

✓ Lời mở đầu

Cuốn sách này dành cho ai?

Nếu bạn muốn học lập trình, bạn đã đến đúng nơi. Python là một trong những ngôn ngữ lập trình tốt nhất cho người mới bắt đầu – và cũng là một trong những kỹ năng được săn đón nhất hiện nay.

Bạn cũng đến đúng thời điểm, vì việc học lập trình bây giờ có lẽ dễ dàng hơn bao giờ hết. Với các trợ lý ảo như ChatGPT, bạn không cần phải học một mình. Xuyên suốt cuốn sách này, tôi sẽ gợi ý các cách bạn có thể sử dụng những công cụ này để tăng tốc quá trình học tập của mình.

Cuốn sách này chủ yếu dành cho những người chưa bao giờ lập trình và những người đã có chút kinh nghiệm với các ngôn ngữ lập trình khác. Nếu bạn đã có kinh nghiệm đáng kể với Python, bạn có thể thấy vài chương đầu khá chậm.

Một trong những thách thức khi học lập trình là bạn phải học hai ngôn ngữ: một là ngôn ngữ lập trình và một là ngôn ngữ chuyên môn mà chúng ta dùng để nói về lập trình. Nếu bạn chỉ học ngôn ngữ lập trình, bạn có thể gặp khó khăn khi cần giải thích thông báo lỗi, đọc tài liệu, trao đổi với người khác hoặc sử dụng các trợ lý ảo. Nếu bạn đã từng lập trình nhưng chưa học ngôn ngữ thứ hai này, tôi hy vọng cuốn sách này sẽ hữu ích với bạn.

Mục tiêu của cuốn sách

Khi viết cuốn sách này, tôi đã cố gắng cẩn thận với cách sử dụng từ ngữ. Tôi định nghĩa từng thuật ngữ ngay khi nó xuất hiện lần đầu tiên. Ngoài ra, ở cuối mỗi chương, có một bảng thuật ngữ để ôn lại các thuật ngữ đã được giới thiệu.

Tôi cũng cố gắng viết ngắn gọn. Càng ít nỗ lực tinh thần cần thiết để đọc cuốn sách, bạn sẽ càng có nhiều khả năng tập trung cho việc lập trình.

Tuy nhiên, bạn không thể học lập trình chỉ bằng cách đọc sách – bạn phải thực hành. Vì lý do đó, cuốn sách này bao gồm các bài tập ở cuối mỗi chương để bạn có thể luyện tập những gì mình đã học.

Nếu bạn đọc kỹ và làm bài tập một cách đều đặn, bạn sẽ tiến bộ. Nhưng tôi cũng phải cảnh báo trước – học lập trình không dễ, và ngay cả với những lập trình viên có kinh nghiệm, đôi khi cũng rất nản lòng. Trong quá trình học, tôi sẽ gợi ý các chiến lược giúp bạn viết các chương trình đúng và sửa lỗi các chương trình sai.

Hướng dẫn sử dụng cuốn sách

Mỗi chương trong cuốn sách này được xây dựng dựa trên những chương trước đó, vì vậy bạn nên đọc theo thứ tự và dành thời gian thực hiện các bài tập trước khi tiếp tục.

Sáu chương đầu tiên giới thiệu các yếu tố cơ bản như phép toán, câu lệnh điều kiện và vòng lặp. Chúng cũng giới thiệu khái niệm quan trọng nhất trong lập trình, **hàm**, và một cách mạnh mẽ để sử dụng chúng, **đệ quy**.

Chương 7 và 8 giới thiệu về chuỗi – đại diện cho chữ cái, từ, và câu – cùng các thuật toán để làm việc với chúng.

Từ chương 9 đến 12, bạn sẽ tìm hiểu về các cấu trúc dữ liệu cốt lõi của Python – danh sách, từ điển và bộ giá trị – những công cụ mạnh mẽ để viết các chương trình hiệu quả. Chương 12 trình bày các thuật toán để phân tích văn bản và tạo văn bản mới ngẫu nhiên. Các thuật toán này là nền tảng của các mô hình ngôn ngữ lớn (LLMs), vì vậy chương này sẽ giúp bạn hiểu cách các công cụ như ChatGPT hoạt động.

Chương 13 nói về cách lưu trữ dữ liệu trong bộ nhớ dài hạn – bao gồm tệp và cơ sở dữ liệu. Bài tập trong chương này sẽ hướng dẫn bạn viết một chương trình tìm kiếm trong hệ thống tệp và phát hiện các tệp trùng lặp.

Từ chương 14 đến 17, bạn sẽ được giới thiệu về lập trình hướng đối tượng (OOP) – một cách tổ chức các chương trình và dữ liệu mà chúng làm việc. Nhiều thư viện Python được viết theo phong cách hướng đối tượng, vì vậy các chương này sẽ giúp bạn hiểu thiết kế của chúng – và tự định nghĩa các đối tượng của riêng mình.

Mục tiêu của cuốn sách không phải là bao quát toàn bộ ngôn ngữ Python. Thay vào đó, tôi tập trung vào một tập hợp các khái niệm cung cấp khả năng tối đa với số lượng khái niệm tối thiểu. Tuy nhiên, Python có rất nhiều tính năng bạn có thể sử dụng để giải quyết các vấn đề thông thường một cách hiệu quả. Chương 18 sẽ trình bày một số tính năng đó.

Cuối cùng, chương 19 sẽ chia sẻ những suy nghĩ cuối cùng và gợi ý cho bạn cách tiếp tục hành trình học lập trình.

Có gì mới trong phiên bản thứ ba?

Những thay đổi lớn nhất trong phiên bản này được thúc đẩy bởi hai công nghệ mới – sổ ghi chép Jupyter và trợ lý ảo.

Mỗi chương trong cuốn sách này là một sổ ghi chép Jupyter, tức là một tài liệu chứa cả văn bản

thông thường và mã. Đối với tôi, điều này giúp viết mã, kiểm thử và giữ cho mã luôn đồng bộ với nội dung văn bản dễ dàng hơn. Đối với bạn, nó có nghĩa là bạn có thể chạy mã, sửa đổi nó và thực hiện các bài tập – tất cả đều trong một nơi duy nhất. Hướng dẫn sử dụng sổ ghi chép Jupyter được trình bày trong chương đầu tiên.

Thay đổi lớn thứ hai là tôi đã thêm các lời khuyên về cách làm việc với trợ lý ảo như ChatGPT và sử dụng chúng để tăng tốc quá trình học của bạn. Khi phiên bản trước của cuốn sách được xuất bản vào năm 2016, các công cụ tiền thân của chúng còn kém hữu dụng và hầu hết mọi người chưa biết đến chúng. Giờ đây, chúng đã trở thành công cụ tiêu chuẩn trong kỹ thuật phần mềm, và tôi tin rằng chúng sẽ là một công cụ cách mạng để học lập trình – cũng như học nhiều thứ khác.

Các thay đổi khác trong cuốn sách xuất phát từ những điều tôi cảm thấy tiếc nuối ở phiên bản thứ hai.

Điều đầu tiên là tôi đã không nhấn mạnh tầm quan trọng của kiểm thử phần mềm. Đây là một thiếu sót đáng tiếc từ năm 2016, nhưng với sự xuất hiện của trợ lý ảo, kiểm thử tự động đã trở nên quan trọng hơn bao giờ hết. Do đó, phiên bản này giới thiệu các công cụ kiểm thử phổ biến nhất của Python, bao gồm `doctest` và `unittest`, cùng với một số bài tập để bạn thực hành sử dụng chúng.

Điều tiếc nuối khác của tôi là các bài tập trong phiên bản thứ hai chưa đồng đều – một số bài rất thú vị, nhưng một số lại quá khó. Việc sử dụng sổ ghi chép Jupyter đã giúp tôi phát triển và kiểm thử một chuỗi bài tập hấp dẫn và hiệu quả hơn.

Trong lần chỉnh sửa này, trình tự các chủ đề gần như không thay đổi, nhưng tôi đã sắp xếp lại một vài chương và gộp hai chương ngắn thành một chương duy nhất. Ngoài ra, tôi mở rộng nội dung về chuỗi ký tự để bao gồm cả biểu thức chính quy.

Một số chương sử dụng đồ họa như `turtle`. Trong các phiên bản trước, tôi sử dụng mô-đun `turtle` của Python, nhưng đáng tiếc là nó không hoạt động trên sổ ghi chép Jupyter. Vì vậy, tôi đã thay thế bằng một mô-đun `turtle` mới, dễ sử dụng hơn.

Cuối cùng, tôi đã viết lại một phần đáng kể nội dung, làm rõ những chỗ cần thiết và cắt giảm những phần chưa đủ ngắn gọn.

Tôi rất tự hào về phiên bản mới này – hy vọng bạn sẽ thích nó!

Bắt đầu

Đối với hầu hết các ngôn ngữ lập trình, bao gồm Python, có rất nhiều công cụ bạn có thể sử dụng để viết và chạy các chương trình. Những công cụ này được gọi là môi trường phát triển tích hợp (IDEs).

Nhìn chung, có hai loại IDE:

1. Một số hoạt động với các tệp chứa mã lệnh, cung cấp các công cụ để chỉnh sửa và chạy những tệp này.
2. Một số khác chủ yếu hoạt động với notebooks, là các tài liệu chứa cả văn bản và mã lệnh.

Đối với người mới bắt đầu, tôi khuyên bạn nên bắt đầu với một môi trường phát triển dạng notebook như Jupyter.

The notebooks for this book are available from an online repository at

<https://allendowney.github.io/ThinkPython>.

Có hai cách để sử dụng notebook:

- Bạn có thể tải các notebook về và chạy chúng trên máy tính của mình. Trong trường hợp đó, bạn cần cài đặt Python và Jupyter. Việc này không khó, nhưng nếu bạn muốn học Python, có thể sẽ khá nản lòng nếu phải dành nhiều thời gian để cài đặt phần mềm.
- Một cách khác là chạy notebook trên Colab, một môi trường Jupyter hoạt động trực tiếp trên trình duyệt web, vì vậy bạn không cần cài đặt gì cả. Colab được vận hành bởi Google và hoàn toàn miễn phí.

Nếu bạn mới bắt đầu, tôi thực sự khuyên bạn nên bắt đầu với Colab.

Tài nguyên dành cho giáo viên

Nếu bạn đang giảng dạy với cuốn sách này, dưới đây là một số tài nguyên hữu ích dành cho bạn:

- Bạn có thể tìm thấy các sổ ghi chép chứa lời giải cho các bài tập tại <https://allendowney.github.io/ThinkPython>, cùng với các liên kết đến những tài nguyên bổ sung khác.
- Các câu hỏi cho từng chương và một bài kiểm tra tổng hợp cho toàn bộ cuốn sách sẽ sớm có sẵn tại [COMING SOON].
- Dạy và học với Jupyter là một cuốn sách trực tuyến cung cấp các gợi ý về cách sử dụng Jupyter hiệu quả trong lớp học. Bạn có thể đọc cuốn sách này tại <https://jupyter4edu.github.io/jupyter-edu-book>.
- Một trong những cách tốt nhất để sử dụng sổ ghi chép là viết mã trực tiếp – nơi giảng viên viết mã trực tiếp và học viên theo dõi trên sổ ghi chép của riêng mình. Để tìm hiểu về viết mã trực tiếp – và nhận thêm các lời khuyên hữu ích khác về việc dạy lập trình – tôi khuyến nghị chương trình đào tạo giảng viên của The Carpentries tại <https://carpentries.github.io/instructor-training>.

Lời cảm ơn

Chân thành cảm ơn Jeff Elkner, người đã chuyển thể cuốn sách về Java của tôi sang Python, giúp khởi đầu dự án này và đưa tôi đến với ngôn ngữ lập trình mà tôi yêu thích nhất. Cảm ơn cả Chris Meyers, người đã đóng góp nhiều phần nội dung cho cuốn **Cách tư duy như một Nhà khoa học máy tính**.

Cảm ơn Free Software Foundation vì đã phát triển Giấy phép Tài liệu Tự do GNU, điều này đã giúp tôi có thể hợp tác với Jeff và Chris, và cảm ơn Creative Commons vì giấy phép mà tôi đang sử dụng hiện tại.

Cảm ơn các nhà phát triển và duy trì ngôn ngữ Python cũng như các thư viện mà tôi đã sử dụng, bao gồm mô-đun đồ họa Turtle; các công cụ hỗ trợ tôi phát triển cuốn sách như Jupyter và JupyterBook; và các dịch vụ tôi đã sử dụng, bao gồm ChatGPT, Copilot, Colab, và GitHub.

Cảm ơn các biên tập viên tại Lulu đã làm việc với cuốn Cách tư duy như một Nhà khoa học máy tính và các biên tập viên tại O'Reilly Media đã làm việc với cuốn Tư duy Python.

Đặc biệt cảm ơn các nhà đánh giá kỹ thuật cho phiên bản thứ hai: Melissa Lewis và Luciano Ramalho, và cho phiên bản thứ ba: Sam Lau và Luciano Ramalho (một lần nữa!). Tôi cũng rất biết ơn Luciano vì đã phát triển mô-đun jupyter, một mô-đun đồ họa Turtle được tôi sử dụng trong nhiều chương.

Cảm ơn tất cả các học viên đã làm việc với các phiên bản trước của cuốn sách và tất cả những người đã gửi góp ý và chỉnh sửa. Trong vài năm qua, hơn 100 độc giả tình ý và nhiệt tình đã gửi những gợi ý và chỉnh sửa quý giá. Những đóng góp và sự nhiệt tình của họ là một nguồn trợ giúp to lớn đối với dự án này.

Nếu bạn có góp ý hay chỉnh sửa, vui lòng gửi email đến feedback@thinkpython.com. Nếu bạn trích dẫn ít nhất một phần của câu mà lỗi xuất hiện, điều đó sẽ giúp tôi dễ dàng tìm kiếm hơn. Số trang và số mục cũng hữu ích, nhưng không tiện bằng. Xin cảm ơn!

