nhận được phần thưởng 130.000 đô la từ Hội đồng Giáo dục Carnegie cho việc nghiên cứu thiết kế bàn phím. Một trong những nhân tố thú vị của nghiên cứu này là nhận ra: QWERTY được thiết kế để giải quyết một vấn đề kỹ thuật cơ khí đã không còn tồn tại nữa. Có còn cách nào tốt hơn để thiết kế bàn phím không?

Bốn năm sau, Dvorak đăng ký bản quyền cho sản phẩm Bàn phím đơn giản hóa Dvorak mà ông tuyên bố là siêu việt hơn những cách bố trí khác. Cơ sở của lời khẳng định này rất đơn giản: Cách bố trí của Dvorak là đặt những ký tự thường được sử dụng nhiều nhất ngay dưới các ngón tay của người sử dụng ở hàng phím cơ sở.



Dvorak cũng khẳng định cách bố trí này dễ học hơn, và tiếp tục các nghiên cứu với các tổ chức, đào tạo một lượng lớn những người đánh máy, mà chủ yếu là các tổ chức trong quân đội Mỹ. Kết quả thu được là: vì Dvorak tự mình thực hiện các nghiên cứu, nên rất nhiều phần trong kết quả bị nghi ngờ.

Một nghiên cứu được tiến hành độc lập của Cơ quan quản lý dịch vụ công Hoa Kỳ, cơ quan chịu trách nhiệm đào tạo người đánh máy cho chính phủ Hoa Kỳ, đã phát hiện ra rằng, những người đánh máy bằng bàn phím kiểu QWERTY đã qua đào tạo phải mất hơn 100 giờ để đạt được tốc độ đánh máy trước đó họ đã có, nếu họ được đào tạo lại với bàn phím Dvorak. Thế là quá lâu. Do đó, nghiên cứu đề xuất những người đánh máy cho chính quyền nên sử dụng bàn phím QWERTY. Các nhà máy và công ty sản xuất máy đánh chữ cũng theo đó mà làm.

Dvorak tiếp tục tồn tại như một lựa chọn phụ, nhưng ở góc độ tiêu chuẩn, nó hoàn toàn thất bại. Dù có rất nhiều lời khẳng định tích cực cho rằng nó hiệu quả hơn cách bố trí bàn phím kiểu QWERTY, nhưng nó vẫn không thể thay thế được vị trí của bàn phím kiểu QWERTY. Dvorak đã phai nhạt theo thời gian, và còn xa mới đứng được ở vị trí thứ hai.

## Thách thức mới xuất hiện: Colemak

QWERTY và Dvorak không phải là hai kiểu bố trí bàn phím tiếng Anh duy nhất tồn tại. Dù phần lớn những người đánh máy đều học QWERTY, nhưng các nhà sáng chế và người yêu thích vẫn tiếp tục công cuộc sáng tạo thiết kế bàn phím hàng thập kỷ qua. Hầu hết những cách bố trí này đều chưa bao giờ được công bố rộng rãi, nhưng năm 2006, nhờ internet, có một cách bố trí bàn phím hoàn toàn mới đã bắt đầu thu hút được sự chú ý của những người đánh máy ưa

mạo hiểm.

Chuyện là thế này. Shai Coleman, một lập trình viên máy tính, đã quyết định thử bắt tay vào thiết kế bàn phím. Mục đích của anh rất đơn giản, chỉ là tạo ra một cách bố trí bàn phím hiệu quả tương đương với cách bố trí bàn phím của Dvorak, nhưng dễ học hơn.

Một trong những vấn đề chính của Dvorak là nó thay đổi mọi thứ: mỗi phím chữ cái trong Dvorak lại khác hoàn toàn so với bàn phím QWERTY. Nếu lúc đầu bạn học cách đánh máy trên QWERTY, thì mức độ thay đổi khiến cho việc học Dvorak thực sự trở thành một cơn ác mộng.

Tệ hơn nữa hiện nay máy tính là công cụ đánh máy cơ bản, nên việc sử dụng các phím tắt để thực hiện các nhiệm vụ lặp đi lặp lại (như lưu một tập tin, cắt và dán văn bản,...) ngày càng trở nên phổ biến. Nếu bạn đã quen làm việc với một số cách kết hợp bàn phím nhất định, thì Dvorak thực sự khó chịu: không chỉ tất cả các chữ cái đều thay đổi vị trí, mà tất cả các phím tắt cũng khác.

Chiến lược của Coleman là kết hợp những phân tích của máy tính về bộ dữ liệu lớn liên quan tới các tài liệu tiếng Anh, với ý tưởng bảo tồn càng nhiều phím càng tốt để duy trì được các phím tắt như cũ. Phần bên trái của bàn phím cũng như hầu hết hàng dưới cùng gần như được giữ nguyên. Chỉ những phím quan trọng mới được thay đổi. Thuật toán của Coleman đề xuất hầu hết các thay đổi, chỉ để lại một số phím mơ hồ cần phải đánh giá.

Cuối cùng, cách bố trí mới của Coleman (được anh gọi là Colemak) đã thay đổi 17 phím so với cách bố trí tiêu chuẩn QWERTY. Dựa trên các con số, nó hiệu quả hơn hẳn so với QWERTY và hiệu quả hơn Dvorak một chút. Hơn nữa, vì Colemak chỉ đổi một số phím, nên có vẻ như những người sử dụng QWERTY sẽ dễ dàng học được hơn.

Coleman đã tạo ra một trang web ([colemak.com](http://colemak.com)), trong đó có các chi tiết về cách bố trí bàn phím mới này, cũng như có hướng dẫn về cách cài đặt và sử dụng bàn phím đó. So với nỗ lực tốn kém của Dvorak để phổ biến cách bố trí bàn phím của ông trong những cuộc họp với các tổ chức lớn, thì internet đã giúp cho việc quảng bá sự tồn tại của bàn phím Colemak trực tiếp và rẻ hơn.

Kết quả là Colemak giờ đã là cách bố trí bàn phím tiếng Anh phổ biến thứ ba, chỉ đứng sau QWERTY và Dvorak. Cách bố trí này giờ đã được cài đặt vào hầu hết các hệ điều hành của máy tính mới, và đã có hàng nghìn người sử dụng trên khắp thế giới. Hoàn toàn không tệ đối với một bàn phím được phát triển

bởi một người say mê.

## Colemak trông như thế nào?

Dưới đây là sơ đồ bố trí bàn phím Colemak:



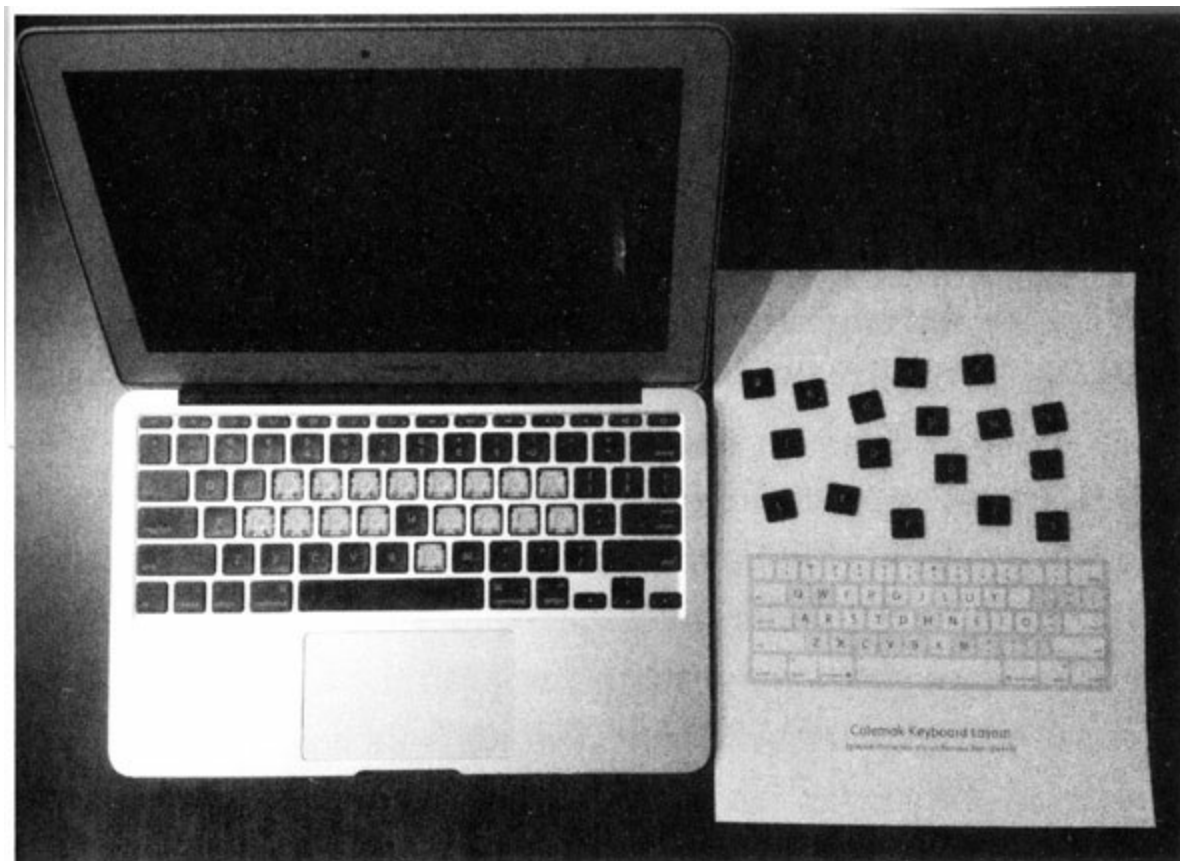
Ngoài việc thay đổi vị trí của một số phím ký tự được sử dụng thường xuyên nhất, Colemak còn biến phím Caps Lock thành phím Delete thứ hai. Thay đổi này là một trong những cải tiến lớn nhất của Colemak. Không có ai (ngoài những kẻ phá hoại mạng internet) sử dụng phím Caps Lock một cách thường xuyên, nhưng phím này lại chiếm một vị trí “đắc địa”: Ở vị trí thuận tiện để ngón út của bàn tay trái có thể dễ dàng chạm tới, và ở ngay cạnh phím A trên hàng phím cơ sở.

Việc đổi phím Caps Lock thành Delete giúp những người sử dụng bàn phím Colemak có thể sửa lỗi sai, mà không phải di chuyển bàn tay phải ra khỏi hàng phím cơ sở, để nhấn vào phím Delete ở tận trên cùng bên phải của bàn phím. Điều này giúp tiết kiệm được rất nhiều công sức. Chỉ một thay đổi nhỏ đó thôi đã giúp giảm được từ 15 đến 20% khoảng cách di chuyển tay so với QWERTY. Càng nhiều lỗi sai được bạn sửa bằng ngón tay út của tay trái thì Colemak càng trở nên hiệu quả.

## Thay đổi máy tính của tôi

May mắn là thật dễ dàng để chỉnh bàn phím Apple và sắp xếp lại chúng mà không làm ảnh hưởng tới máy. Vì Apple sử dụng bàn phím kiểu “Chiclet” có kích thước giống nhau, nên việc chuyển bàn phím sang định dạng Colemak chỉ

mất khoảng năm phút.



Bằng cách sử dụng một chiếc tuốc-nơ-vít có rãnh nhỏ, tôi đã nhẹ nhàng bẩy mặt trên cùng bên phải của mỗi phím lên, luồn đầu của tuốc-nơ-vít vào, sau đó di chuyển nó dọc theo mép bên phải của phím. Khi tuốc-nơ-vít đã vào được đến phần giữa của phím, phím sẽ hủy cơ chế “kéo” được tích hợp sẵn trong bàn phím, khiến cho phím nảy ra ngoài khi được nhấn.

Khi bạn đã hiểu rõ, sẽ rất dễ để sắp xếp các phím cần phải thay đổi. Sau đó, tôi đã sử dụng sơ đồ bố trí bàn phím Colemak để xếp lại các phím vào bàn phím theo đúng vị trí. Tất cả những gì cần làm chỉ là ấn nhẹ nhàng, và các phím sẽ gắn chặt vào chỗ của chúng.

Đó chỉ là một quá trình đơn giản, nhưng cần phải lưu ý rằng việc này rất có thể sẽ làm máy tính bị mất quyền được bảo hành. Vì lý do an toàn, tôi đã làm thử với một bàn phím Apple không dây dự phòng trước, sau đó mới thay đổi các phím trên chiếc MacBook Air của mình, khi đã thấy tự tin với quá trình đó.

Giờ thì tôi đã có bàn phím Colemak trên máy tính xách tay. Thật tuyệt, đúng không?

# Tôi đánh máy nhanh cỡ nào?

Lúc này bàn phím của tôi đã sẵn sàng, tôi đã tiến gần tới việc tạo ra một thay đổi lớn. Tuy nhiên, trước khi tiếp tục, tôi muốn biết hiện tại tôi đánh máy nhanh cỡ nào. Điều này sẽ giúp tôi xác định được mức độ thành thạo mục tiêu.

Trong trường hợp này, mức độ thành thạo mục tiêu của tôi rất đơn giản: Tôi muốn đạt được tốc độ đánh máy bằng Colemak nhanh như bằng QWERTY càng sớm càng tốt. Tôi cũng không kỳ vọng là sẽ vượt qua được tốc độ đó, vì tốc độ đánh máy không phải là một yếu tố hạn chế trong công việc của tôi. Tôi chỉ muốn có thể đánh nhanh giống như trước đây mà không phải tốn nhiều công sức.

Thông tin duy nhất mà tôi thực sự cần trước khi thay đổi là thông tin về tốc độ đánh máy của tôi, vì thế tôi đã tìm bài kiểm tra đánh giá tốc độ đánh máy cơ bản trên mạng.

Bài kiểm tra này rất đơn giản: Khi bạn nhấn nút Bắt đầu (Start), chương trình sẽ đưa ra 100 từ ngẫu nhiên từ một cuốn sách cũ nào đó. Nhiệm vụ của bạn là phải đánh máy đoạn văn bản đó nhanh nhất có thể, với càng ít lỗi càng tốt. Khi hoàn thành, bạn chỉ cần nhấn nút Dừng (Stop), và chương trình sẽ cho bạn biết tốc độ đánh máy và tỉ lệ đánh sai của bạn.

Khi đánh xong, tôi nhấn nút Dừng, và nhận được kết quả là 61 từ/phút, độ chính xác là 100%, không có lỗi sai. Cũng không tệ, tốc độ đánh máy của tôi đạt đến đúng ngưỡng mà David Allen đưa ra. Tôi không phải là một con quỷ tốc độ, nhưng tôi có thể đánh máy khá tốt, đủ để hoàn thành công việc.

Vậy là tôi đã có mọi thứ tôi cần: Tôi biết mình đánh máy nhanh như thế nào, bàn phím của tôi đã ở định dạng Colemak, và tôi biết cách để kích hoạt cách bố trí đó trong hệ điều hành. Chẳng còn gì cần phải chuẩn bị nữa.

Thế đây. Tôi đã sẵn sàng bỏ QWERTY lại phía sau chưa nhỉ?

## Lướt trên bàn phím

Hoặc là bây giờ, hoặc là không bao giờ. Tôi chuyển máy tính sang dạng Colemak, sau đó đóng bảng cài đặt lại. Kể từ giờ phút này, tôi không thể đánh máy bất cứ chữ nào bằng QWERTY cho tới khi tôi hoàn thành cuộc thử nghiệm của mình.

Trình duyệt web của tôi vẫn còn bài kiểm tra tốc độ đánh máy. Tôi bật đồng hồ đếm giờ và bắt đầu gõ.

Và đây là ấn tượng đầu tiên của tôi: #%&@.

Có vẻ hơi cường điệu quá, nhưng *có cảm giác như tôi đã mất đi một phần não bộ*.

Tôi đã quen với việc đánh máy trực tiếp các từ xuất hiện trong tâm trí mình (nghĩ từ nào đánh từ ấy). Bây giờ tôi không biết vị trí của bất cứ phím nào. Tôi phải “truy tìm” hầu hết các phím, kể cả những phím không thay đổi vị trí so với QWERTY. Thật nực cười!

Tôi nhìn thấy thứ gì đó trên màn hình, và các ngón tay của tôi di chuyển một cách vô thức, vì thế tôi đánh máy một cách vụng về, đồng nghĩa là sau đó tôi lại phải xóa đi. Phải mất tới vài giây tôi mới gõ được một từ đơn giản, và tôi lo là tôi đang làm mòn phím Delete.

Mỗi từ lại là một cuộc vật lộn mới. Tôi nhìn đồng hồ, và nhận ra mình đã mất tới vài phút chỉ để gõ được hai câu. Thậm chí, tôi còn chưa làm được một nửa. Tôi thực sự nghĩ tới việc bỏ cuộc, nhưng vẫn tiếp tục nhấn nút. Thời gian trôi qua, tôi gần như muốn bỏ cuộc ít nhất 10 lần.

Cuối cùng, mất khoảng 20 phút để tôi đánh được đoạn văn bản có 100 từ đó. Vậy thì tôi sẽ trả lời thư điện tử như thế nào đây? Tôi sẽ làm việc ra sao? Tôi có thể làm được gì hả trời?

## Tôi là kẻ thù của chính mình

Đây là rào cản chính ngăn cản mọi người với việc học cách sử dụng một bàn phím được bố trí theo cách mới. Thay đổi về mặt kỹ thuật không là gì so với thay đổi về *cảm xúc*.

Khi bạn đã quen với một tốc độ cụ thể nào đó, hoặc khi dễ dàng hoàn thành một nhiệm vụ, thì bất cứ thứ gì cũng bớt đáng sợ hơn. Điều này đặc biệt đúng với đánh máy: Nếu bạn đã quen với việc đánh máy mà không phải tốn chút công sức nào, và đột nhiên lại phải tốn rất nhiều công sức, thì việc đánh máy sẽ trở thành một việc kinh khủng. Tệ hơn nữa là bạn sẽ có suy nghĩ nếu quay trở lại với cách bạn vẫn quen thì mọi thứ sẽ tốt hơn rất nhiều.

Lúc này, bộ não chẳng giúp được gì cho chúng ta: Bộ não thường có xu hướng cho rằng những gì chúng ta đang phải trải qua cũng sẽ tiếp tục như thế trong tương lai. Lúc này, bộ não của chúng ta đang mê muội. Nếu tôi chỉ có thể gõ được năm từ trong một phút thì tôi không bao giờ có thể làm việc được nữa! Sự nghiệp của tôi sẽ tiêu tan! Gia đình tôi sẽ chết đói!

Tất nhiên là điều đó không đúng, nhưng nó lại có vẻ đúng ở hiện tại. Cảm giác đó mới chính là rào cản lớn nhất đối với việc học.

Tôi tắt máy tính và lấy ra một cuốn sổ ghi chép cùng cây bút. Tôi cần một kế hoạch để tăng tốc độ đánh máy khi sử dụng Colemak, và tôi cần nó *ngay lập tức*.

## Vẽ lại bản đồ tư duy

Yêu cầu đầu tiên và cấp bách nhất của việc này là tôi không biết các phím mới được đặt ở đâu. Đúng là tôi có bảng tham khảo vị trí các phím đã được in ra, và các phím trên bàn phím được bố trí theo dạng bàn phím Colemak, nhưng bộ não của tôi lúc này lại không thể khớp được khao khát gõ chữ với chuyển động của ngón tay.

Giúp não vẽ bản đồ cách bố trí mới dưới dạng các chuyển động của ngón tay là ưu tiên hàng đầu của tôi. Tôi cần phải càng thiết thực càng tốt.

May mắn là mọi người đã học cách gõ 10 ngón từ hàng thập kỷ nay, nên có rất nhiều công cụ hỗ trợ. Hướng dẫn cách đánh máy, như Mavis Beacon và Typing Trainer, đã có từ nhiều năm nay và cũng không đắt lắm.

Nhưng đáng tiếc là những chương trình này đều cho rằng bạn muốn học cách gõ 10 ngón bằng cách sử dụng bàn phím QWERTY, vì đó là lựa chọn an toàn nhất khi cho rằng những người sử dụng chương trình này muốn học cách sử dụng bàn phím tiêu chuẩn. Những chương trình này đều bắt đầu với các phím ở hàng phím cơ sở. Mà hàng phím cơ sở ở bàn phím QWERTY và Colemak hoàn toàn khác nhau, vì thế những tài liệu hướng dẫn về QWERTY chẳng giúp ích gì cho tôi.

May mắn là tôi có một “người dẫn đường”. Khi truy cập Hacker News trong lúc nghiên cứu chương trình, tôi đã tìm thấy một bài được đăng trên Keyzen, trang web huấn luyện đánh máy mã nguồn mở do Rye Terrell tạo ra. Terrell đã phát triển Keyzen để giúp những người lập trình học cách đánh máy nhanh



hơn. Chương trình được chạy trong trình duyệt web tiêu chuẩn, và có những dấu câu không phổ biến như ngoặc đơn, ngoặc kép hay dấu gạch ngang: Những dấu người lập trình thường xuyên sử dụng nhưng hầu hết những người dạy cách đánh máy lại bỏ qua. Terrell đã tải lên trọn bộ mã nguồn của Keyzen trên GitHub, và kêu gọi những người lập trình khác sử dụng hoặc thay đổi bộ mã nguồn đó.

Chương trình đó rất đơn giản: Nó hiển thị bộ 17 ký tự, bắt đầu bằng các chữ cái trong hàng phím cơ sở. Nhiệm vụ của bạn là gõ các ký tự đó theo đúng thứ tự.

Khi bạn gõ, chương trình sẽ phát ra hiệu ứng âm thanh. Ký tự đúng sẽ phát ra tiếng *tạch tạch* như máy đánh chữ cổ điển, còn ký tự sai sẽ phát ra tiếng *xoạch xoạch* và biến ký tự đó thành màu đỏ.

Khi bạn hoàn thành một tập hợp (từ hoặc ký tự), chương trình sẽ cung cấp thêm một tập hợp mới. Nếu bạn gõ sai, chương trình sẽ tự động đưa ký tự đó vào tập hợp tiếp theo mà bạn cần gõ. Còn nếu bạn gõ đúng tất cả các ký tự của ba tập hợp trong một hàng, bạn sẽ nghe thấy một tiếng “ding!” khiến bạn thấy thỏa mãn, và điều đó có nghĩa là bạn đã nâng trình độ của mình lên một bậc. Khi đó, Keyzen sẽ giới thiệu tiếp một ký tự mới, và quá trình huấn luyện lại tiếp tục.

## Thay đổi Keyzen

Theo mặc định, Keyzen, cũng như hầu hết các chương trình dạy đánh máy, đều hỗ trợ QWERTY. Tuy nhiên, vì chương trình là nguồn mở, nên tôi có thể điều chỉnh sao cho phù hợp với việc học Colemak.

Quá trình học lập trình của tôi có đạt phát huy tác dụng rồi đây.

Tôi sao chép đoạn mã nguồn của Keyzen, sau đó mở các tập tin của chương trình trên máy tính cá nhân. Bản thân chương trình rất đơn giản, nên rất dễ tìm ra phần chương trình kiểm soát ký tự nào sẽ được hiển thị, và cũng rất dễ để sắp xếp lại chúng theo bất cứ cách nào mà tôi muốn.

Chương trình gốc giới thiệu các ký tự trên hàng phím cơ sở của QWERTY trước tiên, bắt đầu với ngón trỏ, sau đó thêm các ký tự cho tới khi hàng phím cơ sở hết chỗ. Tiếp đến là hàng trên cùng, bắt đầu từ giữa bàn phím tới vùng ngoại biên, sau đó là hàng dưới cùng, cũng theo nguyên tắc như vậy.

Sử dụng Keyzen như một mô hình đào tạo, tôi đã chỉnh sửa chương trình để dùng bộ ký tự của Colemak. Kiểu phím vật lý vẫn giữ nguyên, nhưng tuần tự lúc này lại là dưới dạng Colemak chứ không phải QWERTY.

Khi tôi đóng chương trình, tôi đã có tài liệu hướng dẫn đánh máy bằng Colemak của riêng mình. Thành công rồi!

## Kỹ năng vận động tinh

Tôi đã có tài liệu hướng dẫn đánh máy đầu tiên của mình, và mục tiêu đầu tiên của tôi rất đơn giản, chỉ là: Nhận biết vị trí của các phím trên bàn phím.

Tôi mở Keyzen và bắt đầu đánh máy (có nghĩa là chỗ trống)

nn\_nn\_n

n\_nn\_n\_

nnn\_nn\_

tntt\_tn

t\_tntt\_

ttttnt

Cách luyện tập này chẳng có gì thú vị, nhưng lại cần thiết. Bằng cách rèn luyện cách xác định vị trí của mỗi ký tự theo trật tự nửa-ngẫu nhiên, tôi đã giúp cho não bộ của mình biến việc nhìn thấy (hoặc nghĩ tới) một ký tự thành chuyển động tinh của các ngón tay.

Ngoài ra, Keyzen còn giúp tôi học rất hiệu quả theo một số cách. Trước tiên, Colemak được thiết kế để đặt những chữ cái thường được sử dụng nhiều nhất vào hàng phím cơ sở dưới các ngón tay khỏe nhất. Keyzen giới thiệu những chữ cái này trước tiên, nên tôi có thể thành thạo ARSTDHNEIO trước QWZX YM.

Khi tôi mắc bất cứ lỗi sai nào, Keyzen lại đưa ra đúng ký tự đó trong dãy ký tự luyện tập. Kết quả là tôi đã dành hầu hết thời gian luyện tập những ký tự mà tôi thấy khó. Đây là học theo kiểu lặp lại cách quăng. Cách học này khi kết hợp với những phản hồi ngay lập tức có thể giúp cho việc luyện tập trở nên cực kỳ hiệu quả.

Tôi đã xây dựng lịch trình làm việc: Ít nhất là hai phiên, mỗi phiên 20 phút, giữa hai phiên có thời gian nghỉ giải lao ngắn vào cuối ngày, trước khi đi ngủ.

## Bỏ thói quen nhìn bàn phím

Lúc này tôi đã có bảy tiếng luyện tập làm vốn giắt lưng, tương đương với 45 phút mỗi tối. Tôi sẽ hoàn thành được toàn bộ bài luyện tập Keyzen không mấy khó khăn. Tôi vẫn gõ sai, nhưng những lỗi sai này càng lúc càng ít xuất hiện. Khi tôi làm lại bài kiểm tra tốc độ đánh máy, kết quả của tôi đã tốt hơn rất nhiều: 20 từ/phút.

Trong ngày, tôi còn nhận được thêm các bài tập bổ sung dưới dạng các bức thư điện tử quan trọng cần phải trả lời gấp. Khi ngồi xuống đánh máy, tôi không còn thấy kinh khủng nữa. Tôi gõ chậm, nhưng có thể từ từ tiến bộ. Thế là tốt lắm rồi!

Tuy nhiên, tôi nhận thấy mình rất hay nhìn bàn phím. Nhớ vị trí chính xác của các phím trên bàn phím rất hữu ích cho việc gõ những chuỗi ký tự kỳ quặc, ví dụ như mật khẩu, nhưng đó cũng là một bất lợi: Bất cứ khi nào tôi cảm thấy không chắc chắn, tôi lại nhìn xuống. Nếu tôi muốn gõ nhanh bằng 10 ngón, tôi cần phải bỏ thói quen đó càng nhanh càng tốt, nhưng việc đó thật khó thực hiện, vì nhìn xuống bàn phím là việc tốn rất ít công sức.

## Bàn phím Das

Để bỏ thói quen xấu là cứ nhìn chăm chăm vào bàn phím, tôi đã quyết định sử dụng một công cụ mới: bàn phím hoàn toàn trống không.

Bàn phím Das là bàn phím “bá đạo” nhất mà bạn có thể tìm thấy. “Model S bản cuối” không có bất cứ ký hiệu gì trên bàn phím. Nếu bạn không thể gõ mà không nhìn bàn phím thì bạn không thể sử dụng được bàn phím Das. Thật thú vị khi nhìn thấy phản ứng của mọi người khi trông thấy bàn phím này: Ngay cả những người đánh máy lão luyện cũng thấy hốt hoảng.

Tôi cắm bàn phím Das vào, và dùng một tờ giấy để che bàn phím máy tính xách tay của tôi lại, để tôi không thể nhìn thấy bất cứ phím nào. Cảm giác tương tự như khi tôi chuyển sang Colemak lần đầu tiên: Tôi thấy bối rối và hốt hoảng, nhưng chỉ trong một lúc thôi. Những kỹ năng chuyển động mà tôi đã

học được trong bảy tiếng luyện tập đã phát huy tác dụng, và tôi nhận thấy mình có thể gõ khá tốt. Bất cứ khi nào tôi quên một chữ cái nào đó, tôi lại mất vài giây để tìm nó bằng cách thử và sai, nhưng cuối cùng tôi cũng gõ được.

Khi không thể nhìn vào bàn phím, tôi chỉ còn biết tập trung vào màn hình. Bàn phím Das đã hoàn thành trọng trách của nó. Bằng cách đổi bàn phím, tôi đã tự động thay đổi được hành vi của mình.

Cùng với bàn phím Das, tôi đã đổi cả phương pháp tập luyện của mình. Gõ phím bất kỳ đã trở nên lỗi thời, vì thế tôi chuyển sang một chương trình có tên là Type Fu. Ngoài các ký hiệu và các từ ngẫu nhiên, Type Fu còn có một kho những câu thành ngữ và các trích dẫn, giúp cho việc luyện tập trở nên thú vị hơn. Chương trình cũng ghi lại những ký tự mà bạn bỏ lỡ nhiều nhất. Việc này rất có ích. Lúc này, tôi thấy J, U, V và B là khó gõ nhất.

Mỗi tối, tôi lại luyện tập khoảng 45 phút. Sau khoảng 14 tiếng luyện tập cẩn thận, lúc này tôi đã có thể đánh máy với tốc độ 40 từ/phút.

Với tốc độ này, tôi đã có thể sử dụng thư điện tử và lướt web gần như bình thường. Thậm chí tôi còn có thể đánh máy một đề án dài năm trang giấy mà không quá khó khăn. Việc đó khiến tôi mất nhiều thời gian hơn bình thường, nhưng hoàn toàn không phải việc khó chịu nhất mà tôi từng làm.

## Chú tâm luyện tập và luyện tập thêm

Vì tôi có thể làm việc được, nên tôi muốn kiểm tra giả thuyết: Chú tâm luyện tập quan trọng như thế nào?

Hiện tại, tôi đang chuyên tâm luyện tập, vì tôi tập trung vào nhiệm vụ và làm việc tích cực để tiến bộ. Còn những lần đánh máy trong ngày là luyện tập thêm: Bất cứ khi nào tôi viết một bức thư hay một bài luận, tôi lại đánh máy bằng Colemak, dù tôi tập trung tới nội dung của đoạn văn bản hơn là kỹ thuật đánh máy.

Tôi tự hỏi chuyện gì sẽ xảy ra nếu không chuyên tâm luyện tập một thời gian, mà chỉ soạn thảo thư điện tử hoặc lướt web thôi? Tôi đã đi được 2/3 quãng đường để đạt tới mức độ thành thạo mục tiêu (60 từ/phút) chỉ sau 14 tiếng chuyên tâm luyện tập. Liệu luyện tập thêm có thể giúp tôi đi hết đoạn đường còn lại mà không cần phải nỗ lực tập trung không?

Tôi đã quyết định làm một thí nghiệm: Tôi sẽ tạm thời dừng việc luyện tập chuyên tâm trong 30 ngày, và xem chuyện gì sẽ xảy ra. Tôi sẽ tiếp tục đánh máy một cách bình thường với Colemak mà không chuyển về QWERTY. Với khoảng thời gian đó tôi có thể luyện tập thêm để đạt được tốc độ 60 từ/phút, đúng không?

Sau 30 ngày, tôi lại làm bài kiểm tra tốc độ. Bạn có muốn đoán tốc độ đánh máy của tôi không?

40 từ/phút. Không có một chút tiến bộ nào.

Dù tôi cũng hay đánh máy, nhưng tôi lại không chủ động tập trung vào việc cải thiện kỹ năng của mình. Như vậy là luyện tập thêm không đủ để tiến bộ.

Nếu bạn muốn cải thiện một kỹ năng, bạn cần chuyên tâm luyện tập, ít nhất là trong giai đoạn đầu của quá trình học kỹ năng đó. Đó là bài học tôi đã rút ra được.

## Cảm tưởng từ 60 từ/phút

Tôi rất thích đánh máy bằng Colemak. Tôi không còn thấy khó chịu nữa. Thay vào đó, tôi tự hỏi không hiểu làm thế nào tôi có thể xoay sở với QWERTY lâu như vậy được.

Gõ trên QWERTY có cảm giác như tay bạn phải bay khắp bàn phím: Rất nhiều chuyển động tay, theo mọi hướng. Cách bố trí không hề logic một chút nào: Các ký tự bạn thường xuyên sử dụng lại được đặt ở những chỗ khó với tới nhất trên bàn phím.

Ngược lại, Colemak cho bạn cảm giác bạn đang xoay nhẹ các ngón tay để các từ xuất hiện trên màn hình. Bàn tay của bạn rõ ràng là di chuyển ít hơn, và bạn tốn ít thời gian hơn để với tới các ký tự ở hàng trên cùng và dưới cùng. Đó là sự thay đổi tuyệt vời so với QWERTY, và tôi không thể tưởng tượng là có thể quay lại dùng QWERTY như thế nào.

## Tóm lược lại phương pháp

Hãy cùng tóm lược lại những điểm cốt lõi của phương pháp mà tôi đã sử dụng để học cách gõ bằng 10 ngón:

Tôi đã học cách đổi bàn phím của mình sang định dạng Colemak.

Tôi đã tạo ra một vòng phản hồi nhanh bằng cách sắp xếp lại các phím thật trên bàn phím. Nếu tôi quên ký tự đó được xếp ở đâu, tôi có thể tìm thấy dễ dàng.

Tôi đã sử dụng chương trình dạy đánh máy Keyzen để học cách nhớ vị trí các ký tự, học những ký tự được sử dụng nhiều nhất trước. Keyzen đưa các ký tự mà tôi đánh sai vào bộ ký tự luyện tập tiếp theo, vì thế tôi đã dành hầu hết thời gian để luyện tập những ký tự khó cho tới khi độ chính xác của tôi được cải thiện.

Tôi đã luyện tập 45 phút mỗi tối, ngay trước lúc đi ngủ, để não của tôi có thể củng cố những kỹ năng chuyên động vào trí nhớ dài hạn một cách hiệu quả nhất.

Khi đã đạt được tốc độ đủ để làm việc (20 từ/phút), tôi chuyển sang chương trình Type Fu, tập trung vào gõ câu càng nhanh càng tốt, với độ chính xác là 99%.

## Từ đây tôi sẽ đi đâu tiếp?

Giờ tôi đã có thể đánh được 60 từ/phút trên Colemak, tôi không cần phải tiếp tục luyện tập để tăng tốc độ nữa. Với vận tốc này, tôi có thể viết nhanh như tôi muốn. Vì tốc độ đánh máy không còn là yếu tố hạn chế trong công việc của tôi nữa, nên luyện tập để tăng tốc độ đánh máy không còn là ưu tiên lớn nhất của tôi.

Tốc độ đánh máy cũng chính là một kỹ năng. Một số người đánh máy nhanh nhất trên thế giới có thể đạt tốc độ trên 180 từ/phút với bàn phím bình thường, nhưng tiến bộ trong tốc độ kiểm tra không nhất thiết phải đồng nghĩa với tốc độ viết hoặc mã hóa. Những bài kiểm tra tốc độ này thường yêu cầu bạn đánh máy những gì bạn nhìn thấy trên màn hình, vì thế kỹ năng quan trọng mà những người đánh máy nhanh phải luyện tập là phải đọc trước đoạn văn bản, và giữ nó trong trí nhớ ngắn hạn đủ lâu để các ngón tay có thể gõ lại được.

Tôi cũng thích có thể viết một bài văn hoàn thiện với tốc độ trên 180

từ/phút, nhưng như thế có nghĩa là còn hơn cả người đánh máy có tốc độ nhanh nhất. Đánh máy sẽ khó hơn và chậm hơn khi bạn phải tạo ra những thứ mà bạn sẽ đưa vào trang giấy.

Điều khiến tôi thấy ngạc nhiên nhất với việc học Colemak là thật dễ dàng khi tôi đã có 20 năm kinh nghiệm gõ 10 ngón bằng QWERTY. Tôi đã nghĩ rằng cần phải tốn hơn 20 tiếng để thay thế hai thập kỷ của ký ức cơ bắp<sup>(2)</sup>. Tôi đã sai.

Não của chúng ta dễ thay đổi hơn chúng ta nghĩ.

**BÀI HỌC: HÃY KHÁM PHÁ, SAU ĐÓ QUYẾT ĐỊNH**

**Cờ vây sử dụng những nguyên liệu và khái niệm cơ bản như: đường thẳng và đường tròn, gỗ và đá, trắng và đen, kết hợp những điều đó bằng các nguyên tắc đơn giản để tạo ra những chiến lược tinh tế và chiến thuật phức tạp kích thích trí tưởng tượng.**

**–IWAMOTO KAORU, kỳ thủ cờ vây chuyên nghiệp 9 đẳng**

...

Tôi là một thanh gươm trong bóng tối, là một anh lính gác thành.

Ban đêm trời lạnh, tuyết đang rơi. Phía sau tôi, ánh đuốc bập bùng soi rọi đồng bào tôi. Chúng tôi đứng trên chiến lũy của lâu đài, quan sát. Chờ đợi.

Ở đầu kia của cây cầu dẫn vào lâu đài, người chỉ huy của quân thù đang tức giận hô hào chuẩn bị máy bắn đá để tấn công. Binh lính, trộm cướp và cả những người lang thang đều đang chuẩn bị vũ khí giết người. Thầy đồng đang triệu hồn những sinh vật quái dị. Súng cối và máy bắn lửa – những vũ khí được thiết kế để đập tan cánh cổng của chúng tôi – đã được chuẩn bị.

Lũ người này định cướp tài sản quý giá nhất trong vương quốc của tôi: Quả cầu phát sáng đầy sức mạnh. Họ sẽ không lấy được nó. Không phải trong phiên gác của tôi.

Tất cả chúng tôi đều được chuẩn bị kỹ lưỡng. Cả hai bên chỗ tôi đang đứng, binh lính đều đang bố trí máy bắn đá và bắn tên, sẵn sàng trút cơn mưa đá, mưa tên xuống đội quân xâm lược. Giữa những tuyến phòng ngự là máy bắn lửa đã sẵn sàng nhằm thẳng quân thù. Đội quân bảo vệ và các thầy pháp đã làm phép và khiên phản chiếu quanh vành đai, giúp bảo vệ chúng tôi khỏi tên bay và bùa chú, ít nhất cũng được một thời gian.



Chúng tôi chờ. Tuyết rơi và ánh lửa từ ngọn đuốc bập bùng, nhảy múa trên bộ áo giáp sắt sáng bóng của tôi.

Chúng tôi sẽ không gục ngã. Chúng tôi sẽ không chùn bước. Chúng tôi sẽ chiến đấu. Và chúng tôi sẽ chiến thắng.

Ngoài kia, trong bóng đêm, một tảng đá khổng lồ xuất hiện, bay trên đầu kẻ thù. Tảng đá lao thẳng vào cánh cổng tòa lâu đài, và phá hủy hoàn toàn, làm vỡ vụn lớp gỗ gia cố.

Kẻ thù kêu la, sau đó tràn qua cầu.

Ngay lập tức, gươm và khiên của tôi vung lên. Đá bay tới tấp, mang theo sức mạnh và cơn thịnh nộ của tất cả các chiến binh.

“VÌ CÔNG LÝ!”

Hạ vũ khí xuống, tôi không có ý định làm hại bạn

Chỉ huy một đội quân tiến hành một cuộc vây hãm quy mô lớn là một cách tiêu khiển thú vị vào tối thứ Bảy.

Tôi không phải người thích xem tivi, phim ảnh hay thể thao. Thay vào đó, bạn sẽ thấy tôi chơi những trò chơi video như *World of Warcraft* hay *Guild Wars 2* một tiếng, thậm chí hai tiếng đồng hồ.

Tôi thích trò chơi, đặc biệt là những thử thách liên quan tới việc giải lời nguyền, chiến đấu với quái vật và đánh lừa đối phương.

Thế hệ của tôi là thế hệ đầu tiên lớn lên với những trò chơi trên máy tính và video. Bắt đầu với những trò như Atari hay Nintendo, trò chơi càng lúc càng trở nên chi tiết và phức tạp hơn. Hiện nay, thật khó có thể đánh rông trên internet và chiến đấu với những người chơi khác khi họ có đồng minh ở khắp nơi trên thế giới.

Một trong những lý do khiến tôi thấy hứng thú với trò chơi là vì cần phải có một số kỹ năng để có thể chơi giỏi được. Bất kỳ ai cũng có thể tạo ra một nhân vật, nhưng để có thể tham gia chiến đấu và giành được chiến thắng thì bạn cần phải biết bạn đang làm gì.

Có rất nhiều thứ cần phải học. Khả năng nào có thể gây sát thương cao nhất,

hoặc có thể bảo vệ bạn khỏi nguy hiểm? Quái thú của kẻ thù có thể làm gì? Chiến lược nào là tốt nhất khi chiến đấu với những người chơi thực khác?

Càng chơi nhiều, tôi càng chơi tốt hơn. Đó chính là lý do tại sao trò chơi lại thú vị.

Xét về lịch sử, trò chơi video rất mới. Tuy nhiên, những trò chơi về kỹ năng và may rủi lại là một phần trải nghiệm của con người từ hàng nghìn năm nay.

## Trò chơi bàn cờ chiến thuật cổ nhất thế giới

Cách đây một thời gian, tôi tình cờ biết tới một trò chơi thú vị. So với những trò tôi vẫn đang chơi, trò chơi này rất tĩnh lặng, trầm mặc. Tuy nhiên, ẩn sau vẻ ngoài đó là cả một sự kịch tính. Bàn cờ là bản đồ của một cuộc chiến vĩ đại, và người chơi là những vị tướng, đánh nhau để giành quyền tối cao.

Cờ vây là trò chơi cổ nhất trên thế giới, vẫn được chơi theo dạng thức nguyên thủy. Dựa trên những ghi chép lịch sử, cờ vây bắt nguồn từ thời Trung Quốc cổ đại, và đã tồn tại theo nguyên tắc chơi hiện nay ít nhất là 3.000 năm. Thậm chí, có một số nhà sử học còn ước tính con số đó là 4.000 năm. Nếu số năm được coi là một chỉ số đánh giá chất lượng thì cờ vây rất có chất lượng.

Tên tiếng Trung của cờ vây là *weiqi*. *Wei* có nghĩa là “bao vây”, và *qi* có nghĩa là “trò chơi bàn cờ”. Ghép lại, hai chữ này trở thành *weiqi* có nghĩa là “trò chơi bao vây”. Đó là cách miêu tả thú vị và đơn giản về điều kiện để giành được phần thắng trong trò chơi: bao vây đối thủ.

## Chiến thuật

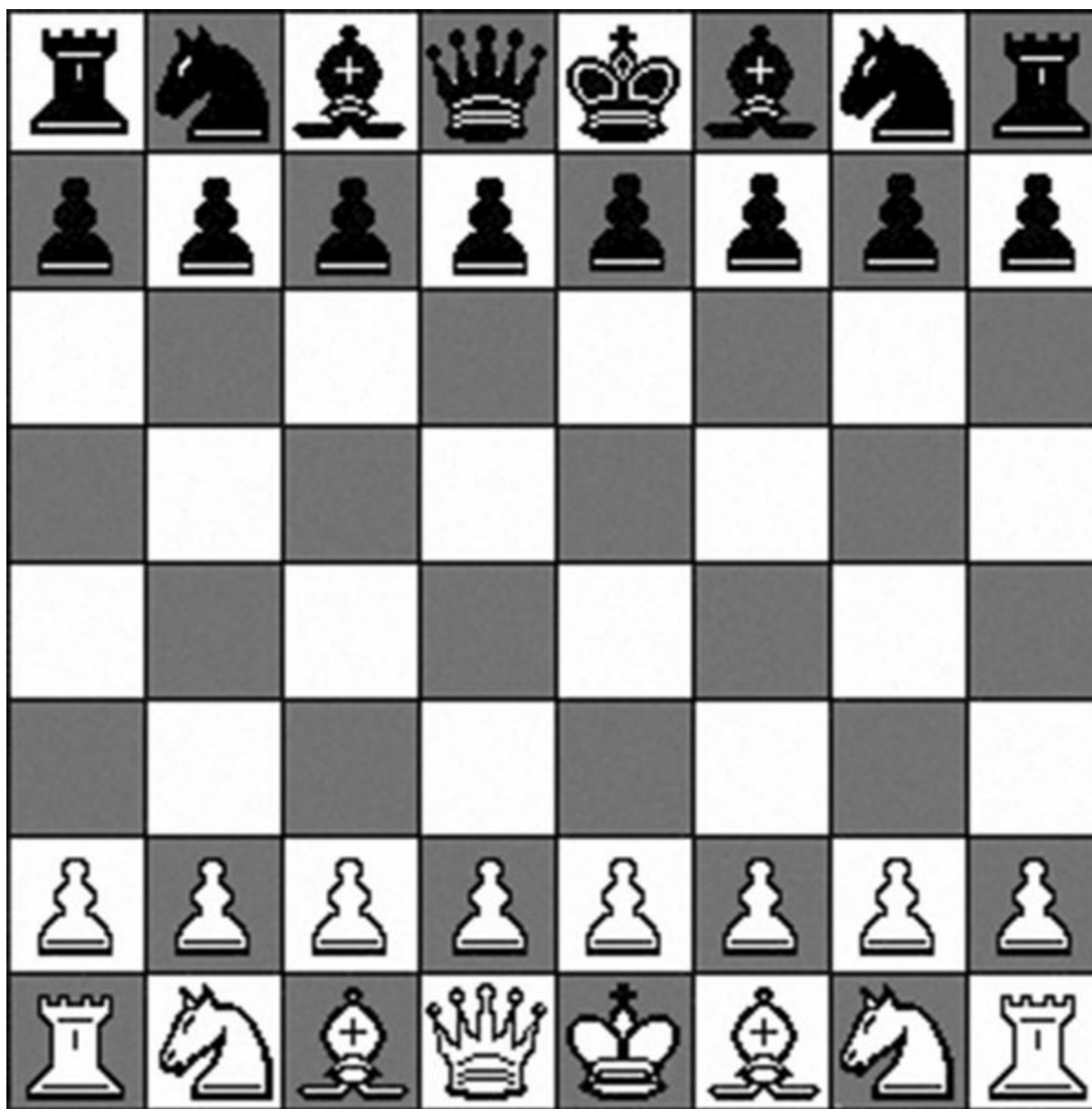
Ở phương Tây, cờ vua là một trò chơi chiến thuật trên bàn cờ phổ biến nhất, do đó, chúng ta có thể sử dụng cờ vua để so sánh. Nhìn qua, hai trò chơi này có nhiều điểm tương đồng.

Mỗi trò chơi đều có hai người chơi, một chơi quân đen và một chơi quân trắng. Trò chơi diễn ra trên một bàn cờ. Quân đen đi trước. Hai người chơi lần lượt di chuyển cho tới khi trò chơi kết thúc. Bạn có thể coi trò chơi như một cuộc đấu quân sự, hai người chơi là hai vị tướng ở hai chiến tuyến.

Đó là điểm giống nhau cuối cùng.

Cờ vua được chơi trên bàn cờ có 64 ô, mỗi cạnh 8 ô. Mỗi quân cờ đứng ở một ô. Bàn cờ mô phỏng một chiến trường và các quân cờ là những người lính.

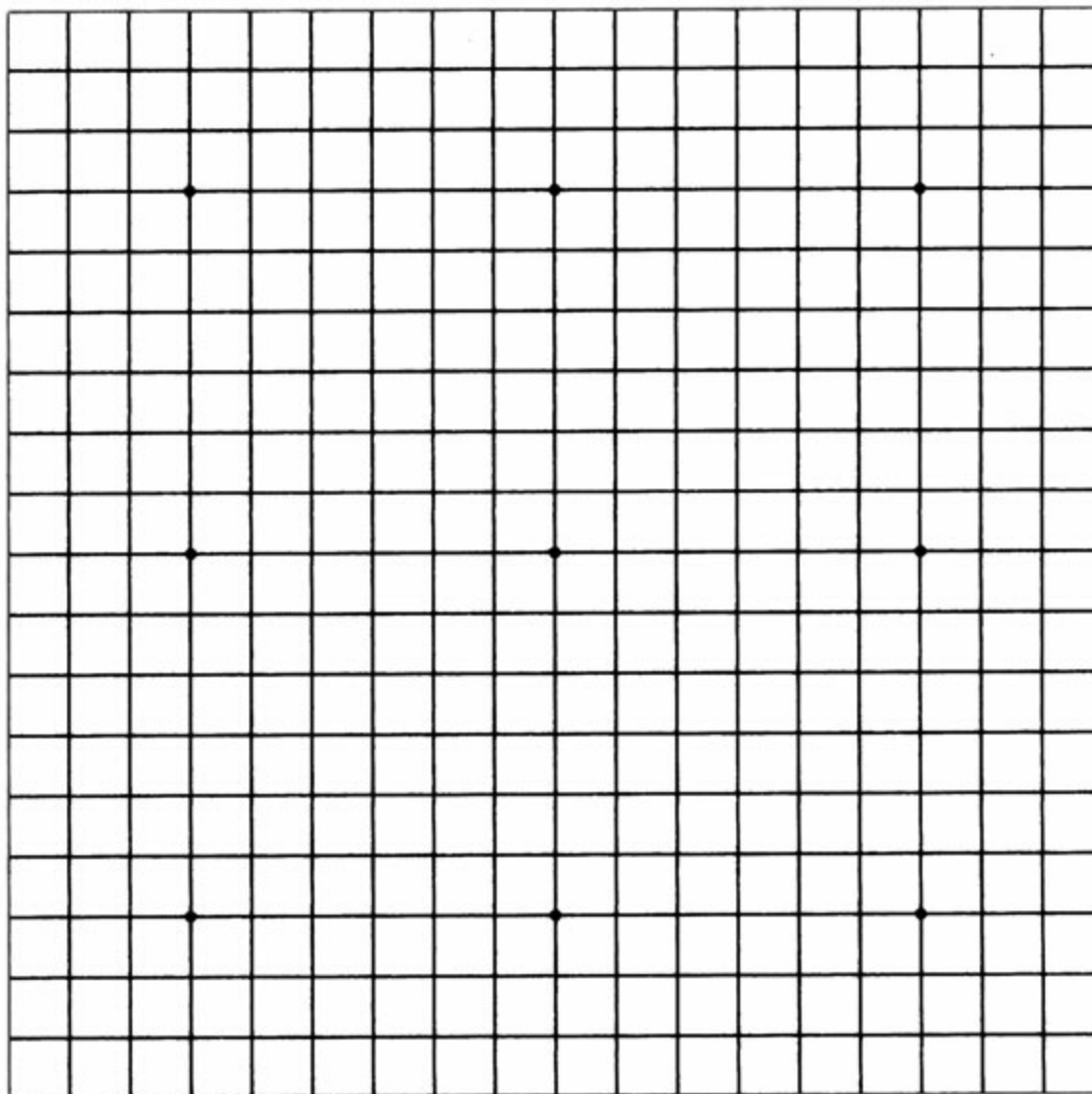
Dưới đây là hình bàn cờ vua khi mới bắt đầu trò chơi:



Ngược lại, cờ vây được chơi trên bàn cờ có 19 ô theo chiều ngang và 19 ô theo chiều dọc. Các quân cờ (các viên đá đen và trắng) được đặt trên giao điểm của các đường thẳng, không phải được đặt trong các ô. Chính vì thế, có 360 giao điểm trên bàn cờ vây, gấp 5.625 lần so với bàn cờ vua.

Dưới đây là hình bàn cờ vây khi mới bắt đầu trò chơi:

Bạn có nhận thấy sự khác biệt không? Bàn cờ vây không chỉ lớn hơn, mà còn không có quân cờ nào trên bàn cờ khi mới bắt đầu. Trong cờ vây, quân cờ thường được *thêm vào* trong quá trình chơi. Còn trong cờ vua, các quân cờ bị *nhắc ra* khi chúng bị ăn.



Hãy chú ý là bàn cờ có hình vuông. Những chấm nhỏ trên bàn cờ (được gọi là “điểm sao”) là những điểm đối xứng. Những điểm này rất quan trọng, và chúng ta sẽ nói về chúng sau.

Trong cờ vua, có sáu loại quân khác nhau, mỗi loại lại có quy tắc và khả năng riêng. Tốt chỉ đi thẳng, trừ khi di chuyển để bắt quân cờ khác. Tượng có thể đi chéo, nhưng không thể đi ngang hoặc dọc. Xe có thể đi ngang hoặc dọc, nhưng không đi chéo. Mã có thể nhảy qua quân cờ khác, nhưng chỉ có thể di chuyển hai ô theo chiều ngang và một ô theo chiều dọc, hoặc hai ô theo chiều dọc và một ô theo chiều ngang. Hậu có thể đi dọc, ngang, chéo, nhưng không thể nhảy qua quân cờ khác. Vua có thể di chuyển theo bất cứ hướng nào, nhưng chỉ có thể di chuyển một ô mỗi lần, trừ khi nó nhảy qua quân xe bằng cách sử dụng một nước cờ đặc biệt gọi là “nhập thành”. Có rất nhiều thứ cần phải nhớ.

So sánh có thể thấy, mỗi thế cờ vây cũng như vậy: quân cờ được đặt vào một giao điểm. Trừ khi quân cờ bị “ăn” sau đó trong lúc chơi, quân cờ còn nằm trên

bàn cờ là quân cờ còn được đi.

Các quân cờ trong cờ vua bị các quân cờ của đối phương “ăn”, giống như một cuộc đấu tay đôi trên chiến trường vậy. Quân xe dùng một cây gậy lớn đập vào đầu quân tượng, và tượng phải rời cuộc chơi.

Quân cờ trong cờ vây bị “ăn” khi bao xung quanh nó là các quân cờ của đối thủ. Khi quân cờ màu Đen bị bao vây mọi phía bởi các quân cờ màu Trắng, thì quân cờ màu Đen phải đầu hàng và bị bắt làm con tin.

Các ván cờ vua chuyên nghiệp thường phải di chuyển từ 30 đến 40 bước. Còn trong cờ vây, 30 nước đầu mới chỉ được coi là mở đầu, ván cờ kết thúc sau khoảng 100 nước đi. Một ván cờ vây thường kết thúc thực sự sau 250 nước đi.

Nói tóm lại, quy mô cờ vây lớn hơn của cờ vua rất nhiều. Nếu mỗi ván cờ là một cuộc chiến, thì bàn cờ chính là bản đồ chiến sự.

Vậy chính xác thì chơi cờ vây như thế nào? Chúng ta hãy cùng đào sâu tìm hiểu.

## Nguyên tắc chơi

Dù bạn tin hay không thì cờ vây cũng chỉ có bảy nguyên tắc lớn, và chúng ta vừa mới đề cập tới hai trong số bảy nguyên tắc đó:

1. Các quân cờ được đặt trên các giao điểm.
  2. Người chơi quân Đen và quân Trắng lần lượt đặt quân cờ trên bàn cờ.
- Năm nguyên tắc xác định diễn biến của trận đấu và điều kiện thắng là:
3. Quân cờ bị “ăn” khi bị quân cờ của địch thủ bao vây mọi phía.
  4. Không được phép chơi một quân cờ vừa mới bị bắt (“tự tử”).
  5. Không được phép lặp đi lặp lại lượt đi trong một vòng vô tận trên bàn cờ.
  6. Trò chơi kết thúc khi người chơi hết quân cờ, một người thua hoặc cả hai người cùng bỏ lượt (pass).
  7. Người chiếm nhiều ô trên bàn cờ nhất vào cuối ván cờ là người thắng.

Trong khi thi đấu, sẽ có thêm một số nguyên tắc được đưa thêm vào để loại bỏ những điều còn mơ hồ (ví dụ như “điều gì tạo thành một vòng lặp?”), xác định phương pháp tính điểm và tránh trường hợp hòa điểm. Còn không, đó là toàn bộ nguyên tắc.

Khá đơn giản, đúng không?

Học nguyên tắc của cờ vây rất đơn giản, chỉ mất vài phút. Bản thân các nguyên tắc không phức tạp chút nào.

Tuy nhiên, có một câu ngạn ngữ cổ nói về cờ vây là “vi kỳ dị học nan tinh” (học chỉ mất một khắc, nhưng để giỏi được phải mất cả đời, hay cờ vây dễ học khó giỏi). Kết hợp các nguyên tắc đơn giản này với bàn cờ đơn giản và các quân cờ đơn giản, bạn lại nhận được một sự phức tạp đến mức muốn nổ tung đầu.

## Quy mô của vũ trụ

**Cờ vua chỉ hạn chế cho nhân loại sống trên trái đất, trong khi cờ vây vượt khỏi thế giới này. Nếu một hành tinh nào có những sinh vật biết lý luận thì ở đó họ phải biết đánh cờ vây.**

—**EDWARD LASKER**, kiện tướng cờ vua và tác giả của cuốn *Goand Go-Moku*

Giả sử chúng ta muốn tạo ra một chương trình máy tính chơi cờ vây rất giỏi, kiểu như Deep Blue, chương trình trí tuệ nhân tạo nổi tiếng đã đánh bại Garry Kasparov, kỳ thủ đoạt chức vô địch thế giới năm 1996.

Thông thường, máy tính đánh lừa người chơi thông qua sức mạnh tính toán siêu việt: Chúng có thể tính toán được tất cả các khả năng di chuyển hợp lý có thể xảy ra trên bàn cờ, sau đó chọn nước cờ có khả năng thành công cao nhất theo tính toán, dựa trên nguồn dữ liệu khổng lồ về những ván cờ trước đó.

Trên bàn cờ vua, kiểu tính toán này không phải là dễ, nhưng cũng không phải là không thể. Có 64 ô, và sự di chuyển của mỗi quân cờ bị ràng buộc bởi những nguyên tắc nhất định. Vì mỗi quân cờ chỉ có thể di chuyển theo những cách nhất định, nên chương trình chỉ cần xem xét một nhóm lựa chọn nhỏ.

Trong cờ vây, người chơi có thể đặt quân cờ vào bất kỳ giao điểm nào trên

bàn cờ. Ván cờ bắt đầu với 360 khả năng lựa chọn, vì thế ngay từ ban đầu, chương trình trí thông minh nhân tạo còn non nớt của chúng ta đã có rất nhiều thứ để phân tích rồi.

Dưới đây là kết quả:

$$360 * 359 * 358 * 357 * 356 = 5.880.282.488.640$$

Vậy là có tới hơn 5 nghìn tỉ khả năng có thể xảy ra, và đó mới chỉ là 5 nước đi đầu tiên.

Tính toán phải nhanh kinh khủng. Bạn có nhớ tôi đã từng nói, thường thì một ván cờ vây sẽ kéo dài trong khoảng 250 nước cờ không? Tùy thuộc vào nhận định của bạn, có khoảng 2,08 lần 10170 cách di chuyển quân cờ hợp lý trên bàn cờ vây có kích thước 19 x 19 ô.

Nếu phép toán đó là đúng thì có nhiều ván cờ vây hơn cả hạt nguyên tử trong vũ trụ.

Theo tính toán về mặt toán học, mỗi một ván cờ vây từng được chơi là một ván cờ vây chưa từng được chơi trước đó trong lịch sử, ngay cả khi bạn sẵn sàng tin vào khả năng có hàng tỉ nền văn minh tiên tiến ngoài vũ trụ cũng biết chơi cờ vây.

Trong thời đại công nghệ hiện nay, ngay cả một chiếc máy tính hiện đại nhất với những thuật toán tốt nhất cũng phải mất khoảng 400 năm để tính toán được một nước đi tối ưu, đây là với giả định máy tính hoàn thành một phép tính chỉ trong vài phần nghìn giây.

Não = nô tung. Trò chơi này *lớn* quá.

Vậy thì kỳ thủ (và máy tính) chơi cờ vây như thế nào?

Nếu người chơi phụ thuộc vào phân tích giải toán để chơi cờ vây, họ sẽ phát điên lên. Nhưng những kỳ thủ có kinh nghiệm lại không hề hóa điên, vì họ có thể xác định được những nước cờ tốt nhất trên bàn cờ chỉ trong vòng vài giây. Làm sao họ làm được điều đó?

Người chơi cờ vây căn cứ vào việc nhận dạng kiểu để xác định các nước cờ có giá trị cao. Người chơi sử dụng rất nhiều từ kích thích trực giác như “hướng” và sáng kiến. Những kỳ thủ cờ vây giỏi nhất có vẻ phụ thuộc nhiều vào hình học,

thâm mỹ và cảm xúc khi tiến hành phân tích và suy luận logic.

Điều đó có nghĩa: Não người không phải được sinh ra để ngẫu nhiên những con số trong giải thuật, nhưng lại rất giỏi trong việc nhận dạng kiểu. Bằng cách nhận dạng các kiểu đi của các quân cờ trên bàn cờ, cũng như kiểu mà các quân cờ thể hiện, những người chơi cờ vây kỳ cựu có thể đọc vị được tình huống hiện tại, sau đó tìm ra nước đi tốt nhất. Tất cả những việc đó không mất đến 400 năm.

Thậm chí, ấn tượng hơn, những người chơi cờ vây giỏi nhất còn có thể đoán trước được các quân cờ sẽ được sắp xếp như thế nào trong tương lai, thường là trong khoảng 30 đến 40 nước đi. Nếu bạn chơi cờ với một người chuyên nghiệp, có thể bạn sẽ cảm thấy anh ta/cô ta đang đọc được tâm trí của bạn vậy.

### Sử dụng cảm giác của bạn...

**Cờ vây đòi hỏi phải có chiến thuật của người lính, có sự chính xác của nhà toán học, có trí tưởng tượng của người họa sĩ, có cảm hứng của nhà thơ, có sự trầm tĩnh của triết gia và có trí thông minh tuyệt đỉnh.**

**—ZHANG YUNQI, trích từ một tài liệu nội bộ của Học viện cờ vây Trung Quốc, 1991**

Một trong những bài báo đầu tiên tôi đọc về cờ vây đã đưa ra một lập luận khá thú vị: chơi cờ thực ra không phải là tranh đấu hoặc chế ngự đối thủ. Chơi cờ là làm chủ *bản thân*.

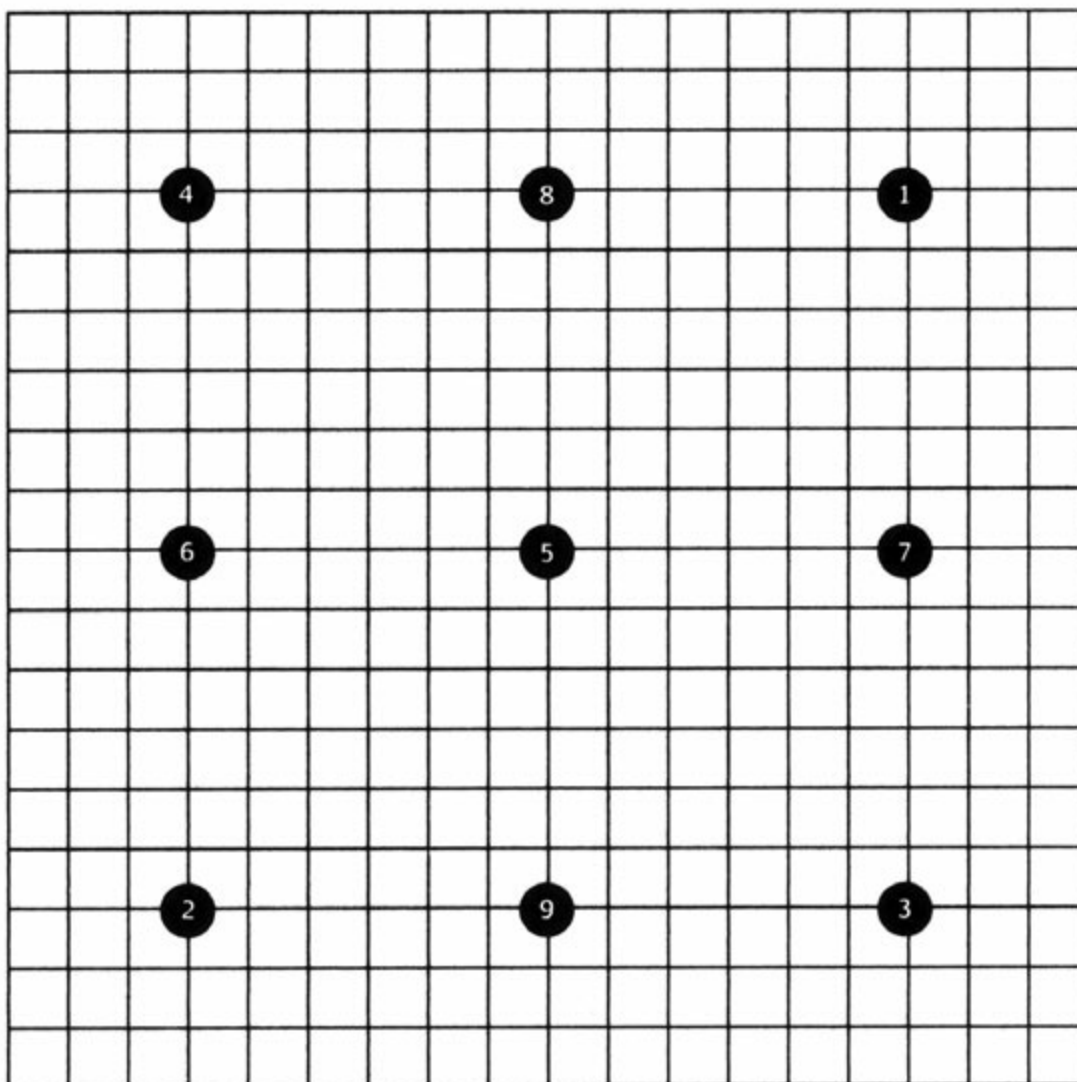
Trò chơi may rủi đã có từ những buổi đầu của nền văn minh nhân loại. Khi bạn đổ xúc xắc, thần may mắn sẽ quyết định ai là người thắng. Kỹ năng không phải là yếu tố quyết định.

Ngược lại, những trò chơi kỹ năng thường tập trung vào việc đánh lừa hoặc chơi tốt hơn người kia. Ai bộc lộ năng khiếu và kiểm soát tốt nhất? Ai có thể tìm thấy những cơ hội lớn trước tiên? Ai giỏi phát hiện điểm yếu của đối phương hơn? Cờ vua thuộc nhóm này: Giỏi đồng nghĩa với việc phải thành thạo các thủ thuật và đọc vị được đối thủ.

Xét về khía cạnh này, cờ vây là một trò chơi độc nhất vô nhị: Chấp quân là một cách được lập sẵn với chủ ý làm yếu những kỳ thủ mạnh. Nếu đối thủ của



bạn bắt đầu ván cờ với một vài quân trên bàn cờ ở những vị trí chiến lược chính, thì điều đó có ảnh hưởng nghiêm trọng tới cách bạn chơi. Dưới đây là vị trí có thể được đặt của những quân cờ chấp:



Trong một ván cờ vây được chấp quân theo đúng chuẩn giữa hai người chơi ngang sức ngang tài, mỗi người chơi có thể giành chiến thắng sau khoảng nửa ván đấu. Nếu một người lấn át người kia, thì trong ván cờ tiếp theo, người thua sẽ bắt đầu với một hoặc hai quân cờ bổ sung trên bàn cờ.

Chính vì đặc điểm chấp quân này mà bạn có thể nghĩ cờ vây là trò chơi bạn đấu lại chính mình. Chính xác, bạn đưa ra lựa chọn dựa trên những bước đi của đối thủ, nhưng thắng trận đấu không phải là mục tiêu duy nhất.

Khi kỹ năng của bạn phát triển hơn, số lượng quân cờ chấp mà bạn cần sẽ giảm đi. Khi bạn chơi với những đối thủ mạnh, bạn có thể đấu ngang ngửa với họ. Và cuối cùng, bạn thậm chí còn có thể chấp quân cho những đối thủ yếu hơn.

Lịch sử cờ vây đã chứng minh vai trò làm chủ bản thân đối với trò chơi này: Muốn chơi giỏi phải làm chủ được suy nghĩ và cảm xúc của bản thân. Những người giỏi có thể học cách nhìn các dấu hiệu của quân cờ để biết được trạng thái tinh thần và tình cảm của đối thủ.

Theo truyện được kể lại, những cao cờ thời xưa có thể đọc kỳ phổ và nhận ra khi nào người chơi tức giận, bối rối, ghen tị hay tham lam, cũng như xác định được chính xác thời điểm “người hầu bước vào để rót trà”.

## Quân cờ Ninja

Qua nhiều thế kỷ, hệ thống chấp quân của cờ vây đã phát triển thành hệ thống xếp loại: Sự khác biệt về thứ hạng được quyết định bởi số lượng quân cờ mà người chơi được đánh giá là kém hơn có thể đặt trên bàn cờ khi bắt đầu ván cờ.

Tin hay không tùy bạn, nhưng hệ thống đánh giá bằng “đai” trong võ thuật bắt nguồn từ cách xếp hạng của cờ vây. Khi kỹ năng của kỳ thủ phát triển hơn, anh ta sẽ tăng được thứ hạng của mình.

Những người mới bắt đầu thường được xếp ở cấp 35 (tương tự như đai trắng). Khi người chơi mạnh hơn, thứ hạng của anh ta sẽ giảm dần cho tới khi đạt cấp 1 (cấp 1 là cấp cao nhất).

Cấp cao hơn cấp 1 là 1 đẳng, tương đương với đai đen. Từ đây, thứ hạng tăng dần cho tới khi kỳ thủ đạt 9 đẳng, thứ hạng chính thức cao nhất, còn 10 đẳng chỉ là danh hiệu vinh danh người giành chiến thắng trong giải đấu quốc tế.

Có nhiều cách để xác định được thứ hạng của kỳ thủ cờ vây. Cách đầu tiên là thông qua thi đấu: Nếu một người có thể thắng hầu hết các trận đấu với những kỳ thủ cấp 12, nhưng lại thua hầu hết các trận đấu với các kỳ thủ cấp 8 thì người đó có lẽ chỉ đạt cấp 10. Kỳ thủ càng tham gia nhiều trận đấu xếp hạng thì thứ hạng của kỳ thủ đó càng chính xác.

Còn một cách khác để đánh giá, xếp hạng kỳ thủ, đó là hoàn thiện thế cờ vây: Sắp xếp bàn cờ thể hiện một tình huống nào đó, và yêu cầu kỳ thủ quyết định nước cờ tiếp theo để đạt được một kết quả cụ thể ví dụ như “bắt nhóm quân đen” hay “cứu quân trắng”. Khi kỳ thủ tăng thứ hạng, khả năng giải quyết vấn đề một cách chính xác của kỳ thủ đó cũng được nâng lên.

Vì cờ vây đã có từ rất lâu, nên có cả một kho tàng đồ sộ về các thế cờ vây để

ngiên cứu. Những thế cờ này được sắp xếp theo độ khó: Kỳ thủ cấp 20 sẽ phải vật lộn với những thế cờ cấp 10, trong khi kỳ thủ 1 đẳng lại thấy những thế cờ cấp 10 thật đơn giản và dễ dàng. Chính vì có những thế cờ xếp hạng này mà cờ vây trở thành trò chơi bạn có thể học trong lúc chơi. Hoàn thiện các thế cờ trong những cuốn sách này là cách hay để tự mình luyện tập kỹ năng chơi cờ vây, cũng như ước lượng được khả năng của bạn.

Tôi đã sẵn sàng để bắt đầu. Tôi cần phải luyện tập gì đây?

## Khởi động

Thật khó để chơi cờ vây nếu không có bàn cờ và quân cờ, vì thế tôi đã mua một bộ cờ vây của Yellow Mountain Imports, công ty phân phối sản phẩm cờ vây nhập từ các nước trên thế giới tại Mỹ.

Bàn cờ và quân cờ vây khác nhau rất nhiều về giá cả và chất lượng. Bạn có thể mua được một bàn cờ với các quân cờ bằng nhựa đơn giản với giá vài đô la, vì thế có thể dễ dàng chuẩn bị được một bộ “trang thiết bị cơ bản” để bắt tay vào thực hành.

Ngược lại, những bàn cờ chất lượng cao cấp được làm bằng gỗ gụ, loại vật liệu truyền thống vẫn được sử dụng ở Nhật Bản, được bán với giá hàng chục nghìn đô la. Cũng như vậy, bạn có thể mua những quân cờ được làm bằng đá và vỏ trai thật, nhưng sẽ phải tốn kha khá đấy.

Sau khi nghiên cứu, tôi đã chọn bàn cờ bằng gỗ vân sam (thường có màu trắng) và bộ quân cờ Yunzi, do công ty cùng tên ở Trung Quốc sản xuất. Công ty xem thành phần của các quân cờ như một bí mật kinh doanh: Quân cờ có cảm giác cứng và mịn, tạo ra âm thanh dễ nghe khi bạn đặt chúng lên bàn cờ, và giá cả lại phải chăng. Đi kèm với các quân cờ là những chiếc bát gỗ truyền thống. Những chiếc bát này được đặt bên cạnh bàn cờ trong suốt quá trình chơi.

Ngoài việc mua bàn cờ và quân cờ, tôi còn chọn thêm một số cuốn sách về cờ vây dành cho người mới bắt đầu, trong đó có:

, *Cờ vây: Giới thiệu về trò chơi*, tác giả Cho Chikun (2010)

, *Cuốn sách thứ hai về cờ vây: Điều bạn cần biết sau khi đã học được các*

*nguyên tắc chơi*, tác giả Richard Bozulich (1998)

*Làm cách nào để không chơi cờ vây*, tác giả Yuan Zhou (2009)

*Những bài học cơ bản về cờ vây*, tác giả Toshiro Kageyama (1996)

*Nguyên lý khai cuộc trong cờ vây*, tác giả Otake Hideo (1992)

Tôi tìm thấy những cuốn sách này qua một trang web có tên là Sensei's Library – trang web có hàng trăm trang thông tin và bình luận về lịch sử cùng thủ thuật chơi cờ vây. Có trang là một danh sách rất dài những ý kiến về những cuốn sách hay nhất liên quan tới cờ vây, do những cao cờ tổng hợp. Đây có thể xem là tài sản quý giá trong giai đoạn đầu của quá trình học hỏi này. Tôi đọc hết cả trang liệt kê gần 100 cuốn sách đó.

Và đây là cách tôi đã chọn xem nên đọc cuốn sách nào trước: *Cuốn Cờ vây: Giới thiệu về trò chơi* có lẽ nằm trong tập những cuốn đầu tiên nên đọc dành cho người mới bắt đầu, thế nên không cần phải lặn tẩn. Tương tự như vậy, *Cuốn sách thứ hai về cờ vây* cũng là cuốn sách rất đáng đọc vì đó là tài liệu mở đầu về chiến lược. Với giả định là bạn đã biết các nguyên tắc cơ bản, cuốn sách tập trung nhiều hơn vào những thủ thuật cơ bản.

*Cuốn Làm cách nào để không chơi cờ vây* là một ví dụ về điều ngược lại, và tôi rất thích đọc cuốn sách này. Bạn có thể học được rất nhiều về việc làm tốt một điều gì đó bằng cách học từ các lỗi sai. Với hầu hết các kỹ năng, bạn cần phải tự mình nghiên cứu điều ngược lại, nhưng trong trường hợp này, có cả một cuốn sách được viết bởi một chuyên gia cho bạn. Thật tuyệt, đúng không?

*Cuốn Những bài học cơ bản về cờ vây*, thường được gọi là “*cuốn sách vàng*”, là cuốn sách mà hầu hết những kỳ thủ cao tay đều giới thiệu như cuốn khoa giáo giúp họ tiến bộ nhanh nhất. Có vẻ như đó là cuốn sách nâng cao, không phù hợp với người mới bắt đầu, nên tôi sẽ để nó lại và đọc sau khi tôi đã học được những điều cơ bản.

Cuối cùng, tôi chọn cuốn *Nguyên lý khai cuộc trong cờ vây*, vì dù là từ phân tích của tôi thì rõ ràng khai cuộc là một điều cực kỳ quan trọng. Vì bàn cờ vây hoàn toàn trống trơn khi bắt đầu ván cờ, ngoại trừ những quân cờ được chấp, vì thế 30 – 40 nước cờ đầu tiên sẽ hình thành một cấu trúc có ảnh hưởng quan trọng tới toàn bộ ván cờ còn lại. Nếu bạn không biết cách khai cuộc chính

xác, trong khi đối thủ của bạn lại biết, thì bạn chắc chắn sẽ thua. Chính vì vậy, khai cuộc là một chủ đề rất đáng để học ngay từ đầu.

## Theo chân Thỏ trắng

Công nghệ hiện đại giúp cho việc luyện tập cờ vây trở nên đơn giản hơn một chút. SmartGo (Cờ vây thông minh), một chương trình có sẵn trên iPhone và iPad, bao gồm một chương trình AI được tích hợp sẵn, một cơ sở dữ liệu về các thế cờ vây dùng để đánh giá thứ hạng, và các ván cờ đi vào lịch sử để bạn nghiên cứu.

SmartGo được thiết kế để tận dụng lợi thế màn hình cảm ứng của thiết bị, vì thế bạn có thể đặt quân cờ lên “bàn cờ” bằng cách chạm vào giao điểm thích hợp. Điều này giúp cho việc hoàn thiện thế cờ vây trở nên đơn giản hơn nhiều. Thay vì tưởng tượng về giải pháp, chương trình sẽ cho bạn phản hồi ngay lập tức. Vòng phản hồi nhanh này giúp cho việc luyện tập các thế cờ trở nên đơn giản hơn rất nhiều, đặc biệt là các thế cờ cần phải di chuyển nhiều hơn một lần để giải được.

Tôi đã sẵn sàng. Tôi đã có mọi thứ tôi cần để học cách chơi. Chỉ còn một việc nữa mà tôi cần làm... đó là đảm bảo rằng tôi sẽ dành thời gian để chơi cờ vây chứ không phải chơi một trò gì khác.

## Loại bỏ điều gây xao lãng

Cờ vây không hấp dẫn, cuốn hút như trò chơi hành động cảm giác mạnh. Học chơi cờ vây cần phải có thời gian và sự tập trung. Tôi sẵn sàng chơi các trò chơi khác, nhưng tôi lại có rất ít thời gian rảnh rỗi. Nếu tôi muốn tiến bộ trong trò cờ vây nhanh nhất có thể, tôi cần phải tập trung.

Điều đó có nghĩa là tôi cần phải loại bỏ những điều có khả năng khiến tôi xao lãng. Mối đe dọa lớn nhất chính là những trò chơi khác: Thời gian dành cho việc đánh quái thú không phải là thời gian dành cho việc học cờ vây. Nếu tôi còn tiếp tục chơi trò chơi video, tôi sẽ không còn chút thời gian nào để học.

Hãy nhớ, thời gian là thứ không bao giờ có thể tìm thấy được, mà phải tạo ra.

Chính vì vậy, tôi đã quyết tâm sẽ không chơi bất kỳ trò chơi nào khác cho tới

khi tôi đã dành ra ít nhất 20 tiếng cho cờ vây. Tôi không được phép xao lãng nếu muốn học một cách nhanh chóng.

Tạm biệt nhé, *World of Warcraft*... thật tuyệt khi biết bạn.

Trước khi bước vào cuộc phiêu lưu mang tên cờ vây, tôi đã phải hủy tài khoản *World of Warcraft* và xóa trò chơi này trong máy tính của tôi. Nếu trò chơi không được cài đặt thì dù có muốn tôi cũng không thể chơi được. Tôi sẽ không chơi trò chơi video trong bất cứ hoàn cảnh nào cho tới khi thí nghiệm với cờ vây của tôi chính thức hoàn thành.

## Nguyên tắc của trò chơi, kiểm tra lại

Cách tốt nhất để bắt đầu là kiểm tra lại các nguyên tắc. Tôi đã đọc tất cả các nguyên tắc đó, nhưng tôi cần phải xem chúng như thế nào trên bàn cờ.

Có một đặc điểm thú vị của cờ vây khiến cho việc học trở nên dễ dàng hơn, đó là: Bạn có thể thay đổi kích thước bàn cờ. Vì bàn cờ đối xứng, nên bạn có thể mở rộng hoặc thu hẹp nó, miễn sao bạn chọn số dòng lẻ. Nguyên tắc và thủ thuật thì vẫn vậy.

Bàn cờ thi đấu có kích thước 19 x 19, nhưng để học, tốt nhất là bắt đầu với bàn cờ 7 x 7. Cỡ đó là đủ không gian để có thể học những ý tưởng chính, và cũng nhỏ vừa đủ để không bị loạn.

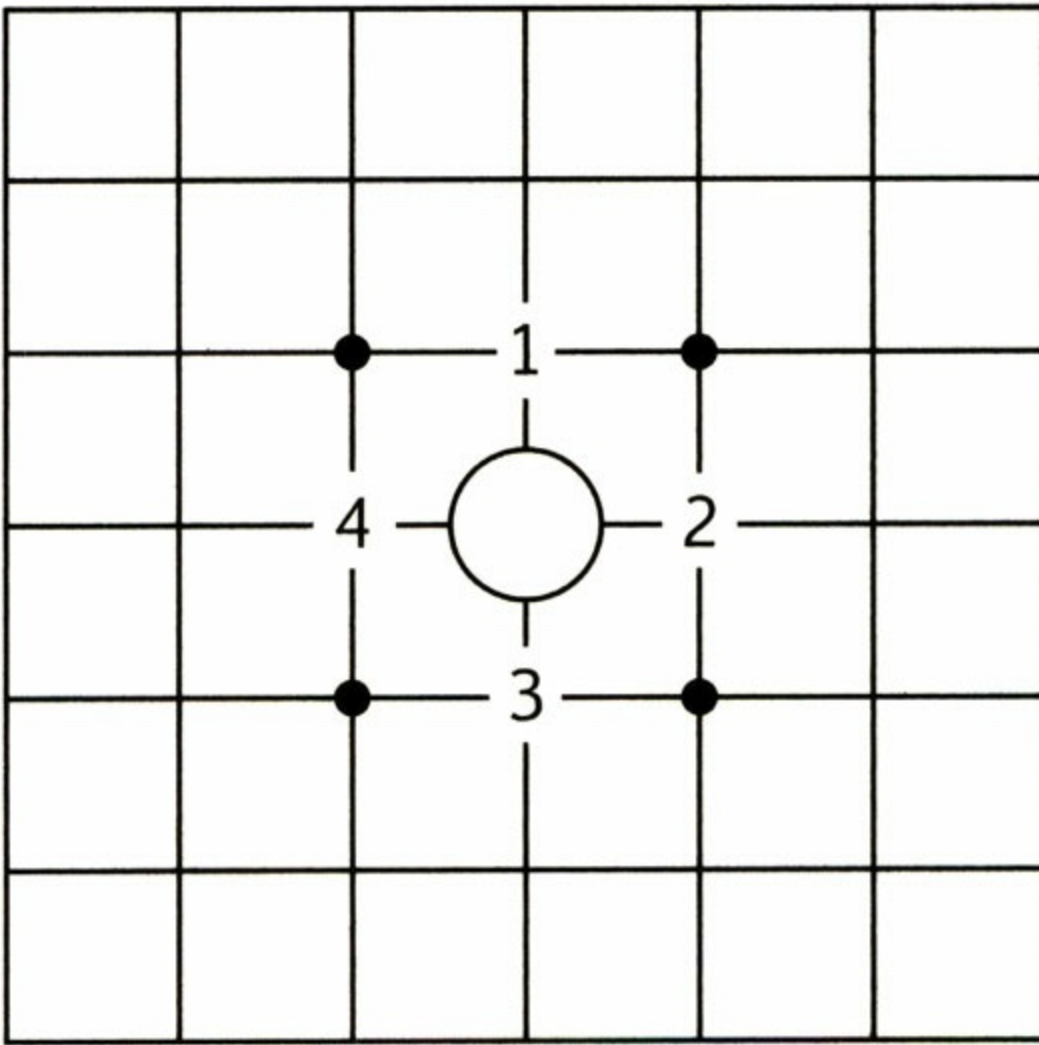
Giờ hãy cùng xem các nguyên tắc sẽ như thế nào trên một bàn cờ thực sự.

## Cho tôi khí hay cho tôi chết

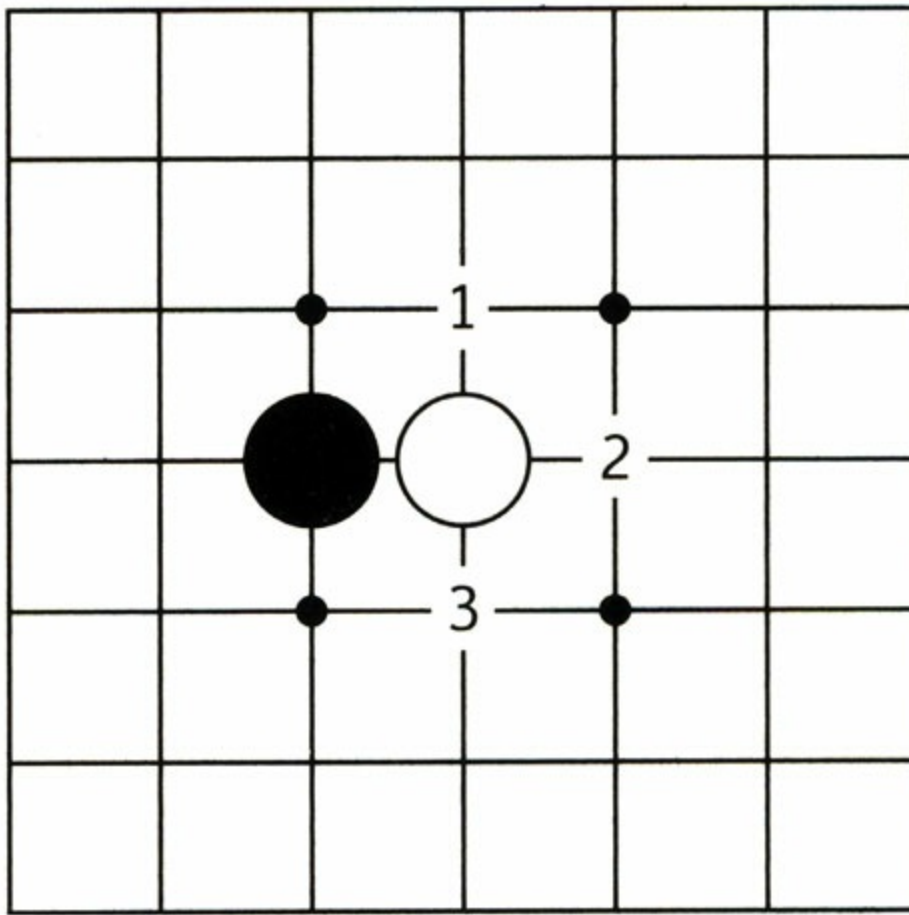
Hãy bắt đầu với nguyên tắc thứ ba của cờ vây: Quân cờ sẽ “bị bắt” khi chúng bị các quân cờ của đối thủ bao vây xung quanh.

Khi bạn đặt một quân cờ vào một giao điểm, những giao điểm liền kề với quân cờ đó được gọi là *khí*. Hãy nghĩ đến người lính trên chiến trường: Nếu có một nơi an toàn để rút lui, người lính sẽ không bị bắt.

Một quân cờ có thể có tới đa bốn khí:



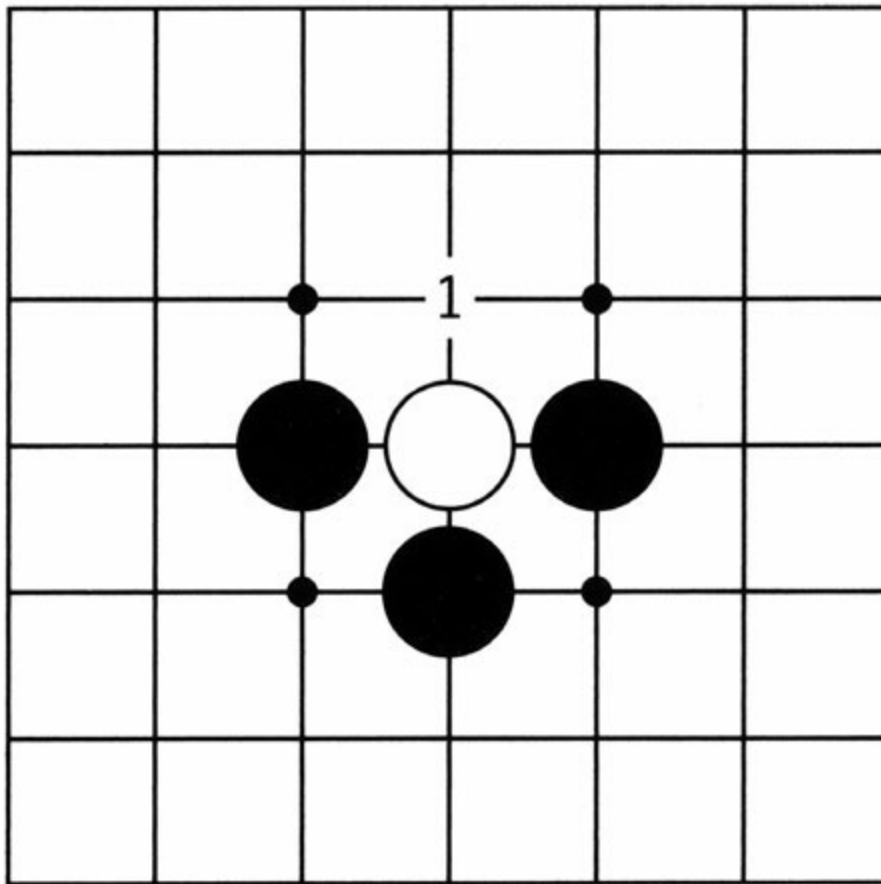
Khí sẽ giảm khi đối thủ tấn công bằng cách đặt một quân cờ vào ngay sát quân cờ của bạn. Cuộc tấn công này giảm số lượng khí của quân cờ trắng xuống còn ba:



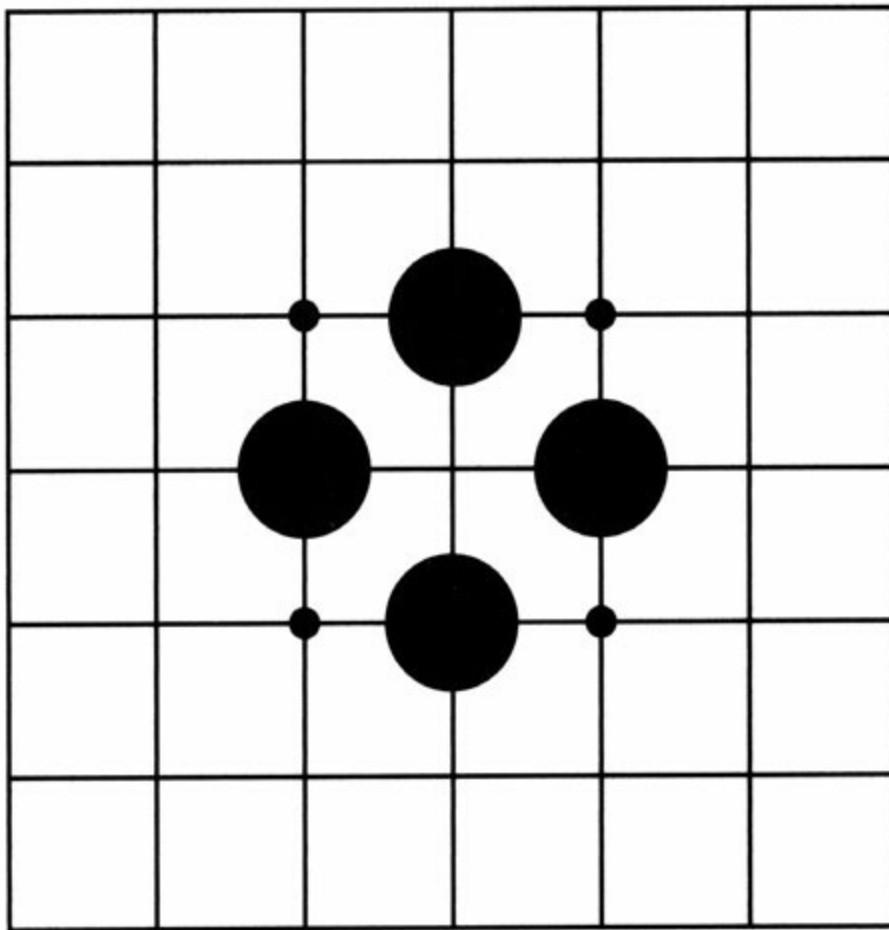
Cuộc tấn công khác lại giảm lượng khí của quân cờ trắng xuống còn hai:







Tình huống này được gọi là *atari (chết khí)*. Nếu quân đen loại bỏ nốt khí còn lại của quân trắng, quân trắng sẽ bị bắt làm tù binh, và bị nhấc ra khỏi bàn cờ:

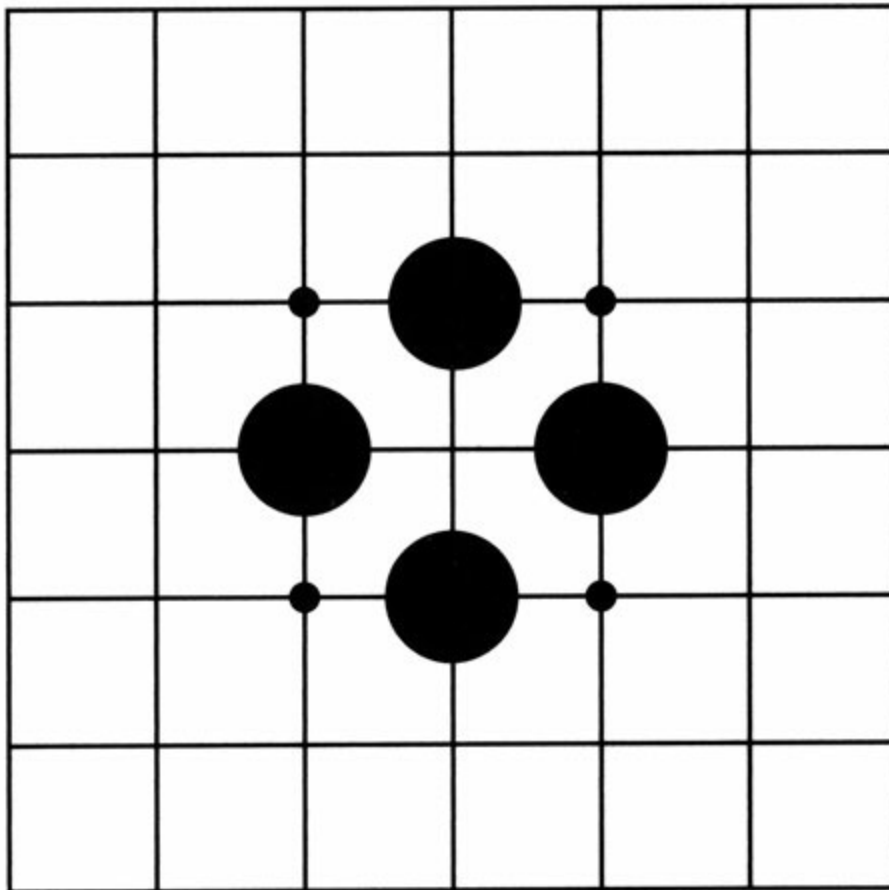


Tính đến lúc này thì vẫn còn khá đơn giản. Nếu bạn không muốn quân cờ của bạn bị bắt, bạn cần phải đảm bảo chúng có càng nhiều khí càng tốt. Nếu bạn muốn bắt quân cờ của đối thủ, hãy loại khí của chúng.

### Làm ơn đừng tự tử

Đây là cơ hội tốt để tìm hiểu nguyên tắc thứ tư: Không được phép chơi quân cờ vừa mới bắt quân (“tự tử”).

Hãy cùng xem lại tình huống cuối cùng:

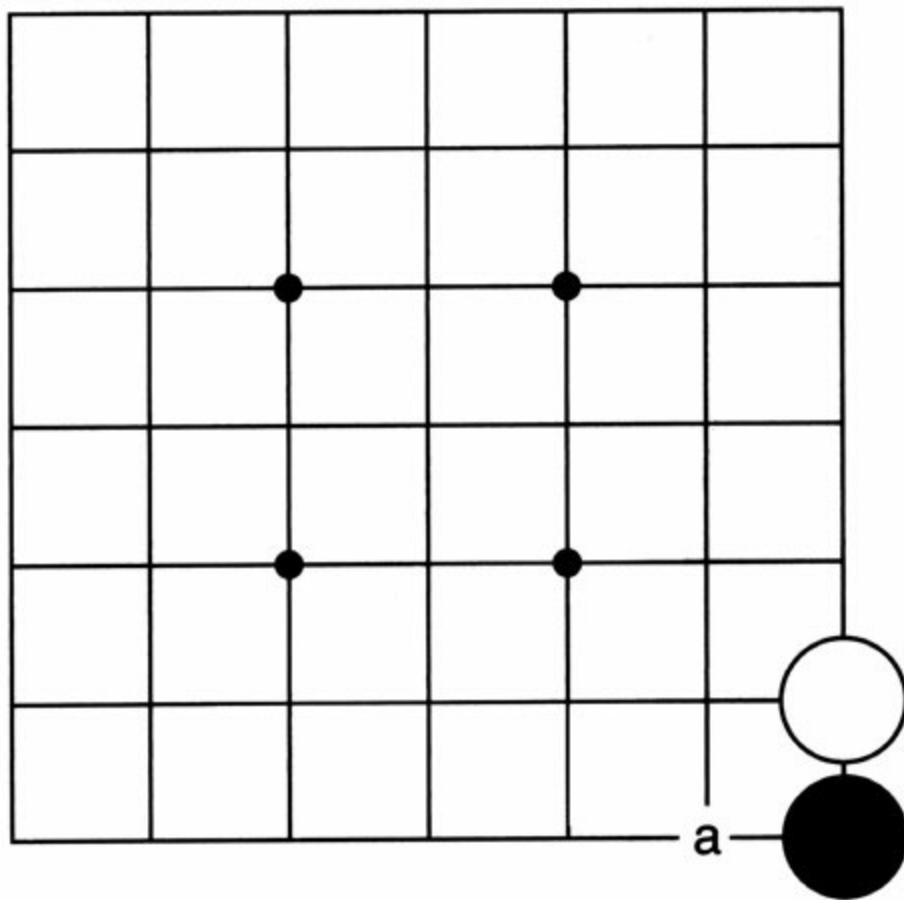


Hình này được gọi là “hình cờ hoa”, và nó rất phổ biến trong cờ vây. Khoảng trống giữa các quân đen không còn khí, vì thế nếu bên trắng đặt một quân (trắng) vào đó, nó chắc chắn sẽ bị bắt ngay lập tức. Nguyên tắc thứ tư cấm nước cờ “tự tử” kiểu này. Nguyên tắc này sẽ trở nên vô cùng quan trọng khi chúng ta bắt đầu nhìn vào một nhóm các quân cờ lớn hơn, vì thế, chúng ta sẽ quay trở lại với nó sau.

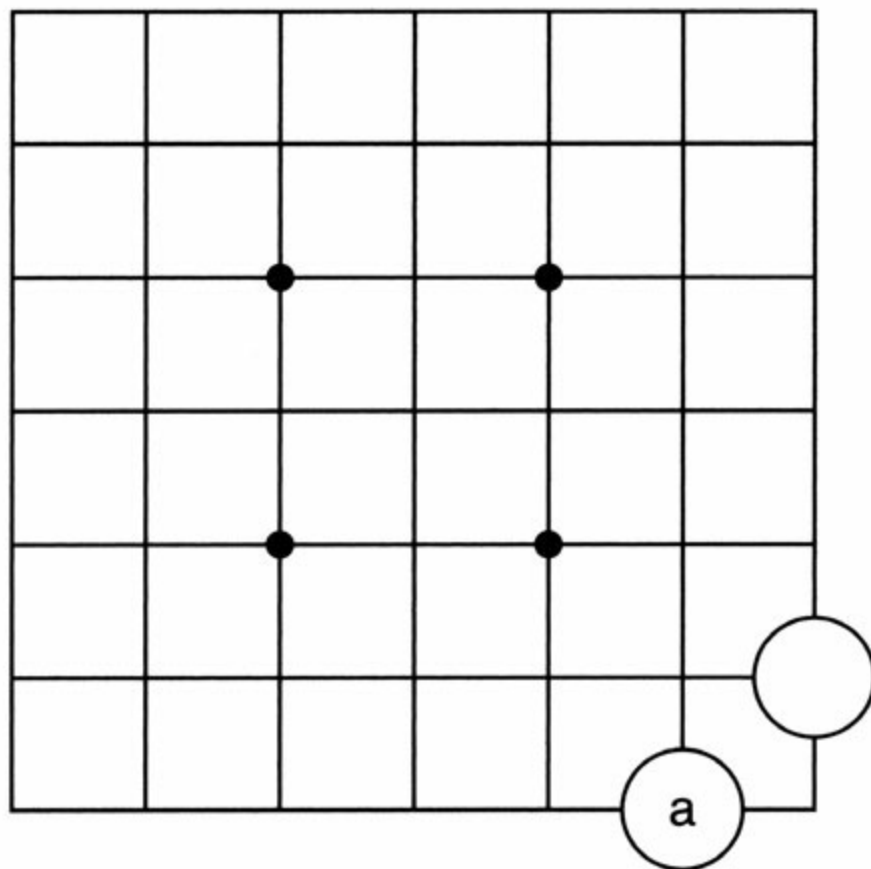
### Trở lại góc

Ở giữa bàn cờ vây, các quân cờ thường có nhiều khí cho tới khi chúng bị tấn công. Ở các góc và các cạnh bên của bàn cờ, có ít khí hơn, vì không có nhiều giao điểm ở vùng lân cận.

Trong hình dưới, quân đen chỉ còn lại một khí duy nhất, chính là giao điểm được đánh dấu là “a”:



Quân đen đang ở trong tình huống atari. Nếu bên trắng đặt một quân vào vị trí “a”, quân đen sẽ bị bắt làm tù binh.



Ví dụ này minh họa sự khác biệt giữa các góc và ở giữa bàn cờ. Ở các góc, chỉ cần hai quân để bắt được quân của đối phương. Ở giữa thì cần có ít nhất bốn quân. Còn ở các cạnh, cần ba quân để bắt được quân của đối thủ.

Kết quả là, thường thì thiết lập và bảo vệ vùng lãnh thổ ở các góc của bàn cờ là dễ dàng nhất. Ở các cạnh bên khó hơn một chút, và ở giữa là khó hơn cả, vì cần quá nhiều quân để bẫy được đối phương.

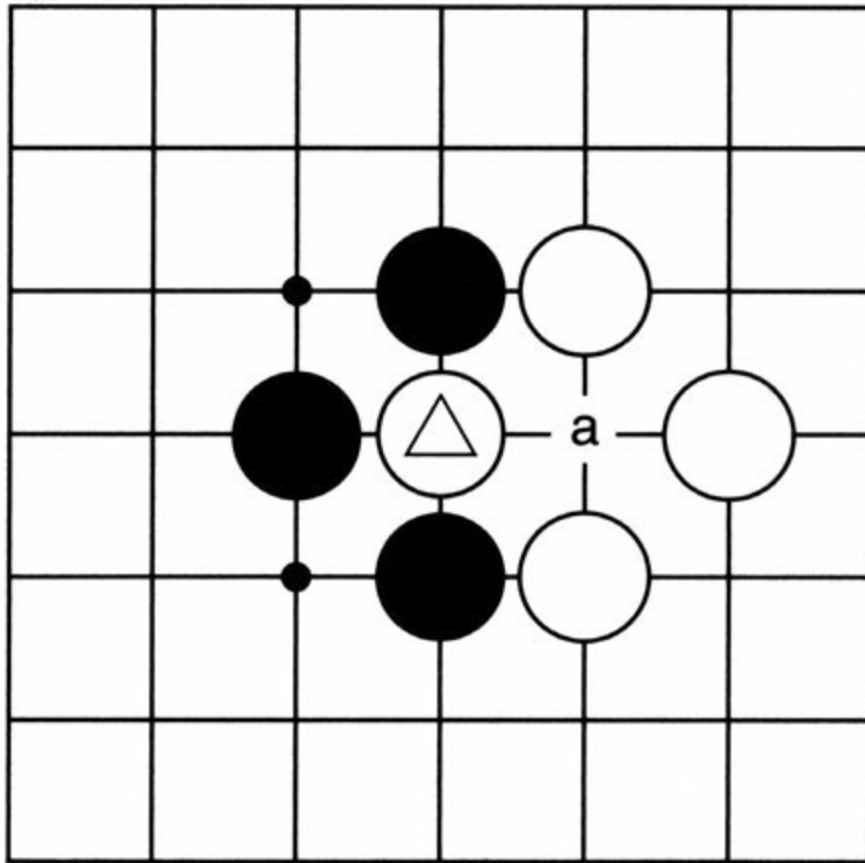
Đó chính là lý do tại sao nếu bạn xem phần mở đầu của một ván cờ vây, bạn sẽ thấy những kỳ thủ giỏi luôn tuân theo một kiểu chung. Trước tiên, họ sẽ hình thành các vị trí ở góc của bàn cờ. Một khi vùng lãnh thổ ban đầu đó được an toàn, họ sẽ mở rộng sang các bên. Giữa bàn cờ là vùng lãnh thổ cuối cùng, và được để dành cho phần cuối ván.

## Tới vô cực, nhưng không xa hơn

Như vậy là chúng ta đã vừa được học nguyên tắc về bắt quân, giờ hãy cùng kiểm tra nguyên tắc thứ 5: Không được lặp đi lặp lại một lượt đi trong một vòng vô tận.

Tình huống này được gọi là *ko* – từ tiếng Nhật có nghĩa là “vô tận”. Trong cờ vây, vòng vô tận là điều có thể xảy ra, và nếu được chấp thuận, nó có thể làm hỏng hầu hết các ván cờ.

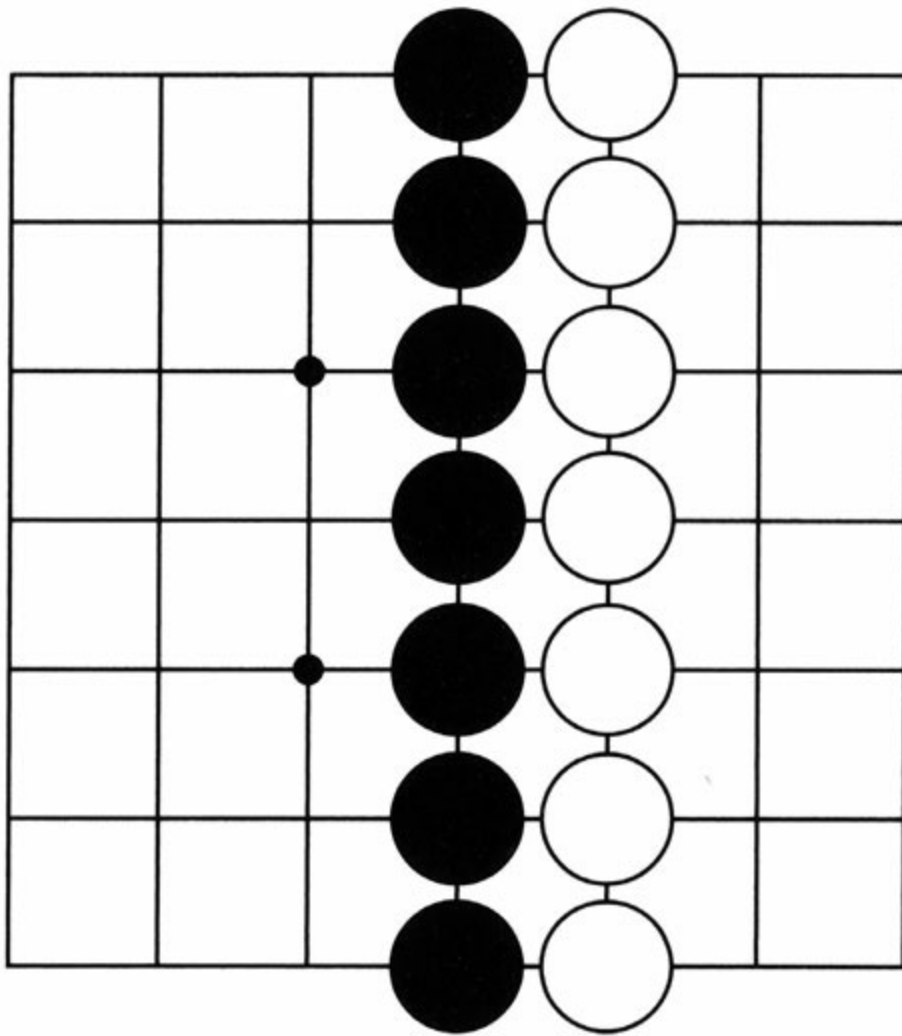
Dưới đây là hình ảnh của vòng vô tận. Quân đen có thể bắt quân trắng (được đánh dấu bằng một hình tam giác), nếu bên đen đặt một quân cờ vào vị trí “a”:



Bạn có để ý chuyện gì đã diễn ra không? Chúng ta có cùng một kiểu, chỉ là đảo ngược lại mà thôi. Giờ thì nếu bên trắng đặt một quân vào vị trí a, thì bên trắng có thể bắt lại được một quân của bên đen:



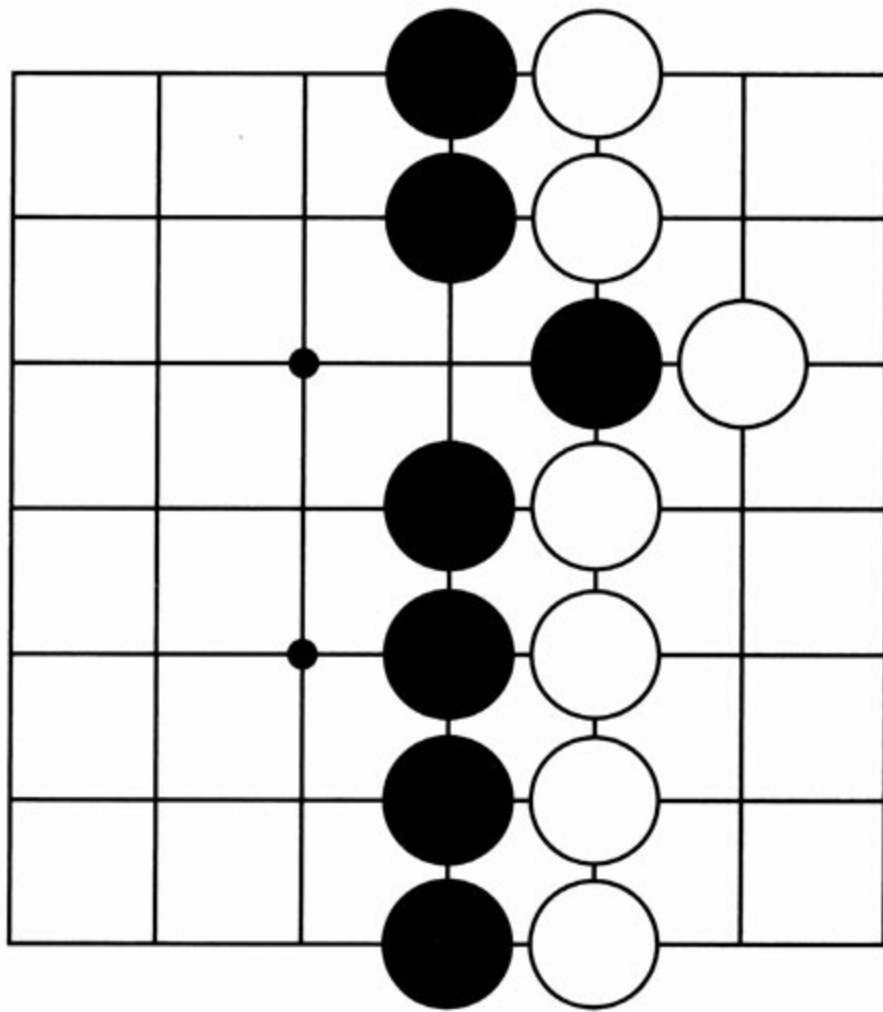




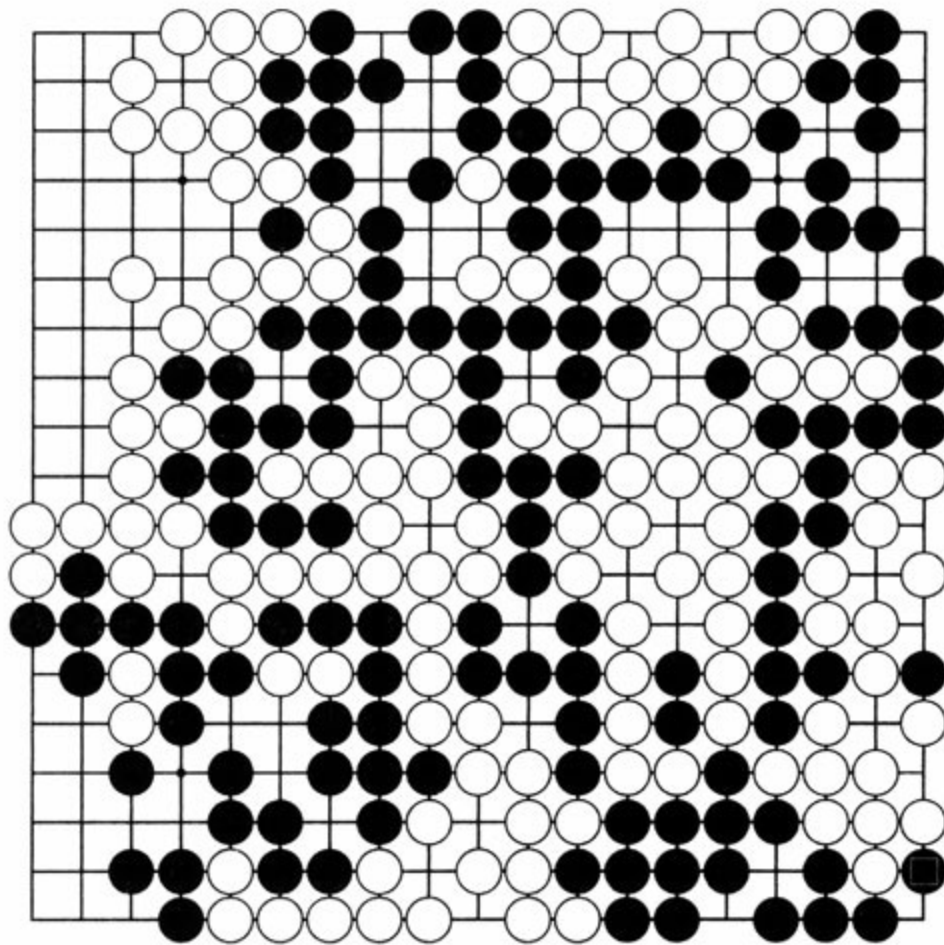
Hãy cùng xem ví dụ. Trên bàn cờ 7 x 7, có 49 giao điểm. Giả sử bàn cờ được chia ra như sau:

Quân đen kiểm soát phía bên trái của bàn cờ, với tổng cộng 28 giao điểm. Quân trắng chiếm giữ bên phải bàn cờ, với 21 giao điểm. Quân đen thắng.

Và đây là một biến thể: Quân đen đã chiếm được một chỗ trong vùng lãnh thổ của quân trắng. Giờ đất của quân đen là 29 và đất của quân trắng là 20:



Việc đếm đất này sẽ dễ nhầm hơn khi bàn cờ lớn hơn với các nhóm quân rải rác ở khắp mọi nơi, ví dụ như ván cờ dưới đây:



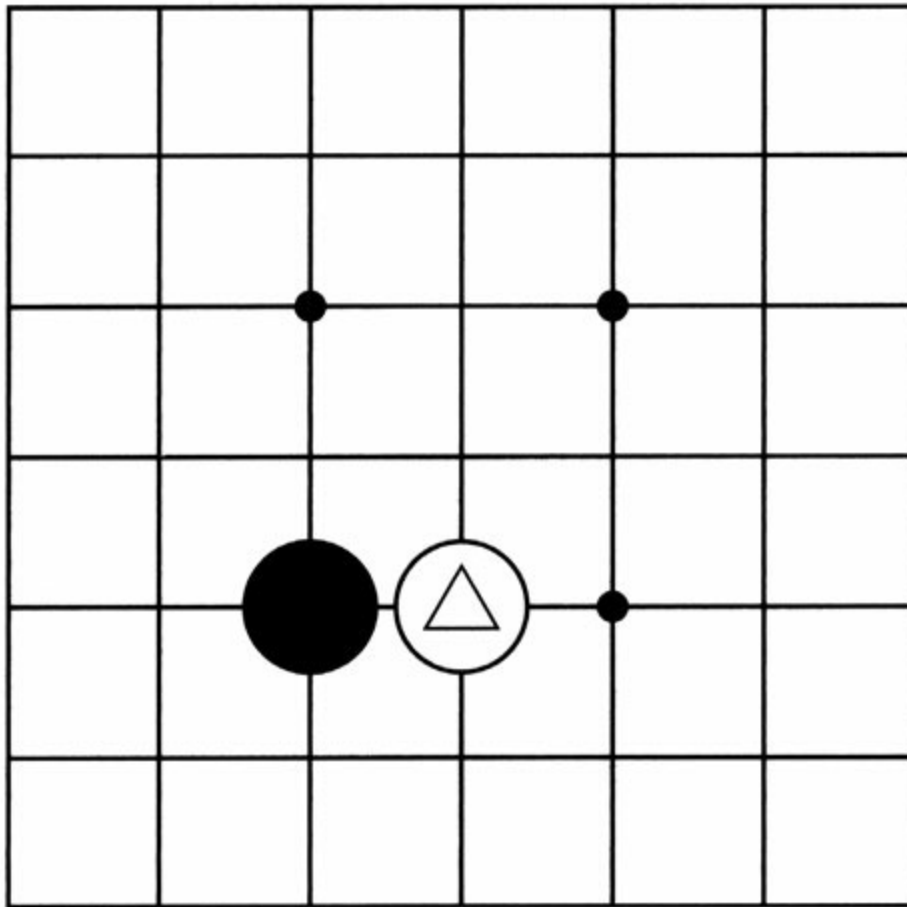
Nguyên tắc chung để đếm đất là: nếu vùng đất nào bị tranh chấp, thì cả hai bên đều không được tính điểm ở vùng đất đó. Đôi khi những người chơi giỏi sẽ nhường đất cho đối thủ, vì thế vùng đất đó sẽ được tính là chiếm được. Nếu có bất đồng về việc ai kiểm soát vùng lãnh thổ nào vào cuối ván cờ, người chơi sẽ tiếp tục chơi cho tới khi tranh cãi được giải quyết.

Tính điểm là một kỹ năng, đòi hỏi phải luyện tập.

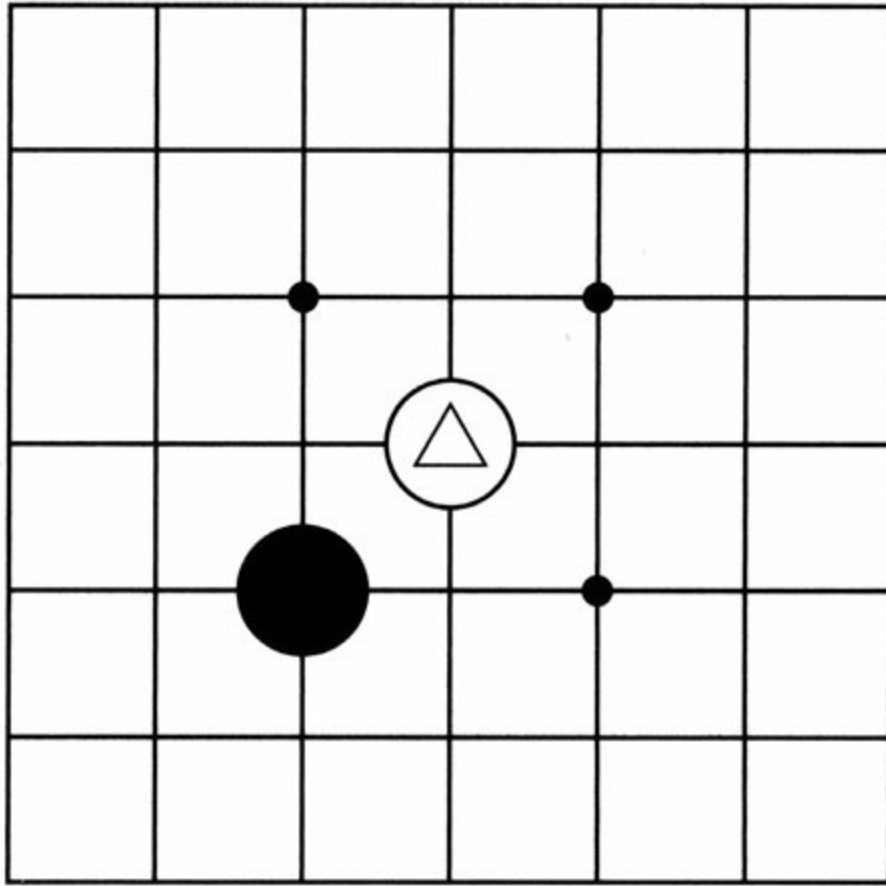
## Tấn công!

Đó là những quy tắc cơ bản. Vậy còn những thủ thuật thông thường thì sao?

Điều đầu tiên cần biết là thế nào là một cuộc tấn công. Tấn công trực tiếp loại khí của quân cờ của đối thủ. Dưới đây là ví dụ cho thấy quân đen đang tấn công quân trắng bằng quân cờ được đánh dấu bằng hình tam giác:



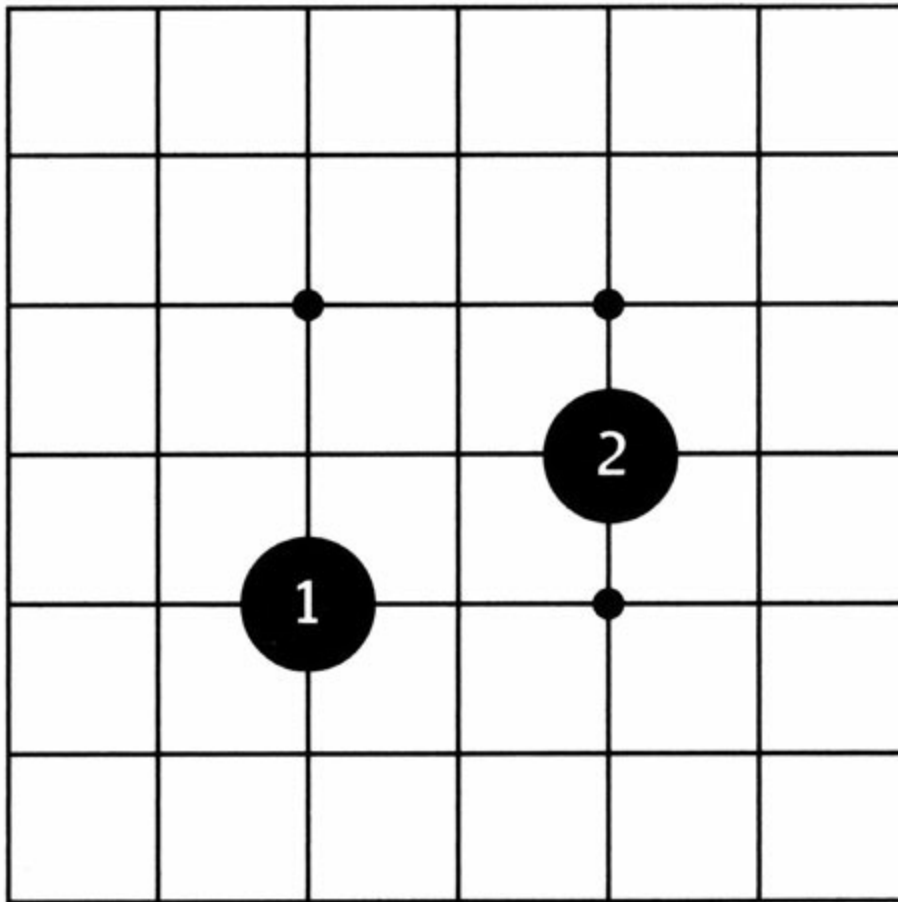
Bạn cũng có thể tấn công gián tiếp, bằng cách đặt một quân cờ gần quân cờ của đối thủ. Tấn công gián tiếp là cách để củng cố vị thế của bạn trong một khu vực trước khi cuộc công kích trực diện của bạn bắt đầu. Hình bên dưới cho thấy quân đen đang đe dọa quân trắng được đánh dấu bằng hình tam giác.



## Mã công

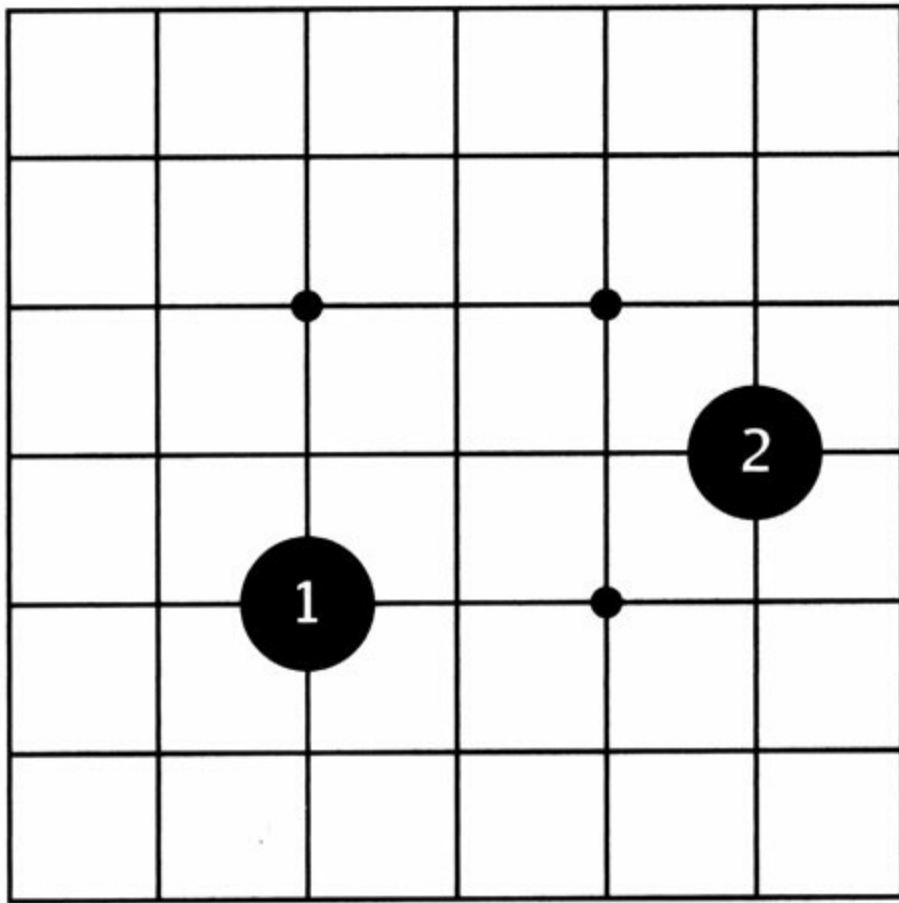
Tân công là yếu tố cơ bản của công. Yếu tố cơ bản của thủ được gọi là *kết nối* và *mối nối*. Những nước cờ cân bằng công và thủ được gọi là *tiếp cận*.

Có một cách tiếp cận phổ biến được gọi là “mã đi tuần” (được đặt tên theo cách di chuyển của quân mã trong cờ vua):



Mã đi tuần là nước cờ cân bằng giữa công và thủ: Nó ở gần các quân cờ khác nên có thể dễ dàng kết nối khi cần thiết, nhưng nó có thể mở rộng ảnh hưởng hơn nước cờ một-bước.

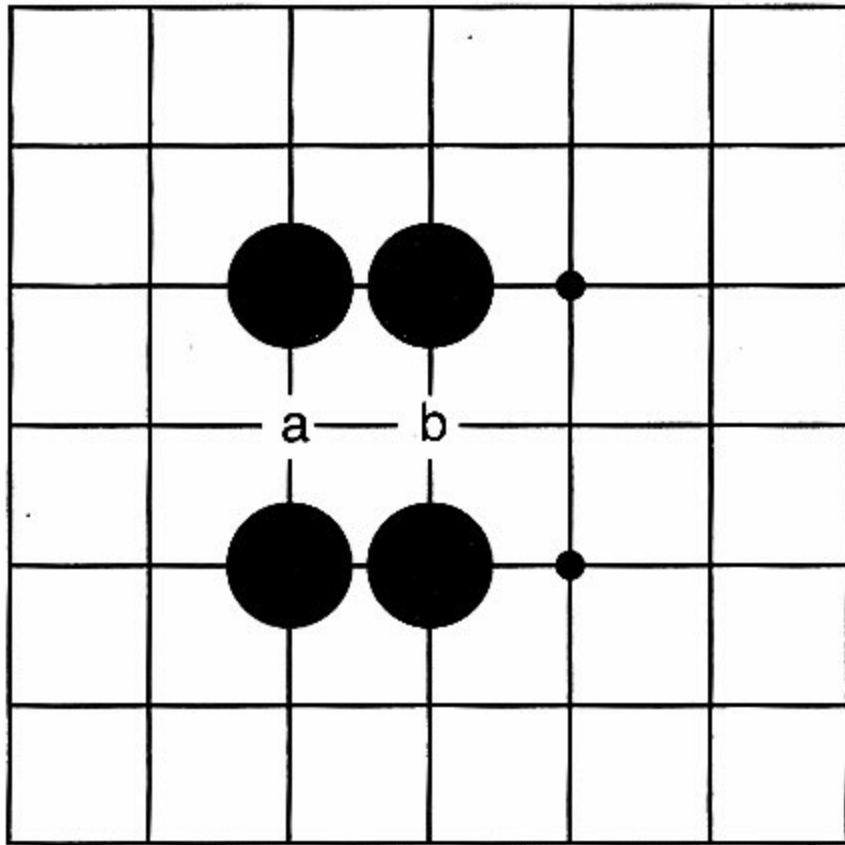
“Mã đi tuần” lớn thậm chí còn hiệu chiến hơn:



Nước cờ này đối thủ lấy sức ảnh hưởng: Nó mở rộng phạm vi của người chơi, nhưng quân cờ đó lại khó có thể được bảo vệ nếu bị tấn công.

## Tre mạnh hơn sắt thép

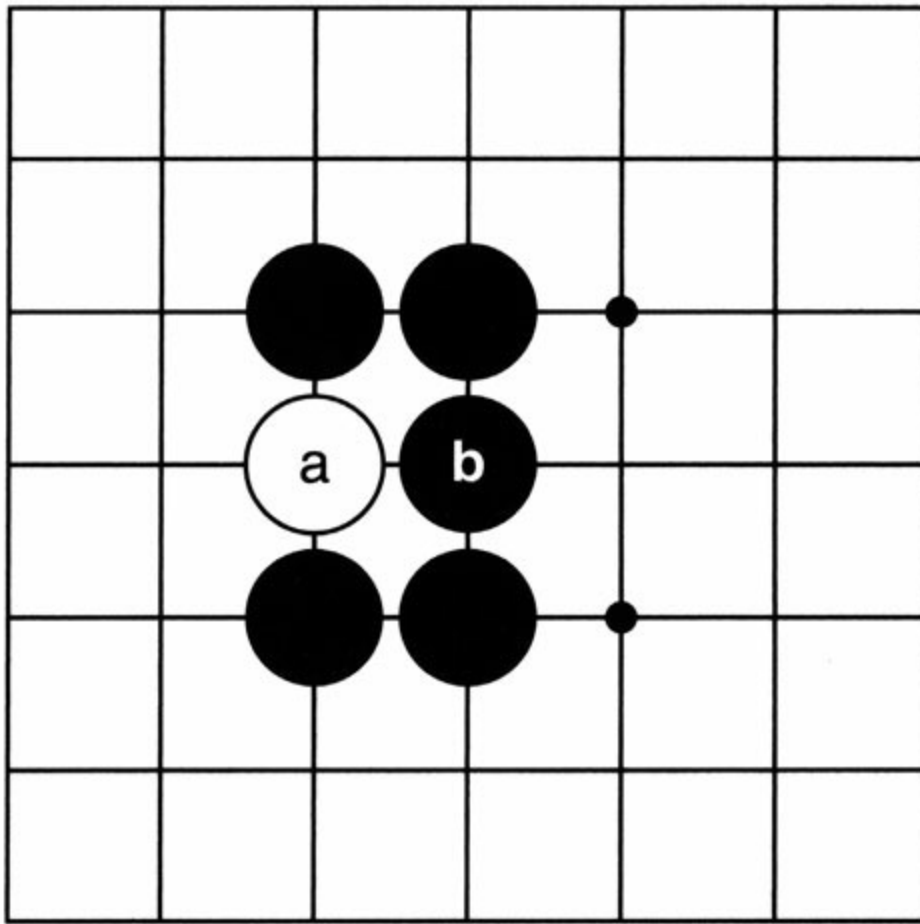
Về chiến lược phòng thủ, có một dạng (thủ) rất vững chắc được gọi là đốt tre (bamboo joint). Chiến lược đó như sau:



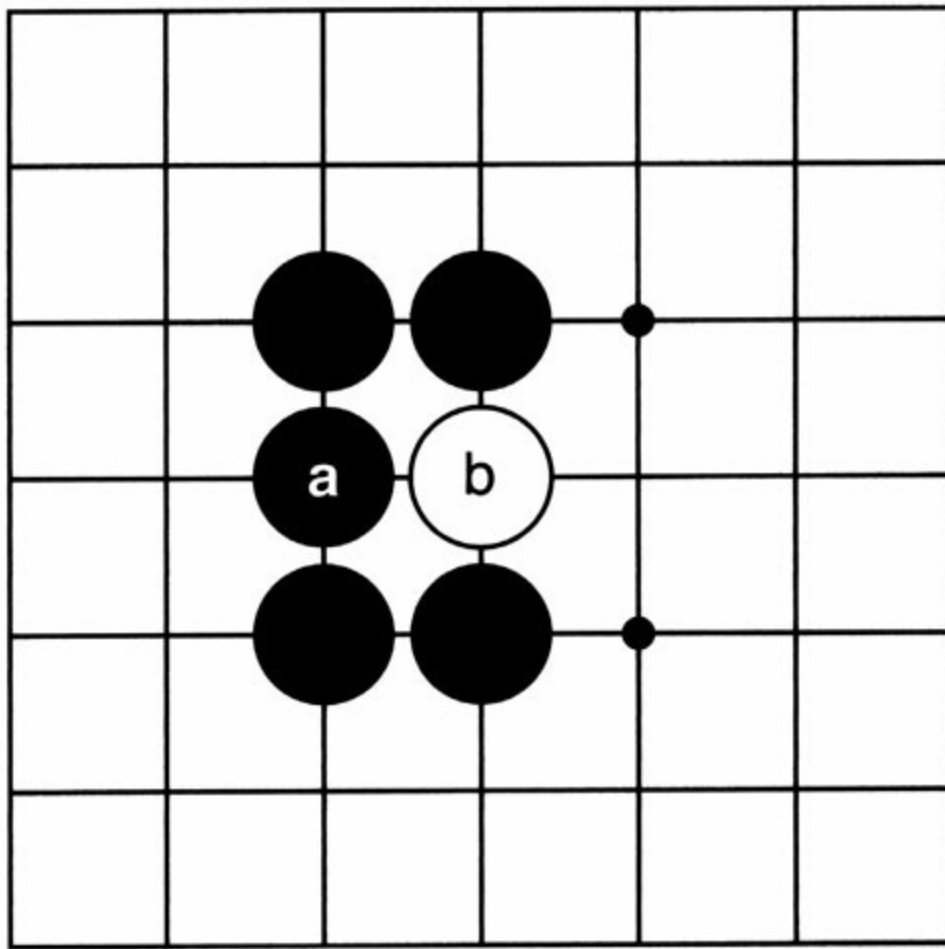
Dạng đột tre rất mạnh vì dù đối thủ của bạn có cố gắng tấn công như thế nào thì bạn vẫn có thể dễ dàng kết nối với quân cờ của mình.

Trong trường hợp này, nếu quân trắng tấn công ở điểm “a”, quân đen có thể kết nối ở điểm “b”.





Tương tự như vậy, nếu quân trắng tấn công ở điểm “b”, quân đen có thể kết nối ở điểm “a”.



Một khi quân đen được kết nối theo kiểu đốt tre, quân trắng sẽ phải mất thêm 9 quân để bắt được cả nhóm. Khó có thể đánh bại được sự bảo vệ đó.

Đơn giản mà, đúng không? Tấn công, phòng thủ và bắt quân có vẻ rất dễ hiểu, và nước cờ mã đi tuần hay đốt tre là những thủ thuật đơn giản dễ hiểu và nhớ.

### Tối đa hóa thời gian luyện tập

Các trang mạng về cờ vây không phải là nơi tốt nhất dành cho những người mới bắt đầu trên mạng. Từ lâu đã có rất nhiều trang web và chương trình như vậy, khoảng từ những năm 1990, nên phải mất một thời gian để tìm ra cách sử dụng chúng.

Có vẻ như không có mấy kỳ thủ thứ hạng thấp. Và nếu bạn không chơi giỏi, thì việc bị thua đi thua lại không phải là một trải nghiệm dễ chịu. Hầu hết những trận đấu mà tôi tìm thấy đều là các trận đấu giữa những kỳ thủ cấp tám hoặc hơn.

Tôi đã cố gắng để xem một số trận, nhưng tốc độ học của tôi rất chậm. Tôi có

thê theo sát được các nước cờ, nhưng lại không hiểu tại sao các kỳ thủ đi nước cờ đó. Đôi khi diễn biến của các nước cờ quá nhanh, và tôi không thể phân tích kịp. Tôi chắc chắn ẩn sau mỗi quân cờ là một hàm ý nào đó, nhưng tôi còn quá non nớt để có thể hiểu được hàm ý đó là gì.

Phải mất cả tiếng để chơi xong một ván cờ vậy. Đây lại là một vấn đề nữa. Nhiều khi tôi bắt đầu theo dõi một trận đấu, nhưng chỉ được một lúc lại phải bỏ đây để vật lộn với Lela hoặc giúp Kelsey làm việc nhà. Vì ván cờ đang diễn ra trực tiếp, nên điều đó có nghĩa là tôi không thể theo dõi trận đấu để phân tích được.

Tôi đã thử cách tiếp cận “trăm trận”, và dựa trên thử nghiệm trước đó của mình, tôi nhận thấy đó không phải là cách hiệu quả nhất để một người mới bắt đầu có thể tiến bộ được. Tôi đã quyết định đổi chiến lược, thay thế việc mò mẫm lúng túng bằng việc nghiên cứu có tổ chức.

## Người đấu với máy

Tối nào tôi cũng dành ra 45 phút để nghiên cứu các thế cờ vậy, trước tiên là bằng cách sử dụng chương trình SmartGo trong iPad của tôi, sau đó là nghiên cứu seri sách *Thế cờ vây xếp hạng* (Graded Go Problems) của Kiseido.

Sau khi nghiên cứu các thế cờ, tôi đã chơi một ván cờ với chương trình trí thông minh nhân tạo (AI), được tích hợp sẵn trong SmartGo trên bàn cờ nhỏ với kích thước 9 x 9. Lúc đầu, SmartGo quật tôi tơi tả, dù đã chấp tôi tới tám quân. Thật là xấu hổ!

Tuy nhiên, khi tôi nghiên cứu, các trận đấu của tôi bắt đầu tiến bộ nhanh chóng. Tôi đã học được cách lợi dụng những quân cờ chấp của mình bằng cách kiên trì tấn công, tấn công quân trắng nhanh gọn và hình thành sợi dây liên kết vững chắc bất cứ khi nào có thể.

Không phải mất nhiều thời gian để giảm số quân cờ được chấp của tôi xuống còn sáu, rồi năm khi tôi thắng AI được vài ván. Sau khoảng 30 ván đấu, tôi chỉ cần chấp bốn quân.

Ước tính điểm số là một việc khó khăn: Đôi khi tôi nghĩ tôi sẽ làm tốt, nhưng khi tính điểm, tôi lại thấy điểm mình không cao. Vì thắng có nghĩa là phải tạo ra những nước cờ có lợi, giúp tăng vùng đất bạn kiểm soát, nên đó là cả một vấn đề. Thông thường, nước cờ mà tôi cho là có thể làm tăng lãnh thổ của

mình thì lại không phải như vậy, lãng phí một nước cờ mà lại tạo cơ hội cho đối phương.

## Các dạng thất bại

Cùng với việc nghiên cứu các thế cờ vây, tôi đã đọc những cuốn sách chọn từ danh sách sách hay nên đọc của Sensei's Library. Những gì tôi học được đã giúp tôi đáng kể.

Nghiên cứu cuốn *Làm cách nào để không chơi cờ vây* thật không phí thời gian. Theo Yuan Zhou, tác giả cuốn sách, người mới bắt đầu thường mắc phải những sai lầm khiến họ phải trả giá đắt:

**Người chơi chưa có kinh nghiệm thường mù quáng đi theo đối phương.** Chẳng hạn, khi bị tấn công, người kém hơn sẽ thường phản ứng bằng cách phòng thủ, bỏ chạy hoặc phản công tại đúng khu vực đó, thay vì xem xét tất cả các khả năng có thể xảy ra.

**Người chơi chưa có kinh nghiệm thường không chú ý đến toàn bộ bàn cờ.** Đôi khi, nước cờ tốt nhất lại không nằm ở nơi xảy ra hành động, mà ở phía bên kia của bàn cờ, cách xa tất cả mọi thứ. Trận đấu thú vị, và người chơi kém thường không nhìn thấy cơ hội ẩn sau những khu vực tĩnh lặng hơn của bàn cờ.

**Người chơi chưa có kinh nghiệm không đi nước cờ có lợi nhất.** Cờ vây đòi hỏi phải suy nghĩ về lợi ích và chi phí: Mỗi nước cờ có thể phải trả giá bằng một quân cờ, và bạn luôn muốn được lợi nhiều nhất cho quân mình. Thường thì điều này có nghĩa là phải hi sinh một hoặc hai quân, để bạn có thể thu được lợi ích lớn hơn ở đâu đó.

**Người chơi chưa có kinh nghiệm không coi trọng tiên thủ (sente).** Trong cờ vây, “khởi đầu” rất quan trọng nếu bạn muốn đoán được càng nhiều càng tốt phản ứng của đối phương. Thay vì tập trung vào đường đi nước bước của họ, bạn muốn đối phương phải lo lắng về việc ngăn những mất mát có thể xảy ra, để lơ là phòng thủ. Người nào có thể tạo ra và duy trì tiên thủ, người đó sẽ luôn giành được phần thắng.

**Người chơi chưa có kinh nghiệm không giới ước tính lãnh thổ.** Kết

quả là, họ sẽ mất rất nhiều thời gian để chiến đấu hoặc bảo vệ một góc nhỏ nào đó trên bàn cờ, trong khi đối phương đã chiếm được một vùng đất rộng lớn không được bảo vệ.

**Người chơi chưa có kinh nghiệm thường ghen tị với vị trí của đối phương.** Zhou gọi đây là “hiện tượng mất đồ” và thường khiến người chơi thực hiện những nước cờ bất lợi chỉ vì cảm thấy vùng đất của đối phương đang ngày càng rộng hơn.

**Người chơi chưa có kinh nghiệm không chống đỡ được những mơ tưởng.** Bạn rất dễ bị cám dỗ bởi một nhóm các quân cờ mà bạn có thể bắt được chỉ bằng một hoặc hai quân, mà không tính toán là bạn không thể đi hai nước cùng một lúc và đối phương sẽ phản ứng với điều bạn làm. Kết quả là bạn lãng phí quân cờ quý giá của mình, chỉ với mơ tưởng hão huyền là đối phương quá ngốc để biết bạn đang làm gì.

Những điều này đều đúng với tôi, đặc biệt là điều cuối cùng về mơ tưởng. Đã nhiều lần tôi tấn công một nhóm, với hi vọng đối phương không nhận ra kế hoạch của mình? Đã bao lần tôi “tham bát bỏ mâm”, lao vào một cuộc chiến nhỏ mà quên mất bức tranh lớn hơn?

Sau khi đọc cuốn *Làm cách nào để không chơi cờ vây*, ván cờ của tôi đã được cải thiện ngay lập tức. Một trong những điều lớn nhất mà tôi cần phải sửa là suy nghĩ cho rằng bắt quân của đối phương là cách ngắn nhất để giành chiến thắng. Trong cờ vua và cờ tướng, điều đó đúng, nhưng trong cờ vây, điều đó lại không đúng.

Bắt quân cũng có ích, nhưng đó không phải là điều kiện cần và đủ để thắng. Bảo đảm an toàn cho vùng lãnh thổ, vùng đất của mình mới là mục tiêu của cờ vây, và nhiều khi, bạn có thể hoàn thành được mục tiêu đó mà không cần phải bắt bất cứ quân nào của đối phương. Phải mất một lúc mới “tiêu hóa” nội bài học đó: Bắt quân có vẻ tốt, nhưng thực ra nó có thể phản tác dụng.

## Cách ngôn

Cách ngôn là công cụ dạy cờ vây hiệu quả. Vì ván cờ rất phức tạp nên các quy tắc giúp người chơi có thể nhớ được cách chơi trong những tình huống thông thường.

Một trong những ví dụ đầu tiên về cách dạy này là “Mười quy tắc vàng của cờ vây” của Wang Jixin – một kỳ thủ cờ vây đời Đường.

1. “Tham lam không đem lại phần thắng”.
2. “Không được vội vã khi xâm nhập lãnh thổ của đối phương”.
3. “Đề ý đến quân ta khi tấn công quân địch”.
4. “Hi sinh một quân cờ để nắm quy ên tiên thủ”.
5. “Bỏ nhỏ để cứu lớn”.
6. “Khi gặp nguy, hi sinh”.
7. “Tránh nước cờ nóng vội”.
8. “Nước cờ cần phải phản lại đối phương”.
9. “Chống những vị trí mạnh mẽ, chơi an toàn”.
0. “Tìm kiếm an toàn, tránh chiến đấu ở khu vực biệt lập hoặc trong tình huống yếu thế”.

Có rất nhiều câu châm ngôn giống như vậy về cờ vây:

- Điểm trọng yếu của đối phương cũng là điểm trọng yếu của mình.
- Khi nghi ngờ, hãy thoát tiên (*tenuki* – chơi ở chỗ khác).
- Có khả năng chết trong nước đi chéo giữa hai quân mình sát với quân của đối phương.
- Người bắt đầu chơi chẹt khí (*atari*).
- Hình cờ hoa (*ponnuki*) đáng giá 30 điểm.
- Hạ quyết tâm trước khi tấn công.
- “Thả con săn sắt bắt con cá rô”.

Người giàu không cần tranh chấp vật.

Đã cưỡi lên lưng hổ thì khó có thể xuống được. (Đừng bắt đầu việc gì mà bạn không sẵn sàng để kết thúc.)

Có rất nhiều thứ để học: Không những phải học các câu cách ngôn này và biết tại sao chúng tồn tại, mà bạn còn cần phải hiểu khi nào thì bỏ qua chúng. Vì suy cho cùng, chính cách ngôn cũng dạy chúng ta rằng: “Không được mù quáng nghe theo những câu cách ngôn”.

Còn có những câu cách ngôn dành riêng cho người mới bắt đầu. Câu cách ngôn yêu thích của tôi là “Chỉ sau cú đấm thứ 10 bạn mới nhìn thấy nắm đấm”, và sau đó là “Chỉ sau cú đấm thứ 20 bạn mới có thể chặn được chúng”. Theo kinh nghiệm của tôi từ trước tới giờ, điều đó rất đúng: Những người kỳ cựu có thể tấn công tôi theo những cách mà tôi thậm chí còn không nhận ra.

Tôi còn phải học nhiều lắm.

## Đạt mốc 20 giờ

Dựa trên ghi chú của tôi, sau khoảng một tháng nghiên cứu, tôi đã đạt mốc 20 giờ. Tôi đang ở đâu và tôi đã hoàn thành được gì?

Theo cơ sở dữ liệu trong SmartGo, tôi đã giải quyết được 150 thế cờ xếp hạng, và đã xử lý đến các thế cờ của cấp 18. Tôi đã chơi 33 ván cờ, và hiện tôi chỉ cần được chấp bốn quân khi chơi với chương trình trí thông minh nhân tạo.

Tôi cũng đã hoàn thành cuốn đầu tiên trong seri *Thế cờ vây xếp hạng* của Kseido, tương đương với cấp từ 30 đến 25. Tôi đang đọc cuốn sách thứ hai, tương đương với cấp từ 25 đến 15, vậy nên có thể ước lượng tôi đang ở khoảng cấp thứ 18 đến 20. Cũng không tệ.

## Tóm lược lại phương pháp

Hãy cùng tóm lược lại phần cốt lõi của phương pháp mà tôi đã sử dụng để học cách chơi cờ vây:

Tôi đã học những quy tắc cơ bản của trò chơi.

Tôi đã mua bàn cờ, quân cờ và một số cuốn sách về chiến lược cờ vây dành cho người mới bắt đầu.

Tôi đã tìm ra vài cách để học những nguyên tắc chiến lược quan trọng nhất bằng cách giải quyết các thế cờ vây xếp hạng, cả trên máy tính và trong sách.

Tôi dành quãng thời gian luyện tập ban đầu để giải quyết các thế cờ vây xếp hạng, lặp lại những bài tập mà tôi không thể trả lời hoặc giải thích trong lần thử đầu tiên.

Tôi luyện tập các thế cờ và chơi cho tới khi đạt mốc 20 giờ. Lúc này, tôi đã có thể thoải mái giải quyết các thế cờ cấp 20, tăng 15 cấp từ cấp thứ 35 lúc mới đầu.

## Từ đây tôi sẽ đi đâu tiếp?

Tôi có cảm giác lẩn lộn về cờ vây. Tôi đã học được tương đối về trò chơi này, đủ để hiểu được nó sâu xa thế nào, và đủ để biết tôi còn cách danh hiệu bậc thầy cờ vây xa như thế nào. Tôi đã tiến bộ đáng kể từ khi bắt đầu, nhưng thực sự để chơi giỏi trò này, tôi còn phải đi một chặng rất dài nữa.

Mặt khác, thời gian rảnh của tôi rất hạn chế, trong khi cờ vây có vẻ cần phải chuyên tâm, tập trung cao độ giống như viết mã và lập trình. Dù cờ vây cũng thú vị, nhưng sau một ngày dài, có cảm giác chơi cờ vây cũng giống như *làm việc* vây.

Thêm nữa, thường phải mất cả tiếng đồng hồ mới hoàn thành được một ván cờ trên bàn cờ 19 x 19, và thật không lịch sự khi người chơi phải dừng cuộc chơi để hoàn thành một nghĩa vụ gia đình nào đó. Từ trước đến giờ, hầu hết các ván cờ của tôi đều là chơi với máy tính, vì máy tính lúc nào cũng sẵn sàng và không quan tâm tới chuyện tôi phải dừng lại để làm việc gì đó một lúc.

Tôi bắt đầu đánh giá cao đặc tính phức tạp đến đau đầu nhức óc của các thế cờ vây, vì thế tôi sẽ tiếp tục giải các thế cờ đó khi có cảm giác muốn giải các câu đố. Tôi thích luyện tập cờ vây hơn là giải ô chữ sudoku hay những trò chơi kiểu như vậy, vì vậy các chương trình cờ vây đã được cài vào điện thoại của tôi, luôn sẵn sàng cho tôi chơi bất cứ khi nào tôi có chút thời gian rảnh rỗi.



Tôi định dạy Lela cách chơi khi con bé lớn hơn một chút. Các quy tắc cũng đơn giản, một đứa trẻ bốn tuổi có thể hiểu được, và tôi cũng từng có những kỷ ức đẹp khi chơi cờ vua với bố tôi khi còn nhỏ. Cờ vây là cách tốt để dạy trẻ những kỹ năng quan trọng như phân tích, chiến lược hay trao đổi, và tôi nghĩ là bố con tôi sẽ có nhiều thời gian thú vị trong những năm sắp tới.

Hơn thế nữa, tôi không có khao khát cháy bỏng là phải dành thêm thời gian để học cờ vây. Tôi đã học về nó đủ để thỏa mãn trí tò mò của mình, và tiếp tục học không còn là ưu tiên lớn của tôi nữa.

Thế là đủ rồi. Chẳng có luật nào quy định bạn phải học tới khi giỏi mọi thứ mà bạn đã từng học. Cuộc sống phải đánh đổi, và chẳng có gì sai khi khám phá những điều mới mẻ, học một chút, sau đó quyết định khám phá điều gì đó khác.

Bạn không cần phải đạt đai đen trong mọi việc mới có thể sống cuộc đời thoải mái, thỏa mãn được.

## BÀI HỌC: VẤN ĐỀ MÔI TRƯỜNG

**Chúng ta chiến đấu không phải bằng máu và thịt, mà bằng cầu oxy, sợi carbon, bằng cái tôi, ước mơ và bằng ý chí bất khuất của chúng ta.**

**–CHRIS ZEITVOGEL, người chơi lướt ván**

...

Tôi đang ở giữa hồ, liềm mình để căng buồm lên. Thật là khó, vì các cơ ở cánh tay, ở chân và ở lưng của tôi đều đang kêu gào.

Một cơn gió tây bắc mạnh cuốn tung những gợn sóng nhỏ khiến con thuyền dưới chân tôi chao đảo. Tôi để cho cánh buồm rơi xuống nước, tạm dừng một lúc để nghỉ ngơi và điều chỉnh sự rung lắc của con thuyền dưới chân.

Tôi nhìn về hướng tây: Một cơn bão đang hình thành, đẩy các đám mây đen về phía núi. Đã đến lúc để quay thuyền lại.

Tôi cúi xuống với đoạn dây kéo, sau đó kéo, nâng thuyền của tôi ra khỏi vùng nước ít nhất là 13 lần trong vòng 30 phút. Tôi hi vọng có thể trở lại đất liền trước khi cơn bão ập xuống.

Bão gần lắm rồi... Tôi không giỏi lái cái máy kỳ cục này, nhưng tôi phải học. Chèo thuyền quay lại bờ biển sẽ khiến tôi già nhanh mắt.

## Cần nhắc ý tưởng lướt ván

Ý tưởng đầu tiên của tôi là chèo thuyền: Hồ đủ rộng để chèo thuyền. Đáng tiếc là thuyền có mái chèo tiêu chuẩn lại to và kênh cang, thậm chí những mái chèo nhỏ nhất cũng rất dài và dễ gãy. Đây là cả một vấn đề vì tôi không có khả năng giữ an toàn cho một chiếc thuyền to và dài trên hồ.

Vấn đề tương tự cũng xảy ra với thuyền buồm, cộng thêm vấn đề chi phí nữa: Ngay cả một thiết bị cơ bản nhất cũng tốn cả đồng tiền. Ngoài vấn đề phức tạp liên quan đến việc neo đậu, tôi không muốn phải bỏ hàng chục nghìn đô la cho một bộ cơ cấu truyền động.

Khi nghiên cứu các lựa chọn của mình, tôi tình cờ xem được một video của James Douglass, một giáo sư về sinh học biển ở trường Đại học Florida Gulf Coast. Ngoài là một chuyên gia về sinh học, Jim còn là một người chơi lướt ván lão luyện và đã chia sẻ rất nhiều thông tin hữu ích về môn thể thao này trên trang blog riêng của ông.

Trong video, Jim đã trang bị cho chiếc thuyền buồm Formula-class một chiếc máy quay HD chống nước, được gắn trên đầu sào căng buồm, quay được mọi hoạt động cùng một lúc. Sau đó, ông lao xuống nước, lướt trên những con sóng nhỏ và thực hiện những cú đảo chiều nhanh và ấn tượng.

Tôi bị choáng ngợp và đã xem đi xem lại video đó vài lần. Tôi chưa bao giờ chơi lướt ván, nhưng đột nhiên tôi lại muốn thử.

Lướt ván thỏa mãn mọi tiêu chí tôi đề ra: Tôi có thể tự mình trải nghiệm và trang thiết bị cần thiết cũng không lớn hay cồng kềnh. Và dựa trên những tìm hiểu do tò mò của tôi, đó cũng không phải thú vui quá xa xỉ. Tôi cần mua thiết bị, nhưng những thiết bị cơ bản không thể khiến tôi phá sản được.

Lướt ván có vẻ thú vị, nhưng trước khi vào cuộc, tôi muốn chắc chắn nó phải an toàn. Bản chất tôi không phải là một người ưa mạo hiểm. Nếu lướt ván là một môn thể thao nguy hiểm, tôi sẽ chọn chơi một môn khác.

Nếu chơi lướt ván, tôi muốn biết các nguy cơ có thể xảy ra trước khi chơi. Chuyện gì có thể xảy ra?

## Lướt ván có nguy hiểm không?

Lướt ván thường được gắn mác là môn thể thao “mạo hiểm”, nhưng mức độ mạo hiểm lại phụ thuộc vào kiểu lướt mà bạn muốn thực hiện. Trên những mặt hồ phẳng lặng, khép kín kiểu như mặt hồ mà tôi sẽ học, điều khiển một chiếc thuyền nhỏ sẽ thú vị, phấn khích hơn. Đó không phải là việc điên khùng.

Ngược lại, lướt ván trên biển sẽ có nhiều nguy hiểm hơn. Sóng xô vào bờ, khiến cho việc lướt ván khó khăn hơn, và nếu thuyền ở xa bờ thì đó là một vấn

đề rất lớn. Biển thường đem lại nhiều niềm vui hơn, kiểu như nhảy lên những sóng, nhưng cũng gia tăng nguy cơ. So với việc lướt ván ngoài biển, lướt ván ở sông hồ trong đất liền an toàn hơn nhiều.

Tuy nhiên, dù bạn chọn lướt ván ở đâu thì vẫn có những nguy cơ tiềm tàng, mà chủ yếu là chết đuối và giảm thân nhiệt. Khi bạn đứng trên một miếng ván trôi nổi ở giữa dòng nước, dùng tay giữ cho chiếc thuyền đứng thẳng trong một ngày nhiều gió, những chuyện xấu có thể xảy ra nếu bạn không chuẩn bị trước.

Nguy cơ chết đuối bắt nguồn từ hai nguyên nhân chính: bị thương và kiệt sức. Nếu bạn ngã theo cách có thể khiến cho chân tay bị thương nghiêm trọng hoặc chấn động, bạn có nguy cơ bị bất tỉnh dưới nước. Nếu bạn mệt tới mức không thể quay trở lại ván, hoặc không thể điều khiển ván quay trở lại bờ được, chết đuối là khả năng hoàn toàn có thể xảy ra.

Cách tốt nhất để ngăn việc chết đuối là (1) sử dụng thiết bị nổi cá nhân (thường được gọi là áo phao) và (2) luôn đi cùng người khác, dù ở trên bờ hay trên mặt nước. Nếu xảy ra vấn đề gì, bạn cần có người nào đó ở bên để giúp đỡ.

Giảm thân nhiệt là một vấn đề nguy hiểm tới tính mạng. Trong điều kiện lạnh, ướt và có gió, cơ thể bạn mất nhiệt rất nhanh. Một khi thân nhiệt của bạn tụt xuống dưới 35 độ C, tim, phổi và hệ thần kinh của bạn bắt đầu ngừng hoạt động, dẫn đến tử vong, dù nhiệt độ cơ thể bạn đang khôi phục lại mức độ bình thường (khoảng 37 – 37,5 độ C).

Điều khiến cho việc giảm thân nhiệt trở nên đặc biệt nguy hiểm là nó xảy ra từ từ. Khi nhiệt độ cơ thể giảm, những triệu chứng như run rẩy, rối loạn, mất kiểm soát và mệt mỏi sẽ xuất hiện, làm giảm độ linh hoạt và khả năng phán đoán. Nếu bạn phải dựa vào khả năng phán đoán và kỹ năng vận động để tìm đường về nhà, thì sự tụt công của hiện tượng giảm thân nhiệt là một nguy cơ lớn nhưng lại hay bị coi thường.

Khi nhiệt độ nước trên 25 độ C, giảm nhiệt không phải là một nguy cơ nghiêm trọng. Dưới ngưỡng 25 độ C đó, cần phải đảm bảo chắc chắn bạn có lớp cách nhiệt trước khi mạo hiểm dấn thân vào dòng nước. Đó chính là lúc cần đến quần áo lặn.

Quần áo lặn được làm bằng chất liệu mỏng, linh hoạt, có khả năng cách nhiệt dưới nước kiểu như cao su tổng hợp. Những bộ quần áo lặn hiện đại được phân loại dựa trên độ dày (tính bằng milimét) của chúng, và thường được thiết kế phần thân dày hơn so với chân tay. Thiết kế này nhằm hai mục đích: cách

nhiệt tốt hơn ở phần thân, giúp giữ nhiệt cho cơ thể hiệu quả hơn, còn phần chân tay mỏng hơn, giúp cho việc cử động dễ dàng hơn. Khi lướt ván trong điều kiện lạnh, đồ của bạn cần có tính linh hoạt và cả sự cách nhiệt.

## Chuẩn bị

Nguồn thông tin đầu tiên của tôi là trang web của Jim với hàng tá bài viết về bí quyết dành cho người mới bắt đầu. Một trong những bài viết đầu tiên mà tôi đọc là “T rả lời 18 câu hỏi thường gặp nhất về lướt ván”, trong đó có hướng dẫn về các thiết bị cần thiết.

Từ danh sách Q&A (Hỏi & đáp), có vẻ như tôi sẽ cần ít nhất một tấm ván, một cánh buồm, một áo phao và một bộ quần áo lặn. Tôi đọc tất cả các bài viết về lướt ván trên trang web của Jim, lập một danh sách các thiết bị có vẻ hữu ích, cố gắng cắt giảm danh sách đó xuống còn hai lựa chọn mà tôi cho là tốt nhất, sau đó viết cho Jim một e-mail để cảm ơn ông, nhân tiện xin ông lời khuyên.

Điều quan trọng nhất là loại ván và buồm mà tôi cần có. Video lướt trên tấm ván Formula-class của Jim rất ấn tượng, và tôi thích mình cũng có thể làm được như vậy, nhưng tôi không biết là bắt đầu với một tấm ván Formula có phải là một ý tưởng tốt hay không. Theo nghiên cứu của tôi thì một lựa chọn khác là tấm ván hiệu Rio do công ty Starboard sản xuất. Rio nổi tiếng là thân thiện (dễ sử dụng) với người mới bắt đầu, và cũng là loại ván phù hợp với hầu hết các điều kiện: nó không bay nhanh như ván Formula, nhưng có thể được sử dụng trong nhiều điều kiện khác nhau.

Tôi không biết phải làm gì, nên tôi đã hỏi ý kiến của Jim:

*Tôi là một người hoàn toàn mới đối với môn lướt ván – thậm chí còn chưa đặt chân lên tấm ván lần nào.*

*Tôi hi vọng ông có thể cho tôi vài lời khuyên để bắt đầu.*

*Tôi đang tìm mua đủ bộ thiết bị lướt ván, và tôi đang phân vân lựa chọn giữa tấm ván Rio cỡ M và Starboard Formula 167.*

*Rio có vẻ là tấm ván phù hợp với người mới bắt đầu, nhưng tôi muốn học [lướt trên mặt nước] nhanh nhất có thể. Tôi chỉ lướt ván trên mặt nước phẳng lặng, và lướt trong điều kiện ít gió.*

*Ván Formula, từ những gì tôi đọc đến bây giờ, lướt nhanh hơn và tốt hơn trong điều kiện ít gió.*

*Xin hỏi là: Học bằng ván Formula có sao không? Tôi có khả năng kiên nhẫn cao độ với những khó chịu ban đầu, và tôi thích mua một tấm ván đơn (chức năng) hơn là một tấm ván đa chức năng. Nếu học bằng ván Formula, liệu tôi có nguy cơ làm gãy nó và bị thương không?*

*Cảm ơn bất cứ lời khuyên nào mà ông đưa ra. Tôi rất trân trọng sự giúp đỡ của ông!*

Khoảng một tiếng sau, Jim trả lời:

1. Việc ván Formula tốt hơn trong điều kiện gió nhẹ là một hiểu lầm. Loại ván này đúng là có tiềm năng lướt trong điều kiện gió nhẹ hơn bất cứ loại ván nào khác (7 – 8 hải lý), nhưng bạn cần phải là một thủy thủ lão luyện để có thể điều khiển một cánh buồm rộng khoảng 11 – 12 mét vuông thì mới đạt được tiềm năng đó.
2. Chắc chắn bạn sẽ không bị thương nếu cố gắng học bằng một tấm ván công thức, trừ khi cố kéo một chiếc buồm quá to có thể khiến bạn bị đau lưng. Lý do khiến bạn không làm mình bị thương là vì nhiều khả năng bạn sẽ không thể làm cho tấm ván di chuyển được một chút nào! Tuy nhiên, bạn hoàn toàn có thể “làm bị thương” tấm ván vì ván công thức có bề mặt mỏng tới mức có thể dễ dàng kêu vang khi bạn làm rơi nó lúc ngã.
3. Ván Rio cỡ M có lẽ là loại ván tốt nhất cho cả việc học cấp tốc và lướt thoải mái, dù GO 171 có thể là một lựa chọn tốt.
4. Chiến lược của tôi là nên có một số loại buồm với các kích thước khác nhau để bạn có thể tăng giới hạn sức mạnh mà bạn có thể xử lý lên, bất chấp sức gió. Với những bước đầu tiên, bạn sẽ cần một cánh buồm nhỏ hơn sáu mét vuông... Tôi có một thiết bị đo sức gió giúp xác định cần phải sử dụng buồm cỡ nào để có thể lướt được trong điều kiện sức gió như vậy.
5. Cơ hội tốt nhất để bạn có thể lướt được gắn liền với buồm loại nhỏ trong điều kiện gió mạnh, vì cần ít kỹ thuật để có thể lái được một cánh buồm loại nhỏ trong điều kiện gió mạnh hơn so với một cánh buồm loại lớn trong điều kiện gió nhẹ.

Những thông tin này đúng là rất quý. Jim đã xóa bỏ một số hiểu lầm của tôi

về loại ván mà tôi cần có để bắt đầu, cũng như loại buồm mà tôi nên sử dụng khi bắt đầu. Lòng tốt của Jim khi chia sẻ, đưa ra lời khuyên đã giúp tôi tiết kiệm được vài nghìn đô la, và giúp tôi tránh được tâm trạng thất vọng.

Lúc đầu, tôi muốn mua một chiếc buồm lớn, nhưng Gary – người đã có hơn 20 năm dạy lướt ván – đã nhanh chóng khuyên tôi bỏ ý định đó. Ông nói buồm lớn có thể tốt trong điều kiện gió nhẹ hơn, nhưng chỉ trong trường hợp khi tôi đã biết cách sử dụng nó.

Điểm trừ của buồm lớn là chúng nặng hơn và khó nâng lên khỏi mặt nước hơn. Nếu tôi bắt đầu với một chiếc buồm lớn khi vẫn phải tìm hiểu cách giữ thăng bằng và cách rẽ sóng, có nghĩa là tôi đã “chán sóng”, theo lời Gary nói.

Tôi không muốn “chán sóng”, nên tôi nghe theo lời khuyên của Gary và chọn một chiếc buồm nhỏ hơn. Một lần nữa, cần phải nghe theo lời khuyên của người đi trước dày dặn kinh nghiệm hơn trước khi đưa ra quyết định. Sau khoảng nửa tiếng nói chuyện với Gary, tôi đã quyết định đặt hàng của Isthmus.

Hóa ra các công ty thương mại vận tải tàu thủy có thể, và sẵn sàng chuyển những đồ ngoại cỡ kiểu thiết bị lướt ván tới bất cứ nơi nào bạn muốn, nên việc mua sắm thiết bị đơn giản hơn tôi tưởng. Gary dự tính phải mất khoảng một tuần để nhận được các vật dụng cần thiết, nên trong khi chờ đợi, tôi bắt đầu tự học cách sử dụng chúng.

Dừng lại, anh bạn tay mơ!

Một chút thời gian duyệt web dẫn tôi tới một số nguồn giới thiệu về lướt ván:

, *Hướng dẫn dành cho người mới bắt đầu và nghệ thuật lướt ván* của Frank Fox (1988)

, *Lướt ván* của Peter Hart (2005)

, *Học lướt ván trong một tuần* của Phil Jones (1992)

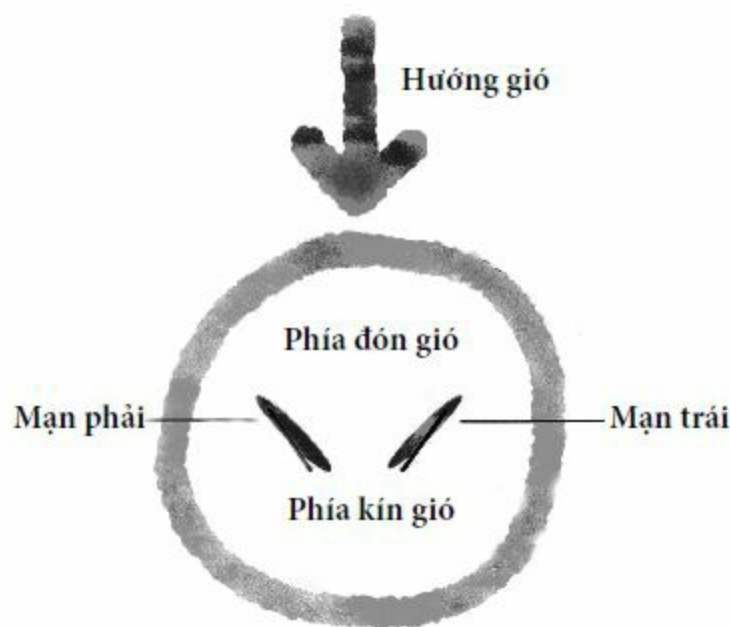
, *Người mới bắt đầu tới người thành công* (DVD) của Jem Hall (2006)

Những nguồn tài liệu này đều làm tốt công việc giải thích các thuật ngữ, lý

thuyết và kỹ thuật cơ bản. DVD hướng dẫn của Jem Hall đặc biệt làm tốt công việc giảng giải cách xử lý ván khi ở dưới nước: Giảng giải các chuyển động phức tạp bằng cách làm mẫu sẽ dễ dàng hơn. Trong trường hợp này, sách bất lợi hơn, vì ngay cả những hình minh họa tốt nhất cũng có thể gây nhầm lẫn nếu so với một video hướng dẫn chi tiết.

Có một điều khiến tôi ấn tượng ngay lập tức là có rất nhiều kiến thức chuyên ngành hàng hải được sử dụng trong lướt ván. Người lướt ván cũng được gọi là những “thủy thủ” và các thuật ngữ được sử dụng trong tài liệu hướng dẫn đều bắt nguồn trực tiếp từ ngành hàng hải.

Đã nhiều năm trôi qua kể từ khi tôi đi thuyền, thế nên tôi cần phải xem lại. Nếu bạn không biết các thuật ngữ thì việc đọc sách hướng dẫn có thể rất khó chịu, vì vậy quan trọng là bạn cần phải biết những điểm chính. Dưới đây là một vài điểm như thế:



- **Hướng gió** – các hướng đông, tây, nam, bắc không mấy hữu ích trong khi đi thuyền, vì gió liên tục thay đổi. Hướng thường được gắn liền với hướng gió: *phía đón gió* có nghĩa là di chuyển vào vùng gió, còn *phía kín gió* có nghĩa là di chuyển cùng với gió.

• **Hướng tàu** – tương tự như vậy, “trái” và “phải” có liên quan tới hướng đi hiện tại của tàu, thuyền. *Mạn trái* có nghĩa là phía bên trái của tàu thuyền, còn *mạn phải* có nghĩa là bên phải.



**Trước và sau** – phía trước của tàu thuyền được gọi là *mũi tàu* còn phía sau được gọi là *đuôi tàu*. Hướng về mũi tàu được gọi là ở *đằng mũi tàu*, còn hướng về đuôi tàu được gọi là ở *đuôi tàu*.

**Quay** – thuật ngữ dùng để chỉ việc quay tàu là *đổi hướng* và *trở buồm*. Sự khác nhau giữa chúng nằm ở chỗ đầu nào của tàu/thuyền sẽ đi xuyên qua gió trong lúc quay tàu. Nếu bạn di chuyển vào vùng có gió và quay đầu, nghĩa là bạn *đổi hướng*. Nếu bạn ra khỏi vùng có gió và quay đầu, nghĩa là bạn đang *trở buồm*.

## Để ý gió

Ngoài việc đọc sách, tôi cũng bắt đầu chú ý nhiều hơn tới thời tiết. Trước khi quan tâm tới lướt ván, tôi không bao giờ để ý đến gió, trừ khi gió thổi đặc biệt khác thường. Còn bây giờ, tôi thấy mình thường xuyên theo dõi gió, nghiên cứu cẩn thận ngọn cây để ước lượng tốc độ gió.

Tất nhiên, “loại” gió duy nhất thực sự quan trọng là gió ở vùng nước mà bạn dự định lướt ván. Để kiểm tra tốc độ và hướng gió ở hồ, tôi đã chọn mua một thiết bị đo gió bỏ túi có tên Kestral 3000. Thiết bị nhỏ nhưng tiện dụng này giúp tôi kiểm tra được tốc độ gió hiện tại ở hồ, thế nên, thông tin đó chính xác hơn so với thông tin trên mạng. Chiếc máy đó cũng có thể đo được cả nhiệt độ nước và không khí, vì thế tôi có thể đảm bảo sẽ không ra ngoài (chơi lướt ván) nếu nguy cơ giảm nhiệt cao.

Khi theo dõi gió trong vài ngày, tôi bắt đầu chú ý tới các kiểu gió. Ở hồ, gió nhẹ thường thổi từ hướng đông vào buổi sáng, sau đó tăng tốc và chuyển hướng thổi từ hướng bắc – tây bắc vào buổi chiều (thường là vào khoảng bốn giờ chiều). Vì tôi sẽ hạ thuyền từ hướng đông của hồ, nên tốt nhất tôi nên đi vào lúc chiều. Bốn giờ cũng là thời điểm lý tưởng cho lịch trình chung của tôi: Nếu tôi sắp xếp ngày làm việc hợp lý, tôi có thể ra khỏi nhà trong khoảng một tiếng hoặc lâu hơn mà không bị làm phiền.

## Lắp ghép

Lúc này tôi đã có tất cả các thiết bị. Tôi cần phải xác định được cách lắp ván và buồm lại với nhau – quá trình này gọi là *lắp ghép*.

May mắn là Isthmus đã hướng dẫn cho tôi. Lắp ghép là cả một vấn đề đối với những người mới sở hữu thuyền lướt sóng như tôi, vì đó là một quá trình phức tạp liên quan đến nhiều bộ phận khác nhau. Khi bạn cố gắng xác định cái nào ở chỗ nào, bạn sẽ rất dễ bị nhầm.

Isthmus đã giải quyết vấn đề này bằng cách cung cấp một video chi tiết, vì thế bạn có thể theo dõi quá trình lắp ghép và làm theo. Video hướng dẫn là điều lý tưởng cho việc học kiểu này: Tôi xem video hướng dẫn vài lần và ghi chép lại trước khi thử lắp ghép thiết bị của mình.

Nhờ có những ghi chép của mình, tôi đã có thể lắp ghép thuyền lướt sóng của tôi trong khoảng 30 phút: Không tệ lắm cho lần thử nghiệm đầu tiên. Tôi đang mặc quần áo lặn, thuyền của tôi đã sẵn sàng và gió đang thổi. Đã đến lúc để “hạ thủy”.

## Cuộc hành trình đầu tiên

Tôi đem ván và buồm xuống mép nước, gắn cột buồm vào bệ, sau đó thả ván thuyền xuống hồ. Gió đang thổi từ hướng tây bắc với tốc độ khoảng 19km/giờ, tạo thành những gợn sóng nhỏ trên mặt hồ. Không vấn đề gì, tôi đã sẵn sàng.

Tôi trèo lên ván, đứng lên, sau đó nghiêng người để với dây kéo buồm lên (sợi dây được gắn với cột buồm). Bằng cách giật mạnh lên phía trên, tôi có thể kéo buồm ra khỏi mặt nước cho tới khi nó thẳng đứng, gần như vuông góc với ván thuyền.

Khi buồm rời khỏi mặt nước, đoạn dây kéo tự di chuyển về phía kín gió, tránh gió. Điều đó giúp tôi có thể căng buồm lên mà không cần dùng thêm sức, vì thế tôi chỉ cần đứng nguyên tại chỗ, dùng cả hai tay để giữ cột buồm. Vị trí “tự nhiên” này chính là điểm khởi đầu của lướt ván, vì vậy tôi có cảm giác khá dễ chịu. Tính tới thời điểm này, tôi vẫn đang làm rất tốt.

Việc tiếp theo của tôi là phải buộc căng dây giữ buồm: Bỏ một tay ra khỏi cột buồm, chụp lấy sào căng buồm và kéo. Bằng cách này, tôi sẽ tạo thêm lực và bắt đầu di chuyển.

Không còn gì nữa...

## Tự giải cứu

Trong 40 phút ở trên hồ ngày hôm đó, tôi đã ngã theo đủ các tư thế: ngã về đằng trước, đằng sau và ngã sang hai bên. Tôi thấy đã uống đủ nước hồ để thấy buồn nôn và chân, cánh tay tôi, lưng tôi đều đau nhức vì phải căng đi căng lại buồm.

Cuối cùng, tôi cũng quyết định là đã đủ rồi. Tôi lạnh, mệt và kiệt sức. Đã đến lúc về nhà thôi.

Đáng tiếc là tôi không biết cách điều chỉnh cái ván chét tiết này, và mỗi khi tôi cố gắng căng buồm, tôi lại khiến cho nó chìm sâu xuống nước nhiều hơn. Gió đẩy tôi về phía nam, rất xa điểm đầu tiên khi tôi xuống nước.

Chính vào lúc đó, tôi quyết định là đã đến lúc luyện tập kỹ thuật tự cứu lấy mình mà tôi đã đọc được từ một cuốn sách nào đó. Tôi nằm xuống ván thuyền, và xếp buồm sao cho nó nằm dẹt trên đầu tôi, với phần đầu của cánh buồm hướng ngược về phía đuôi thuyền. Sau đó, tôi bắt đầu dùng tay làm mái chèo, từ từ đẩy mình về phía bờ.

Thời gian chậm chạp trôi. Tôi không bị đẩy ra quá xa, nhưng cũng phải mất tới 10 phút chèo cật lực để trở về bờ. Tay không phải là mái chèo hiệu quả, ngay cả trong những trường hợp tốt nhất, mà lúc này lại không phải là trường hợp tốt nhất.

Cuối cùng, khi tôi đặt chân lên bờ, thử thách của tôi vẫn chưa kết thúc. Vì tôi ở cách khá xa chỗ xuống lúc đầu, nên cuộc “ra khơi” đầu tiên của tôi kết thúc bằng một cuộc “đi bộ trong xấu hổ”. Tôi buộc cơ thể mệt mỏi rã rời của mình phải mang các vật dụng trở về xe ô tô, lái về nhà và đổ vật ra.

Đó không phải một khởi đầu thuận lợi.

## Phân tích sự việc sau chấn thương

Sau khi nghỉ ngơi, tôi đã tổng kết lại chuyến đi của mình. Chuyện gì đã xảy ra ở đó? Tôi đã sai ở đâu?

Trước tiên, gió quá mạnh đối với một người vừa mới bắt đầu. Với người nào đó đã có kinh nghiệm, thì chuyện đó có lẽ không thành vấn đề, nhưng với một người lần đầu tiên đặt chân lên ván thuyền như tôi, gió như thế là quá mạnh,

quá nhanh.

Thứ hai, tôi không có kinh nghiệm giữ thăng bằng trên bất cứ loại ván nào. Tôi lướt ván, chưa bao giờ chơi ván trượt, cũng chưa bao giờ làm bất cứ việc gì đòi hỏi phải giữ thăng bằng trên một bề mặt chuyển động. Những cơn sóng do gió tạo ra đã khiến cho ván thuyền đu đưa và điều đó khiến tôi sợ hãi.

Thứ ba, việc thêm lực cho cánh buồm làm thay đổi trọng lực trên ván thuyền. Nếu tôi không thay đổi vị trí của mình theo đúng cách để bù lại, chắc chắn tôi sẽ ngã. Nếu tôi không nhanh chóng bỏ cánh buồm ra khi cảm thấy mất kiểm soát, thì có khả năng tôi sẽ bị lộn nhào. Tôi cần phải học cách thay đổi vị trí theo bất cứ hướng nào để có thể đứng được trên ván thuyền, đặc biệt khi có gió mạnh.

Thứ tư, tôi không có đủ các thiết bị an toàn thích hợp. Rõ ràng là tôi cần có một chiếc mũ bảo hiểm để bảo vệ đầu khỏi cột buồm phòng trường hợp tôi bị ngã. Tôi bị “thần chết” gọi tên một lần, nhưng tôi đã may mắn. Tôi sẽ không mắc sai lầm tương tự một lần nữa: Tôi đã đặt mua ngay một chiếc mũ bảo hiểm rồi.

## Học từ quá khứ

Có vài việc tôi có thể làm để tránh cho lần luyện tập tiếp theo của tôi không giống lần đầu tiên.

Thứ nhất, tôi phải chắc chắn lần luyện tập tiếp theo của mình sẽ diễn ra vào một ngày ít gió hơn. Có thể sẽ không thú vị, nhưng điều đó sẽ giúp tôi luyện tập mà không bị choáng ngợp.

Thứ hai, tôi cần phải tập đứng thăng bằng trên ván bằng cách dùng ván thuyền mà không có buồm gắn kèm, sử dụng mái chèo để di chuyển. Bằng cách loại bỏ biến số của cánh buồm, tôi có thể cảm nhận được thế nào là thăng bằng trên ván, và học nghiêng theo bất cứ hướng nào mà không bị lật úp. Đó không phải là một cuộc thử nghiệm hoàn hảo, vì buồm sẽ làm thay đổi trọng tâm của ván thuyền một khi nó được gắn vào ván thuyền, nhưng như vậy vẫn còn tốt hơn là liên tục ngã.

Thứ ba, tôi có thể dành lần tiếp theo để luyện tập với cánh buồm, học cách cảm nhận sự thăng bằng của cánh buồm. Nếu tôi chú tâm, tôi có thể cảm nhận được hướng di chuyển của cánh buồm, và biết điều đó có ảnh hưởng như thế

nào tới ván thuyền. Tôi có thể biết khi nào nên đẩy trọng lực của mình lại để cân bằng lực của cánh buồm, khi nào dừng lại để tránh bị ngã về phía sau, và khi nào để mặc vì gió đột nhiên quá mạnh khiến cho buồm quá nặng, không xử lý được.

Những ngày tiếp theo trời lặng gió, giúp tôi có cơ hội tách bạch việc luyện tập giữ thăng bằng trên ván. Cuộc thử nghiệm đó đã thành công: Cuối buổi luyện tập, tôi không còn sợ nữa và sau đó, một ngày gió mạnh đã cho tôi cơ hội để luyện tập giữ thăng bằng trên ván thuyền khi nước dập dềnh. Tôi vẫn phải vượt qua tính tò mò muốn nhìn xuống và xem nước chảy dưới chân mình, nhưng chỉ một lúc trải qua cảm giác đó đã giúp tôi tiến bộ vượt bậc trong việc chế ngự nỗi sợ.

Vài ngày sau đó, trời nắng ấm, gió không quá mạnh. Tôi chuẩn bị, hạ thủy và kéo dây lên trên, hướng vào trung tâm. Liệu tôi sẽ có một ngày giống ngày đầu tiên không?

Không hẳn: Ngày hôm đó tôi chỉ ngã có hai lần. Tập giữ thăng bằng trên ván thuyền không đã giúp tôi rất nhiều, và tôi đã có thể tránh không bị ngã khi cảm thấy ván thuyền mất kiểm soát. Tôi cũng kéo buồm nhiều lần vào ngày hôm đó, nhưng như thế cũng là tốt lắm rồi, tốt hơn nhiều so với việc phải uống nước hồ. Tôi tập cách thêm lực cho buồm và đột nhiên tôi di chuyển được.

## Làm thế nào bạn quay được thứ này?

Làm cho ván thuyền di chuyển mà không ngã là một thành công, nhưng nó lại kéo theo một vấn đề căng thẳng mới: Làm thế nào bạn rẽ được?

Tôi đã thử nghiệm với những điều học được từ sách vở và đĩa DVD hướng dẫn: Việc quay ván thuyền dưới chân khi giữ buồm ở vị trí trung tâm rất có ích. Cũng có ích khi bạn nhớ rằng ván thuyền (và thuyền buồm nói chung) không thể di chuyển trực tiếp vào cơn gió.

Có một khu vực gọi là khu vực “chết” hay “không đi được”, là khu vực mở rộng một góc 45 độ sang mỗi phía của hướng gió. Hãy thử di chuyển ở bất cứ vị trí nào trong khu vực đó, bạn sẽ thấy mình “bị khóa”, và bạn sẽ dừng lại, hoặc bắt đầu di chuyển giật lùi. Để di chuyển vào vùng có gió, bạn cần phải hướng ván thuyền ít nhất 45 độ về một bên (trái phải đều được) của hướng gió, giữ buồm bằng dây lèo, sau đó đổi đường dây theo hướng khác sau khi đã di

chuyên được một lúc. Bằng cách trở đi trở lại, bạn có thể chạy theo đường đích dắc tới điểm đích, ngay cả khi ngược gió.

Một điều cũng quan trọng không kém là phải dùng lực để quay/rẽ được ván thuyền. Khi bạn di chuyển, việc thay đổi vị trí của cánh buồm sang trái hoặc sang phải sẽ làm thay đổi trọng tâm của lực, điểm trọng tâm của toàn bộ lực mà gió tạo ra trên cánh buồm. Bằng cách thay đổi vị trí của cánh buồm về phía mũi, đuôi, mạn trái hoặc mạn phải, tương quan giữa những trọng điểm này cũng thay đổi, và ván thuyền sẽ xoay để thẳng bằng lại.

Khi quay trở lại ván thuyền sau khi bị ngã, thường bạn sẽ thấy mình ở một vị trí kỳ cục. Lý tưởng nhất, bạn muốn ván thuyền vuông góc với gió, và buồm chĩa theo hướng gió thổi. Với vị trí này, bạn có thể kéo dây buồm lên trên không quá khó khăn.

Tuy nhiên, nếu buồm chĩa theo hướng ngược chiều gió thổi, bạn có nguy cơ bị sào căng buồm đập vào mặt khi bạn kéo dây buồm lên để buồm đón gió. Tương tự như vậy, bạn có thể thấy ván thuyền của bạn chĩa về hướng gió, tiến thẳng vào khu vực chết.

Để cân bằng, tốt nhất là sử dụng chân để quay cánh buồm sao cho nó vuông góc với ván thuyền, bất chấp hướng gió thổi. Khi ván thuyền và buồm đã vuông góc với nhau, bạn có thể từ từ kéo dây buồm lên, để gió xoay ván thuyền theo đúng vị trí. Một khi buồm đã ở cùng chiều gió thổi, bạn có thể kéo căng hoàn toàn, sau đó trở buồm theo bất cứ hướng nào mà bạn muốn đi.

Sau vài tiếng tập luyện, tôi đã hiểu rõ. Tôi không được nhanh nhạy cho lắm, và những cú xoay của tôi cũng không được đẹp mắt cho lắm, nhưng tôi có thể hạ thủy ván thuyền của mình, lướt ra giữa hồ, và quay trở lại gần như đúng nơi tôi đã xuất phát. So với lần lướt ván đầu tiên của tôi, thì đây là cả một sự tiến bộ lớn.

## Tóm lược phương pháp

Hãy cùng tóm lược lại những điều cốt lõi trong phương pháp mà tôi đã sử dụng để học cách lướt ván:

Tôi đã mua các vật dụng, thiết bị cần thiết: một ván thuyền, một cánh buồm, quần áo lặn, áo phao, mũ bảo hiểm và những thiết bị an toàn khác.

Tôi đã học cách lắp ghép ván thuyền và buồm, cũng như tháo dỡ, bảo dưỡng, vận chuyển và lưu giữ thiết bị một cách an toàn.

Tôi đã học được cách bảo vệ bản thân khỏi những nguy cơ như chết đuối, chấn động và giảm nhiệt.

Tôi đã học được cách hạ thủy ván thuyền từ bờ, kéo buồm lên vị trí trung gian, và thêm lực để buồm bắt đầu di chuyển.

Tôi đã học được cách quay ván thuyền và cách để điều chỉnh buồm di chuyển trong điều kiện gió khác nhau.

Tôi đã học được cách để không bị ngã khỏi ván thuyền, trở lại ván thuyền và kéo dây căng buồm trong trường hợp bị ngã.

## Từ đây tôi sẽ đi đâu tiếp?

Tôi sẽ nhanh chóng học lại những điều cơ bản của kỹ năng này, vì thế tôi đang chuẩn bị cho một thử thách mới: chuyển từ ván thuyền sang xuống máy, nghĩa là tăng tốc một cách chóng mặt. “Kế hoạch” chỉ có thể thực hiện trong điều kiện gió mạnh, và tốc độ tăng cũng đồng nghĩa với nguy cơ va chạm và chấn thương cũng cao hơn, vì thế tôi cần phải thực sự thoải mái với những điều cơ bản trước khi thử.

Tôi cũng sẽ thử sử dụng một cánh buồm lớn hơn. Điều kiện gió nhẹ (và biến động) trên hồ khiến cho việc sử dụng loại buồm lớn hơn trở thành một ý tưởng thú vị, vì thế ngay khi sử dụng thành thạo loại buồm 4,7 mét vuông, tôi sẽ bắt đầu luyện tập với loại buồm 7,5 mét vuông. Với hai loại buồm đó, tôi cần phải thống nhất là sẽ sử dụng buồm loại lớn cho những ngày gió nhẹ và buồm loại nhỏ cho những ngày gió mạnh.

Nói tóm lại, lướt ván là một môn thể thao thú vị. Tôi thích thời gian ở trên mặt nước, đặc biệt là bây giờ, khi tôi không hay ngã nữa. Tôi đang nóng lòng trông chờ mùa lướt ván tiếp theo và sẽ tận dụng hết những ngày trời gió nếu có.

# LỜI BẠT

**Thành tích có liên quan tới hành động. Người thành công, dù nam hay nữ, đều tiếp tục tiến bước. Họ có thể phạm sai lầm, nhưng họ không bỏ cuộc.**

**-CONRAD HILTON, Nhà sáng lập chuỗi khách sạn Hilton**

...

Trong vòng chưa đầy một năm, tôi đã học được năm kỹ năng phức tạp.

Tôi không phải là một thiên tài. Tôi cũng không phải một người có khả năng bẩm sinh. Tôi không bỏ công việc ban ngày của mình. Tôi cũng không bỏ mọi thứ để chuyên tâm vào một thứ gì. Tôi không lơ là với gia đình.

Tôi chỉ dành ra một tiếng mỗi ngày để luyện tập, và tôi luyện tập theo cách thông minh. Những kỹ năng mà ban đầu có vẻ bí hiểm, khó nhằn đều trở nên dễ hiểu theo từng ngày, thậm chí theo giờ. Tất cả những gì bạn cần chỉ là một chút nghiên cứu, và khoảng 20 tiếng luyện tập chuyên tâm, tập trung và kiên trì.

Hơn nữa, việc luyện tập của tôi đã trở thành một thói quen: Những kỹ năng này hiện nay đã trở thành một phần trong cuộc sống thường nhật của tôi. Việc học những kỹ năng nhỏ quan trọng nhất trước tiên khiến bạn có thể dễ dàng tiến bộ. Vào thời điểm bạn đọc những dòng chữ này, tôi sẽ còn tiến bộ hơn nhiều trong mỗi kỹ năng. Tiến bộ tới mức nào phụ thuộc vào mức độ luyện tập của tôi.

Nếu bạn muốn học một kỹ năng mới, bạn cần phải luyện tập. Không còn cách nào khác.

Bạn có thể chuẩn bị. Bạn có thể nghiên cứu. Bạn có thể loại bỏ những cảm dỗ khiến bạn sao nhãng, và thay đổi môi trường của bạn sao cho có thể dễ dàng luyện tập hơn. Bạn có thể tìm được những phương thức thông minh để việc luyện tập hiệu quả, năng suất hơn. Nhưng cuối cùng bạn vẫn phải luyện tập.

Con đường có vẻ dài lại là con đường ngắn nhất. Không có “con đường tắt” mang tên không cần luyện tập. Không luyện tập đồng nghĩa với việc không học



được kỹ năng. Chỉ đơn giản vậy thôi.

Tại sao chúng ta không luyện tập? Đơn giản là vì chúng ta bận và chúng ta sợ. Cách đây rất lâu, Shakespeare từng viết trong vở kịch *Measure for Measure* rằng: “Nghĩ ngờ là kẻ phản bội, khiến chúng ta đánh mất những điều tốt đẹp mà lẽ ra chúng ta đã giành được, chỉ vì sợ phải thử”.

Rào cản chính đối với việc học kỹ năng nhanh không phải là rào cản thể chất hay trí tuệ, mà là rào cản về cảm xúc. Việc gì mới cũng khiến chúng ta không thoải mái, và thật dễ để tốn rất nhiều thời gian, năng lượng cho việc nghĩ về việc luyện tập thay vì luyện tập thực sự.

May mắn là rào cản khó chịu này có thể dễ dàng bị phá vỡ: Học kỹ năng luôn đem lại cảm giác thích thú hơn so với thực tế. Bằng cách dành thời gian để luyện tập, nghiên cứu sớm một chút, chống lại cảm giác khó chịu ban đầu, bạn sẽ nhìn thấy sự tiến bộ rõ rệt trong từ 10 đến 20 tiếng luyện tập đầu tiên. Tất cả những gì bạn cần bỏ ra để gặt hái được thành quả đó là một chút nỗ lực, sự kiên trì và một chút bạo dạn.

Bạn không cần phải chọn học nhiều kỹ năng. Bạn chỉ cần chọn một kỹ năng thôi. Hãy chọn một kỹ năng từ danh sách “việc muốn làm” của bạn và cố gắng học cho tới cùng. Hãy học ngoại ngữ này, chơi nhạc cụ kia, khám phá trò chơi này, thực hiện dự án kia, nấu món ăn đó, tạo ra tác phẩm nghệ thuật này... Thực ra, việc thực hiện dễ hơn nhiều so với bạn tưởng.

Hãy cam kết từ trước là trong vòng một tháng tới, mỗi ngày bạn sẽ dành ra khoảng một tiếng để luyện tập. Một khi bạn đã thực sự bắt đầu luyện tập, bạn sẽ học được nhiều và nhanh hơn so với kỳ vọng của bạn. Hãy chia nhỏ nó ra, hãy phân bổ thời gian, hãy thử những việc mới, và bộ não của bạn sẽ bắt đầu học được những thủ thuật một cách tự động. Đó chính là cách vận hành của não bộ. Khi bạn bị vấp hoặc bối rối, hãy thử một cách tiếp cận mới.

Hãy nhớ: Khi đã bắt đầu, bạn không được phép dừng lại cho tới khi đạt được mức độ thành thạo mục tiêu, hoặc đạt tới mốc 20 tiếng. Hãy “chiến đấu” nếu cần, nhưng đừng dừng lại. Hãy thể hiện sự gan góc của bạn, và tiếp tục tiến về phía trước. Bạn sẽ đến được đích, tất cả những gì bạn cần chỉ là luyện tập.

Suy nghĩ cuối cùng đây: Thời gian duy nhất bạn có thể chọn để luyện tập là *ngay hôm nay*.

Không phải ngày mai. Không phải tuần tới. Không phải tháng tới hay năm

tới. *Ngày hôm nay.*

Khi bạn thức dậy vào buổi sáng, bạn có một lựa chọn. Bạn có thể chọn đầu tư thời gian vào việc học những kỹ năng giúp cho cuộc sống của bạn thành công hơn, đáng sống hơn và thú vị hơn... hoặc bạn có thể lãng phí thời gian của bạn cho việc gì đó.

Bạn sẽ làm gì hôm nay?