Chương 1. Cài đặt, chạy helloworld

Chương 2: biến và các kiểu dữ liệu đơn giản.

Bài 2-1. Simple Message: Gán thông điệp vào một biến và in ra thông điệp đó

Bài 2-2. Simple Message: Gán thông điệp vào một biến và in thông điệp đó ra. Sau đó thay đổi giá trị của biến thành một thông điệp khác và in thông điệp mới đó ra.

Bài 2-3. Personal Message: sử dụng một biến đại diện cho tên của một người và in thông điệp cho người đó. Thông điệp cần đơn giản như “Hello Eric, would you like to learn some Python today?”

Bài 2-4. Name Cases: Sử dụng một biến để biểu diễn tên của một người, sau đó in tên người đó với chữ thường, chữ hoa, và chữ kiểu tiêu đề

Bài 2-5. Famous Quote: Tìm một trích dẫn từ một người nổi tiếng, sau đó in trích dẫn và tên tác giả của trích dẫn đó. Đầu ra như dưới đây (bao gồm cả ký tự trích dẫn):

Albert Einstein once said, “A person who never made a mistake never tried anything new.”

Bài 2-6. Famous Quote2: Lặp lại ví dụ 2-5 nhưng lần này gán tên người nổi tiếng vào biến famous\_person. Sau đó, gán thông điệp của người đó vào biến message, sau đó in ra thông điệp như trong bài tập 2-5

Bài 2-7. Stripping Names: Sử dụng một biến để đại diện cho tên một người bao gồm một số ký tự trắng tại phần đầu và cuối của tên. Đảm bảo rằng mỗi tổ hợp ký tự “\t” và “\n” được sử dụng ít nhất một lần.

In ra tên một lần để thấy các ký tự trắng quanh tên được hiển thị. Sau đó in tên có sử dụng các phương thức lstrip(), rstrip(), strip().

Bài 2-8. Number Eight: Viết các phép tính cộng, trừ, nhân, chia các phép toán mà mỗi kết quả cho ra là số 8. Đầu ra có bốn dòng với số 8 xuất hiện một lần trên mỗi dòng.

Bài 2-9. Favorite Number: Sử dụng một biến để đại diện cho một con số yêu thích. Sau đó, sử dụng biến đó, tạo một thông điệp tiết lộ số yêu thích đó. In ra thông điệp đó.

Bài 2-10. Adding Comments: Chọn 2 chương trình đã viết trong các phần trước và thêm ít nhất mỗi ghi chú cho chương trình đó. Nếu không có gì đặc biệt để ghi chú thì ghi tên, và ngày viết chương trình vào đầu mỗi tệp chương trình. Sau đó ghi chú mô tả chương trình thực hiện vấn đề gì.

Bài 2.11. Zen of Python: gõ lệnh import this ở cửa sổ terminal và nhìn kết quả hiển thị để thấy được các nguyên lý của Python.

Chương 3. Danh sách

3-1. Names: lưu trữ tên của một vài người bạn trong danh sách là names. In ra tên mỗi cá nhân bằng cách truy cập mỗi phần tử trong danh sách, mỗi phần tử một lần.

3-2. Greetings: Khởi đầu giống bài tập 3-1 nhưng thay vì in ra mỗi tên của người bạn đó thì in ra thông điệp tới họ. Dạng thông điệp có thể giống nhau nhưng mỗi thông điệp dành cho mỗi cá nhân riêng biệt.

3-3. Your Own List: tạo một danh sách các phương tiện giao thông ưa thích và lưu vào trong biến danh sách. In ra một loạt câu về các phương tiện trong danh sách đó. Ví dụ: “I would like to own a Honda motorcycle.”

3-4. Guest List: Lập danh sách bao gồm ít nhất ba người mà bạn muốn mời đi ăn tối. Sau đó, sử dụng danh sách để in một thông điệp cho từng người, mời họ đi ăn tối.

3-5. Changing Guest List: Ta vừa nghe nói rằng một trong những khách của bạn không thể làm bữa tối, vì vậy bạn cần gửi một bộ lời mời mới. Bạn sẽ phải nghĩ đến người khác để mời.

- Bắt đầu với bài tập 3-4. Thêm lệnh print() vào cuối chương trình ghi rõ tên của khách không thể tham gia.

- Thay đổi danh sách, thay thế tên của những khách không thể tham gia với tên của những người mới.

- In ra tập thứ hai các thông điệp mời, mỗi cái dành cho một người còn lại trong danh sách.

3-6. More Guests: Bạn vừa tìm thấy một bàn ăn tối lớn hơn, vì vậy hiện có nhiều không gian hơn. Hãy nghĩ đến việc mời thêm ba vị khách đến ăn tối.

- Bắt đầu với các dữ kiện trong bài tập 3-4 và 3-5. Thêm lệnh print() vào cuối chương trình thông báo cho mọi người là đã tìm được chiếc bàn ăn lớn hơn.

- Sử dụng insert() để thêm một người khách mới vào đầu danh sách

- Sử dụng insert() để thêm một người khách mới vào giữa danh sách

- Sử dụng append() để thêm một người khách mới vào cuối danh sách

- In ra bộ thông điệp mời mới, mỗi cái dành cho một khách có trong danh sách.

3-7. Shrinking Guest List: bạn vừa phát hiện ra rằng bàn ăn tối mới của bạn sẽ không đến kịp giờ ăn tối, và bạn chỉ có chỗ cho hai khách.

- Bắt đầu với chương trình trong bài 3-6. Thêm một dòng in ra thông điệp thông báo bạn chỉ có thể mời 2 khách.

- Sử dụng pop() để xóa khách từ danh sách mỗi lần một người cho đến khi chỉ còn hai cái tên trong danh sách. Mỗi khi xóa một tên trong danh sách, in ra thông điệp xin lỗi không thể mời khách đó tới ăn tối.

- In tin nhắn cho từng người trong số hai người vẫn còn trong danh sách của bạn, cho phép họ biết họ vẫn được mời

- Sử dụng del để xóa hai cái tên cuối cùng trong danh sách, kết quả là danh sách rỗng. In ra danh sách để đảm bảo bạn đang có một danh sách rỗng ở cuối chương trình.

3-8. Seeing the World: Nghĩ về ít nhất 5 địa điểm bạn đã từng đi qua

 Lưu trữ các địa điểm trong một danh sách. Đảm bảo danh sách không theo thứ tự bảng chữ cái.

 In danh sách địa điểm theo thứ tự ban đầu

 In danh sách địa điểm theo thứ tự bảng chữ cái được sắp xếp dùng phương thức sorted()

 In lại danh sách để thấy danh sách địa điểm vẫn giữ nguyên

 Sử dụng phương thức sorted() để in danh sách theo thứ tự ngược của bảng chữ cái

 In ra để thấy danh sách vẫn được giữ nguyên

 Sử dụng phương thức reverse() để thay đổi thứ tự của danh sách ban đầu. In ra kết quả

 Sử dụng phương thức reverse() để đổi về danh sách ban đầu

 Sử dụng phương thức sort() để sắp xếp danh sách theo thứ tự bảng chữ cái. In ra danh sách để thấy thứ tự của nó đã thay đổi

 Sử dụng phương thức sort() để sắp xếp danh sách ngược bảng chữ cái. In ra để thấy thứ tự của danh sách đã thay đổi.

3-9. Locations: Sử dụng danh sách trong ví dụ 3-8, dùng phương thức len() để in ra số lượng địa điểm trong danh sách.

3-10. Every Function: Khi tất cả mọi thứ có thể lưu trong một danh sách, nghĩ về một loại dữ liệu nào đó có thể lưu trong danh sách như những ngọn núi, những dòng sông, những quốc gia, những thành phố…Viết một chương trình tạo ra danh sách chứa dữ liệu trên và dùng các phương thức kể trên mỗi phương thức ít nhất một lần.

3-11. Pizzas: Hãy nghĩ về ít nhất ba loại bánh pizza yêu thích của bạn. Lưu các tên pizza này vào danh sách, sau đó sử dụng vòng lặp for để in tên của từng pizza.

- Thay đổi vòng lặp và in ra cả câu thay vì in ra chỉ mỗi tên của loại pizza ưa thích. Ví dụ, đối với mỗi loại pizza có một câu in ra như: I like pepperoni pizza.

- Thêm một dòng vào cuối chương trình, bên ngoài vòng lặp for, cho biết bạn thích pizza đến mức nào. Đầu ra phải bao gồm ba dòng trở lên về loại bánh pizza bạn thích và sau đó là một câu bổ sung, chẳng hạn như: I really love pizza!

3-12. Animals: Hãy nghĩ về ít nhất ba loài động vật khác nhau có đặc điểm chung. Lưu tên của những con vật này trong một danh sách, sau đó sử dụng vòng lặp for để in ra tên của mỗi con vật.

- Sửa đổi chương trình để in cả một câu về từng loài động vật, chẳng hạn như: A dog would make a great pet.

- Thêm một dòng vào cuối chương trình nêu rõ những điểm chung của những con vật này. Có thể in một câu chẳng hạn như: Any of these animals would make a great pet!

3-13. Counting to Twenty: Sử dụng vòng lặp for để in các số từ 1 đến 20

3-14. One Million: Lập danh sách các số từ một đến một triệu, sau đó sử dụng vòng lặp for để in các số. (Nếu đầu ra mất quá nhiều thời gian, hãy dừng quá trình đó bằng cách nhấn ctrl-C hoặc bằng cách đóng cửa sổ đầu ra.)

3-15. Summing a Million: Lập danh sách các số từ một đến một triệu, sau đó sử dụng min() và max() để đảm bảo danh sách thực sự bắt đầu từ một và kết thúc ở một triệu. Ngoài ra, hãy sử dụng hàm sum() để xem Python có thể cộng một triệu số nhanh như thế nào.

3-16. Odd Numbers: Sử dụng đối số thứ ba của hàm range() để tạo danh sách các số lẻ từ 1 đến 20. Sử dụng vòng lặp for để in từng số.

3-17. Threes: Lập danh sách các bội số của 3 từ 3 đến 30. Sử dụng vòng lặp for để in các số trong danh sách của bạn.

3-18. Cubes: Một số được nâng lên lũy thừa thứ ba được gọi là một lập phương. Ví dụ: lập phương của 2 được viết là 2\*\*3 trong Python. Tạo danh sách 10 số lập phương đầu tiên (tức là lập phương của mỗi số nguyên từ 1 đến 10) và sử dụng vòng lặp for để in ra giá trị của mỗi số lập phương.

3-19. Cube Comprehension: sử dụng list comprehension để thực hiện yêu cầu trong bài 3-18.

3-20. Slices: Sử dụng một trong các chương trình bạn đã viết trong chương này, thêm một số dòng vào cuối chương trình để thực hiện các thao tác sau:

- In ra: the first three items in the list are: . Sau đó sử dụng slice để in ra ba giá trị đầu tiên trong danh sách.

- In ra thông điệp: Three items from the middle. Sau đó sử dụng slice để in ra ba giá trị giữa của danh sách.

- In ra thông điệp: The last three items in the list are: sau đó sử dụng slice để in ra ba giá trị cuối cùng của danh sách.

3-21. My Pizzas, Your Pizzas: Bắt đầu như bài tập 3-11. Tạo một danh sách sao chép danh sách pizzas và gọi nó là friend\_pizzas. Sau đó thực hiện:

- Thêm một pizza mới vào danh sách ban đầu

- Thêm một pizza khác vào danh sách friend\_pizza

- Đảm bảo rằng ta có hai danh sách riêng rẽ. In thông điệp: My favorite pizzas are: sau đó dùng vòng lặp for đối với danh sách pizza ban đầu để in ra. In tiếp thông điệp My friend’s favorite pizzas are: sau đó dùng vòng lặp for để in ra danh sách thứ hai. Cần đảm bảo mỗi pizza mới được thêm vào danh sách tương ứng.

3-22. More Loops: Tất cả các phiên bản của Foods.py trong phần này đã tránh sử dụng vòng lặp for khi in để tiết kiệm dung lượng. Chọn một phiên bản của Foods.py và viết hai vòng lặp for để in từng danh sách thực phẩm.

3-23. Buffet: Một cửa hàng buffet phục vụ chỉ 5 loại đồ ăn. Thêm 5 loại đồ ăn đơn giản mà cửa hàng phục vụ lưu vào trong một tuple.

- Sử dụng vòng lặp for để in ra các món ăn của cửa hàng

- Cố gắng sửa đổi một trong các mục và đảm bảo rằng Python từ chối việc biến đổi.

- Nhà hàng thay đổi thực đơn của mình, thay thế hai trong số các món bằng các loại thực phẩm khác nhau. Thêm một dòng viết lại bộ tuple, sau đó sử dụng vòng lặp for để in từng mục trên menu đã sửa đổi.

Chương 4. Câu lệnh rẽ nhánh

4-1. Conditional Tests: Viết một loạt các bài kiểm tra điều kiện. In ra câu mô tả mỗi bài kiểm tra và dự đoán của bạn cho kết quả của mỗi bài kiểm tra. Mã nguồn sẽ trông giống như sau:

car = 'subaru'

print("Is car == 'subaru'? I predict True.")

print(car == 'subaru')

print("\nIs car == 'audi'? I predict False.")

print(car == 'audi')

- Xem xét kỹ kết quả của bạn và đảm bảo rằng hiểu lý do tại sao mỗi dòng đánh giá True hoặc False.

- Tạo ra ít nhất 10 kiểm tra có điều kiện và có ít nhất 5 cái là True, ít nhất 5 cái là False.

4-2. More Conditional Tests: tạo tệp conditional\_tests.py và thêm vào các kiểm tra có điều kiện:

- Kiểm tra đẳng thức và bất đẳng thức với chuỗi (string).

- Kiểm tra sử dụng phương thức lower().

- Kiểm tra có điều kiện với số trong các trường hợp: đẳng thức, bất đẳng thức, lớn hơn, nhỏ hơn, lớn hơn hoặc bằng, nhỏ hơn hoặc bằng.

- Kiểm tra có điều kiện sử dụng từ khóa and hoặc or.

- Kiểm tra xem một phần tử có trong danh sách không.

- Kiểm tra xem một phần tử không có trong danh sách.

4-3. Alien Colors #1: Tưởng tượng con quái vật bị bắn trong một trò chơi. Tạo ra một biến gọi là alien\_color và gán vào các giá trị 'green', 'yellow', or 'red'.

- Viết câu lệnh if kiểm tra mầu sắc của quái vật. Nếu như bắn hạ quái vật mầu xanh thì in ra thông điệp người chơi kiếm được 5 điểm.

- Viết một phiên bản của chương trình này vượt qua kiểm tra if và một phiên bản khác không thành công. (Phiên bản không thành công sẽ không có đầu ra.)

4-4. Alien Colors #2: Chọn một mầu cho quái vật trong ví dụ 5-3 và viết chuỗi if-else:

- Nếu bắn được quái vật mầu xanh, người chơi kiếm được 5 điểm

- Nếu bắn được quái vật mầu khác, người chơi kiếm được 10 điểm.

- Viết một phiên bản của chương trình này chạy khối if và một phiên bản khác chạy khối else.

4-5. Alien Colors #3: thay chuỗi if-else trong ví dụ 5-4 thành chuỗi if-elif-else:

- Nếu quái vật mầu xanh, in ra thông báo người chơi kiếm được 5 điểm

- Nếu quái vật mầu vàng, in ra thông báo người chơi kiếm được 10 điểm

- Nếu quái vật mầu đỏ, in ra thông báo người chơi kiếm được 15 điểm

- Viết ba phiên bản của chương trình này, đảm bảo mỗi thông báo được in cho quái vật có màu sắc thích hợp.

4-6. Stages of Life: Viết chuỗi if-elif-else xác định trạng thái cuộc đời của mỗi con người. Thiết lập một giá trị cho biến age sau đó:

- Nếu tuổi nhỏ hơn 2, gọi là baby

- Nếu tuổi từ 2 đến 4 gọi là toddler

- Nếu tuổi từ 4 đến 13 gọi là kid

- Nếu tuổi từ 13 đến 20 gọi là teenager

- Nếu tuổi từ 20 đến 65 gọi là adult

- Nếu tuổi 65 trở lên gọi là elder

4-7. Favorite Fruit: Lập danh sách các loại trái cây yêu thích, sau đó viết một loạt các câu lệnh if độc lập để kiểm tra các loại trái cây nhất định trong danh sách.

- Tạo danh sách với ba loại trái cây gọi là favorite\_fruits

- Viết 5 câu lệnh if, mỗi câu lệnh kiểm tra xem một loại trái cây nào đó có trong danh sách không. Nếu có thì in ra thông báo như: You really like bananas!

4-8. Hello Admin: Lập danh sách gồm nhiều tên người dùng, bao gồm tên 'quản trị viên'. Hãy tưởng tượng bạn đang viết mã sẽ in lời chào cho mỗi người dùng sau khi họ đăng nhập vào một trang web. Lặp lại danh sách và in lời chào cho mỗi người dùng:

- Nếu tên người dùng là 'quản trị viên', hãy in lời chào đặc biệt, chẳng hạn như Xin chào quản trị viên, bạn có muốn xem báo cáo trạng thái không?

- Nếu không, hãy in một lời chào chung chung, chẳng hạn như Xin chào Jaden, cảm ơn vì đăng nhập lại.

4-9. No Users: Thêm kiểm tra if vào hello\_admin.py để đảm bảo danh sách người dùng là không rỗng.

- Nếu danh sách là rỗng, in ra thông báo: We need to find some users!

- Xóa tất cả tên người dùng khỏi danh sách và đảm bảo rằng thông báo chính xác được in.

4-10. Checking Usernames: Thực hiện như sau để tạo một chương trình mô phỏng cách các trang web đảm bảo rằng mọi người đều có một tên người dùng duy nhất.

- Tạo danh sách gồm nhiều hoặc nhiều tên người dùng được gọi là current\_users.

- Tạo một danh sách tên người dùng khác được gọi là new\_users. Đảm bảo một hoặc hai tên người dùng mới cũng có trong danh sách current\_users.

- Lặp danh sách new\_users để xem từng tên người dùng mới đã được sử dụng chưa. Nếu có, hãy in thông báo rằng người đó sẽ cần nhập tên người dùng mới. Nếu tên người dùng chưa được sử dụng, hãy in thông báo rằng tên người dùng đó khả dụng.

- Đảm bảo rằng so sánh của bạn không phân biệt chữ hoa chữ thường. Nếu 'John' đã được sử dụng, thì 'JOHN' sẽ không được chấp nhận. (Để làm điều này, bạn cần tạo một bản sao của current\_users chứa các phiên bản viết thường của tất cả người dùng hiện có.)

4-11. Ordinal Numbers: Các số thứ tự cho biết vị trí của chúng trong danh sách, chẳng hạn như như 1 hoặc 2. Hầu hết các số thứ tự kết thúc bằng thứ, ngoại trừ 1, 2 và 3.

- Lưu trữ số từ 1 đến 9 trong danh sách

- Lặp trong danh sách

- Sử dụng chuỗi if-elif-else bên trong vòng lặp để in phần cuối theo thứ tự thích hợp cho mỗi số. Kết quả đầu ra của bạn phải là " 1st 2nd 3rd 4th 5th 6th 7th 8th 9th", và mỗi kết quả phải nằm trên một dòng riêng biệt.

Chương 5. Từ điển

5-1. Person: Sử dụng từ điển để lưu trữ thông tin về một người mà ta biết. Lưu trữ họ, tên, tuổi và thành phố nơi họ sinh sống. Ta nên có các khóa như first\_name, last\_name, age và city. In từng phần thông tin được lưu trữ trong từ điển đó.

5-2. Favorite Numbers: Sử dụng từ điển để lưu trữ các số yêu thích của mọi người. Hãy nghĩ về những cái tên khác và sử dụng chúng làm khóa trong từ điển của bạn. Hãy nghĩ về một con số yêu thích của mỗi người và lưu trữ mỗi con số như một giá trị trong từ điển của bạn. In tên của từng người và số yêu thích của họ. Để thú vị hơn nữa, hãy thăm dò ý kiến một vài người bạn và nhận một số dữ liệu thực tế cho chương trình của bạn.

5-3. Glossary: Một từ điển Python có thể được sử dụng để mô hình hóa một từ điển thực tế. Tuy nhiên, để tránh nhầm lẫn, hãy gọi nó là Glossary.

- Hãy nghĩ về các từ lập trình mà bạn đã học trong các chương trước. Sử dụng những từ này làm chìa khóa trong bảng thuật ngữ của bạn và lưu trữ ý nghĩa của chúng dưới dạng giá trị.

- In từng từ và nghĩa của nó dưới dạng đầu ra được định dạng gọn gàng. Bạn có thể in từ theo sau bởi dấu hai chấm và sau đó là nghĩa của từ đó hoặc in từ trên một dòng rồi in nghĩa của nó được thụt lề trên dòng thứ hai.

5-4. Glossary 2: Như bài 5-3 nhưng bằng cách thay thế chuỗi lệnh gọi print() của bạn bằng một vòng lặp chạy qua các khóa và giá trị của từ điển. Khi nào bạn chắc chắn rằng vòng lặp của bạn hoạt động, hãy thêm nhiều thuật ngữ Python hơn vào bảng thuật ngữ của bạn. Khi bạn chạy lại chương trình của mình, những từ và nghĩa mới này sẽ tự động được đưa vào đầu ra.

5-5. Rivers: Tạo một từ điển có chứa ba con sông lớn và quốc gia mà mỗi con sông chạy qua. Một cặp khóa-giá trị có thể là 'nile': 'egypt'.

- Sử dụng vòng lặp để in ra câu về con sông chảy qua quốc gia. Ví dụ The Nile runs through Egypt.

- Sử dụng một vòng lặp để in tên của mỗi con sông có trong từ điển.

- Sử dụng một vòng lặp để in tên của mỗi quốc gia có trong từ điển.

5-6. Polling: Sử dụng mã trong favourite\_languages.py.

- Lập danh sách những người nên tham gia cuộc thăm dò về ngôn ngữ yêu thích. Bao gồm một số tên đã có trong từ điển và một số tên chưa có.

- Lặp lại danh sách những người nên tham gia cuộc thăm dò. Nếu họ đã thực hiện cuộc thăm dò ý kiến, hãy in tin nhắn cảm ơn họ đã trả lời. Nếu họ chưa tham gia cuộc thăm dò ý kiến, hãy in thông báo mời họ tham gia các cuộc thăm dò.

5-7. People: Bắt đầu với phần chương trình đã làm trong bài 5-1. Tạo hai từ điển mới đại diện cho những người khác nhau và lưu trữ cả ba từ điển trong một danh sách được gọi là people. Lặp qua danh sách people này và in ra tất các thông tin biết về mỗi người.

5-8. Pets: Tạo một số từ điển, trong đó mỗi từ điển đại diện cho một con vật cưng khác nhau. Trong mỗi từ điển, hãy bao gồm loại động vật và tên của chủ sở hữu. Lưu các từ điển này trong danh sách gọi là pets. Lặp qua danh sách pets và in ra tất cả các thông tin về thú cưng đã lưu.

5-9. Favorite Places: Tạo một từ điển có tên là favorite\_places. Hãy nghĩ ra ba cái tên để sử dụng làm chìa khóa trong từ điển và lưu trữ một đến ba địa điểm yêu thích cho mỗi người. Để làm cho bài tập này thú vị hơn một chút, hãy nhờ một số bạn bè kể tên một vài địa điểm yêu thích của họ. Lặp lại từ điển và in tên của từng người và địa điểm yêu thích của họ.

5-10. Favorite Numbers: Thay đổi chương trình từ bài 5-2 sao cho mỗi người có thể thích nhiều hơn một số. Sau đó, in tên của từng người cùng với các số yêu thích của họ.

5-11. Cities: Tạo một từ điển gọi là cities. Sử dụng tên của ba thành phố làm chìa khóa trong từ điển của bạn. Tạo từ điển thông tin về từng thành phố và bao gồm quốc gia có thành phố đó, dân số ước tính của nó và một số thông tin thực tế về thành phố đó. Các khóa cho từ điển của mỗi thành phố phải là một cái gì đó như quốc gia, dân số và thông tin thực tế. In tên của mỗi thành phố và tất cả thông tin bạn đã lưu trữ về thành phố đó.

5-12. Extensions: Chúng ta hiện đang làm việc với các ví dụ đủ phức tạp để có thể mở rộng chúng theo bất kỳ cách nào. Sử dụng một trong các chương trình ví dụ từ chương này và mở rộng nó bằng cách thêm các khóa và giá trị mới, thay đổi ngữ cảnh của chương trình hoặc cải thiện định dạng của đầu ra.

Chương 6. Input() và vòng lặp while

6-1. Rental Car: Viết một chương trình hỏi người dùng loại xe nào họ muốn thuê, sau đó viết ra một thông điệp về loại xe đó. Ví dụ: “Let me see if I can fnd you a Subaru.”

6-2. Restaurant Seating: viết một chương trình hỏi người dùng muốn đặt bàn có bao nhiêu chỗ. Nếu câu trả lời lớn hơn 8, in ra thông điệp là họ cần chờ để sắp xếp bàn. Nếu không, in ra thông điệp là bàn họ đặt đã sẵn sàng.

6-3. Multiples of Ten: viết chương trình yêu cầu nhập vào một số, sau đó in ra câu trả lời là số đó có phải là bội của 10 hay không.

6-4. Pizza Toppings: viết một chương trình có vòng lặp yêu cầu người dùng nhập vào tên các lớp phủ bánh pizza (toppings) cho tới khi người dùng gõ vào giá trị ‘quit’. Với mỗi giá trị lớp phủ bánh người dùng đã thêm, in ra một thông điệp rằng ta đã thêm lớp phủ đó vào bánh cho họ.

6-5. Movie Tickets: một rạp chiếu phim bán vé có giá thay đổi dựa trên tuổi của khách hàng. Nếu khách hàng dưới 3 tuổi – miễn phí vé; Nếu khách hàng từ 3 tới 12 tuổi – giá vé là 10$; Nếu khách hàng lớn hơn 12 tuổi – giá vé là 15$. Viết chương trình yêu cầu khách hàng nhập vào tuổi, sau đó in ra giá vé cho người dùng.

6-6. Three Exits: làm các ví dụ 6-4, 6-5 với các cách sau, mỗi cái ít nhất 1 lần:

- sử dụng kiểm tra có điều kiện trong câu lệnh while để dừng vòng lặp

- sử dụng một biến active để điều khiển số lượt chạy của vòng lặp

- sử dụng break để thoát vòng lặp khi người dùng gõ giá trị ‘quit’

6-7. Infinity: viết một vòng lặp không kết thúc và chạy nó. (Để kết thúc vòng lặp nhấn tổ hợp ctrl-C hoặc đóng cửa sổ đang hiển thị đầu ra).

6-8. Deli: Tạo một danh sách có tên là sandwich\_orders và điền vào đó tên của các loại bánh sandwich khác nhau. Sau đó, tạo một danh sách trống được gọi là finish\_sandwiches. Lặp lại danh sách các đơn đặt hàng bánh sandwich và in thông báo cho mỗi đơn đặt hàng, chẳng hạn như tôi đã làm bánh mì cá ngừ cho bạn. Khi mỗi chiếc bánh sandwich được làm xong, hãy chuyển nó vào danh sách finish\_sandwiches. Sau khi tất cả các loại bánh mì đã được làm xong, hãy in một thông báo liệt kê từng loại bánh sandwich đã được làm.

6-9. No Pastrami: Giả sử trong danh sách bài 6-8 có 3 cái thuộc loại pastrami. Tại phần đầu của mã, in ra là cửa hàng hết pastrami và sử dụng vòng lặp while để loại bỏ tất cả các bánh sandwich loại pastrami trong sandwich\_orders. In ra danh sách sandwich\_orders để đảm bảo đã loại bỏ hết các sandwich loại pastrami.

6-10 Dream Vacation: Viết một chương trình thăm dò ý kiến người dùng về kỳ nghỉ mơ ước của họ. Viết lời nhắc tương tự như Nếu bạn có thể đến thăm một nơi trên thế giới, bạn sẽ đi đâu? Bao gồm một khối mã in kết quả của cuộc thăm dò ý kiến.

Chương 7. Hàm

7-1. Message: Viết một hàm có tên display\_message() in ra một câu nói cho mọi người biết bạn đang học gì trong chương này. Gọi và đảm bảo rằng thông báo hiển thị chính xác.

7-2. Favorite Book: Viết một hàm được gọi là favourite\_book() chấp nhận một tham số, tiêu đề. Hàm sẽ in một tin nhắn, chẳng hạn như Một trong những cuốn sách yêu thích là Alice in Wonderland. Gọi hàm, đảm bảo bao gồm tên sách làm đối số trong lệnh gọi hàm

7-3. T-Shirt: Viết một hàm có tên make\_shirt() chấp nhận kích thước và nội dung của thông điệp sẽ được in trên áo. Hàm cần in câu tóm tắt size áo và thông điệp in trên đó. Gọi hàm lần thứ nhất bằng cách sử dụng các đối số vị trí để tạo một cái áo. Gọi hàm lần thứ hai bằng cách sử dụng các đối số từ khóa.

7-4. Large Shirts: Sửa đổi hàm make\_shirt () để áo sơ mi có kích thước lớn theo mặc định với thông điệp có nội dung I love Python. Tạo một chiếc áo sơ mi lớn và một chiếc áo sơ mi vừa với thông điệp mặc định và một chiếc áo sơ mi có kích thước bất kỳ với thông điệp khác.

7-5. Cities: Viết một hàm được gọi là description\_city () chấp nhận tên của một thành phố và đất nước của thành phố. Hàm sẽ in ra một câu kiểu như: Reykjavik is in Iceland. Cho tham số country giá trị mặc định. Gọi hàm với ba thành phố khác nhau và ít nhất một lời gọi hàm không sử dụng giá trị mặc định.

7-6. City Names: Viết một hàm có tên city\_country () lấy tên của một thành phố và quốc gia của nó. Hàm sẽ trả về một chuỗi có định dạng như sau:

"Santiago, Chile"

Gọi hàm của bạn với ít nhất ba cặp thành phố-quốc gia và in các giá trị được trả về.

7-7. Album: Viết một hàm có tên make\_album () để xây dựng một từ điển mô tả một album nhạc. Hàm sẽ nhận tên nghệ sĩ và tiêu đề album và nó sẽ trả về một từ điển chứa hai phần thông tin này. Sử dụng chức năng này để tạo ba từ điển đại diện cho các album khác nhau. In từng giá trị trả về để cho thấy rằng từ điển đang lưu trữ thông tin album một cách chính xác. Sử dụng None có để thêm một tham số tùy chọn vào make\_album() cho phép lưu trữ số lượng bài hát trong một album. Nếu gọi hàm bao gồm một giá trị cho số lượng bài hát, hãy thêm giá trị đó vào từ điển của album. Thực hiện ít nhất một lệnh gọi chức năng mới bao gồm số lượng bài hát trong album

7-8. User Albums: Bắt đầu với chương trình tại bài 7-7. Xây dựng vòng lặp while cho phép nhập tác giả và tiêu đề của album nhạc. Sau khi bạn có thông tin đó, hãy gọi make\_album() với thông tin nhập của người dùng và in từ điển đã tạo. Đảm bảo bao gồm một giá trị thoát trong vòng lặp while.

7-9. Messages: Tạo danh sách chứa một loạt tin nhắn văn bản ngắn. Chuyển danh sách đến một hàm có tên show\_messages(), hàm này sẽ in từng tin nhắn văn bản.

7-10. Sending Messages: Bắt đầu với chương trình trong bài 7-9. Viết một hàm tên là send\_message() in ra thông điệp chuỗi và chuyển các thông điệp đã in ra vào danh sách sent\_messages. Sau khi gọi hàm, hãy in cả hai danh sách của bạn để đảm bảo rằng các tin nhắn đã di chuyển một cách chính xác.

7-11. Archived Messages: Bắt đầu với chương trình trong bài 7-10. Gọi hàm send\_messages() với một bản sao của danh sách messages. Sau khi gọi hàm, hãy in cả hai danh sách của bạn để cho thấy rằng danh sách ban đầu vẫn giữ nguyên tin nhắn.

7-12. Sandwiches: Viết một hàm chấp nhận danh sách các mục mà một người muốn có trên bánh mì sandwich. Hàm phải có một tham số thu thập nhiều mục như lời gọi hàm cung cấp và nó sẽ in ra một bản tóm tắt về bánh sandwich đang được đặt hàng. Gọi hàm ba lần, mỗi lần sử dụng một số đối số khác nhau.

7-13. User Profile: Bắt đầu với bản sao của user\_profle.py trong bài học. Xây dựng hồ sơ của chính bạn bằng cách gọi build\_profile(), sử dụng họ và tên của bạn và ba cặp khóa-giá trị khác mô tả bạn.

7-14. Cars: Viết một hàm lưu thông tin về ô tô trong từ điển. Hàm phải luôn nhận được nhà sản xuất và tên kiểu máy. Sau đó, nó sẽ chấp nhận một số lượng đối số từ khóa tùy ý. Gọi hàm với thông tin bắt buộc và hai cặp tên-giá trị khác, chẳng hạn như màu hoặc tính năng tùy chọn. Hàm của bạn sẽ hoạt động cho một lời gọi như sau:

car = make\_car('subaru', 'outback', color='blue', tow\_package=True)

In từ điển được trả lại để đảm bảo tất cả thông tin đều được lưu trữ một cách chính xác.

7-15. Printing Models: Đặt các hàm cho ví dụ print\_models.py trong một tệp riêng có tên là print\_functions.py. Viết câu lệnh nhập ở đầu print\_models.py và sửa đổi tệp để sử dụng các hàm đã nhập.

7-16. Imports: Sử dụng một chương trình bạn đã viết có một chức năng trong đó, hãy lưu trữ chức năng đó trong một tệp riêng biệt. Nhập hàm vào tệp chương trình chính của bạn và gọi hàm bằng cách sử dụng từng phương pháp sau:

import module\_name

from module\_name import function\_name

from module\_name import function\_name as fn

import module\_name as mn

from module\_name import \*