

[< Previous](#)[Next >](#)

Unit 1 / Lesson 3 / Project 2

Pirate Bartender Project

Estimated Time: 2-3 hours

To demonstrate your mastery of function basics in Python, we'd like you to create a new app which specializes in bartending. Pirate bartending. The bartender will invent a new and delicious cocktail for you based upon your answers to some simple questions.

The basic requirements



Create questions and ingredients dictionaries

The bartender should ask questions that determine your tastes and then identify ingredients to suit those tastes. If you like you can use our example bartender below, but feel free to customize!

```
questions = {
    "strong": "Do ye like yer drinks strong?",
    "salty": "Do ye like it with a salty tang?",
    "bitter": "Are ye a lubber who likes it bitter?",
    "sweet": "Would ye like a bit of sweetness with yer poison?",
    "fruity": "Are ye one for a fruity finish?",
}

ingredients = {
    "strong": ["glug of rum", "slug of whisky", "splash of gin"],
    "salty": ["olive on a stick", "salt-dusted rim", "rasher of bacon"],
    "bitter": ["shake of bitters", "splash of tonic", "twist of lemon |
```

```
"sweet": ["sugar cube", "spoonful of honey", "splash of cola"],  
"fruity": ["slice of orange", "dash of cassis", "cherry on top"],  
}
```

Note that there's a comma at the end of each entry - even the last one. That last comma isn't necessary, but it means you can add or reorder entries without having to remember to add in a comma.

Write a function to ask what style of drink a customer likes

The function should ask each of the questions in the `questions` dictionary, and gather the responses in a new dictionary. The new dictionary should contain the type of ingredient (for example `"salty"` , or `"sweet"`), mapped to a Boolean value. If the customer answers `y` or `yes` to the question then the value should be `True` , otherwise the value should be `False` . The function should return the new dictionary.

Remember that you can use the `input` function to get an answer from the customer. If you can't remember how this works then take a look back over the instructions for the FizzBuzz project.

Write a function to construct a drink

The function should take the preferences dictionary created in the first function as a parameter. Inside the function you should create an empty list to represent the drink. For each type of ingredient which the customer said they liked you should append a corresponding ingredient from the `ingredients` dictionary to the drink. Finally the function should return the drink.

To choose an ingredient from one of the ingredient lists you can use the `random.choice` function. This selects a random item from a list, for example:

```
import random

beatles = ["John", "Paul", "George", "Ringo"]
# Print the name of a random Beatle
print(random.choice(beatles))
```

Provide a main function

Use `if __name__ == '__main__':` to run this function from the command line. The `main` function should call your two functions in order, passing your list of preferences to the drink creation function. It should then print out the contents of the drink.

Use version control

You should create a new git repo for this assignment. Be sure to commit your file when you've completed this project and push it up to Github.

Get feedback from your Mentor

Share a link to the project with your mentor and be sure to get feedback at your next mentor session.

Extra challenges

If you found completing the basic requirements fairly straightforward then you should try to extend your app to add the following features:

Give the cocktails a name

All good cocktails should have a memorable name. Try to write a function which will name your cocktails. The name should be a random combination of an adjective and a noun (for example your bartender could make a "Fluffy Chinchilla", a "Salty Sea-Dog", or a "Fluffy Sea-Dog").

Keep 'em coming

At the moment you can only get one drink at a time from the bartender. A well trained pirate bartender should offer his customer another drink when they've finished their previous one. Try adding a loop in the `main` function which will ask the customer whether they want another drink, and keep creating new recipes as long as they agree.

Extension exercises

If the extra challenges were not a problem and you're running ahead of schedule then you could try to implement one or two of the following features in your app:

- Multiple customers: The bartender could ask for the customer's name before they are served. They could then remember the customer's preferences for when the same customer asks for another drink.
- Stock control: Even pirate bars don't have a limitless supply of ingredients. You could add a stock count for each ingredient which decreases whenever the bartender makes a drink. The bartender could restock the ingredients when supplies are low.



· [Report a typo or other issue](#)

<https://gist.github.com/ray-newby/e24c8c73>

Completed



Previous

Next

