

shirt_exercise

June 30, 2020

1 Use the Shirt Class

You've seen what a class looks like and how to instantiate an object. Now it's your turn to write code that instantiates a shirt object.

2 Explanation of the Code

This Jupyter notebook is inside of a folder called 1.OOP_syntax_shirt_practice. You can see the folder if you click on the "Jupyter" logo above the notebook. Inside the folder are three files: - shirt_exercise.ipynb, which is the file you are currently looking at - answer.py containing answers to the exercise - tests.py, tests for checking your code - you can run these tests using the last code cell at the bottom of this notebook

3 Your Task

The shirt_exercise.ipynb file, which you are currently looking at if you are reading this, has an exercise to help guide you through coding with an object in Python.

Fill out the TODOs in each section of the Jupyter notebook. You can find a solution in the answer.py file.

First, run this code cell below to load the Shirt class.

```
In [8]: class Shirt:

    def __init__(self, shirt_color, shirt_size, shirt_style, shirt_price):
        self.color = shirt_color
        self.size = shirt_size
        self.style = shirt_style
        self.price = shirt_price

    def change_price(self, new_price):

        self.price = new_price

    def discount(self, discount):

        return self.price * (1 - discount)
```

```

In [9]: ### TODO:
#     - instantiate a shirt object with the following characteristics:
#         - color red, size S, style long-sleeve, and price 25
#     - store the object in a variable called shirt_one
#
#
###
Shirt('red', 'S', 'long-sleeve', 25)
shirt_one = Shirt('red', 'S', 'long-sleeve', '25')

In [10]: ### TODO:
#     - print the price of the shirt using the price attribute
#     - use the change_price method to change the price of the shirt to 10
#     - print the price of the shirt using the price attribute
#     - use the discount method to print the price of the shirt with a 12% discount
#
###
print(shirt_one.price)
shirt_one.change_price(10)
print(shirt_one.price)
print(shirt_one.discount(0.12))

25
10
8.8

In [11]: ### TODO:
#
#     - instantiate another object with the following characteristics:
#         - color orange, size L, style short-sleeve, and price 10
#     - store the object in a variable called shirt_two
#
###
Shirt('orange', 'L', 'short-sleeve', 10)
shirt_two = Shirt('orange', 'L', 'short-sleeve', 10)

In [12]: ### TODO:
#
#     - calculate the total cost of shirt_one and shirt_two
#     - store the results in a variable called total
#
###
shirt_one.price + shirt_two.price
total = shirt_one.price + shirt_two.price

In [13]: ### TODO:
#
#     - use the shirt discount method to calculate the total cost if

```

```

#         shirt_one has a discount of 14% and shirt_two has a discount
#         of 6%
#     - store the results in a variable called total_discount
###
shirt_one.discount(0.14)
shirt_two.discount(0.06)

total_discount = shirt_one.discount(0.14) + shirt_two.discount(0.06)

```

4 Test your Code

The following code cell tests your code.

There is a file called tests.py containing a function called run_tests(). The run_tests() function executes a handful of assert statements to check your work. You can see this file if you go to the Jupyter Notebook menu and click on "File->Open" and then open the tests.py file.

Execute the next code cell. The code will produce an error if your answers in this exercise are not what was expected. Keep working on your code until all tests are passing.

If you run the code cell and there is no output, then you passed all the tests!

As mentioned previously, there's also a file with a solution. To find the solution, click on the Jupyter logo at the top of the workspace, and then enter the folder titled 1.OOP_syntax_shirt_practice

```

In [14]: # Unit tests to check your solution
         from tests import run_tests

         run_tests(shirt_one, shirt_two, total, total_discount)

```

```

In [ ]:

```

```

In [ ]:

```