learn-co-curriculum / dsc-object-attributes-lab Public

View license

1 star 109 forks

Star Watch

Code Issues Pull requests Actions Projects Security Insights

solution ...

This branch is 12 commits ahead, 11 commits behind master.

sumedh10 update readme ... on Dec 17, 2019 14

View code

README.md

Object Attributes - Lab

Introduction

In this lab, you'll practice defining classes and instance methods.

Objectives

You will be able to:

- Define and call an instance method
- Define and access instance attributes

Defining Classes and Instance Methods

In the cell below define a `Driver` class.

For this class, create a method called `greet_passenger()` , which returns the string `Hello! I'll be your driver today. My name is` followed by that driver's first name and last name (i.e. `Hello! I'll be your driver today. My name is John Doe`). (Be sure to keep in mind that the driver's name will be stored under two separate attributes: first and last.)

```
# Define Driver Class here with properties for each instance variable
class Driver():
    def greet_passenger(self):
        print("Hello! I'll be your driver today. My name is {} {}".format(self.first
```

Great! Now create an instance of your driver class. Then, create the following attributes for your instance:

- `first` - the first name of the driver. Set it to Matthew.
- `last` - the last name of the driver. Set it to Mitchell.
- `miles_driven` - the number of miles driven by the driver. Set it to 100.
- `rating` - the driver's rating. Set it to 4.9

Finally, use your `greet_passenger()` method for your `Driver` instance object.

```
driver = Driver()
driver.first = "Matthew"
driver.last = "Mitchell"
driver.miles_driven = 100
driver.rating = 4.9
driver.greet_passenger() # Hello! I'll be your driver today. My name is Matthew Mitc
```

```
Hello! I'll be your driver today. My name is Matthew Mitchell
```

Now, create a passenger class with one method `yell_name()` which prints the passenger's first and last name in all caps. (Again first and last will be stored as separate attributes.)

```
# Define Passenger Class here with properties for each instance variable
class Passenger():
    def yell_name(self):
        print("{} {}".format(self.first.upper(), self.last.upper()))
```

Create an instance of your passenger class. Then create an attribute "first" set to "Ron" and an attribute "last" set to "Burgundy". Then call the `yell_name()` method.

```
passenger = Passenger()
passenger.first = "Ron"
passenger.last = "Burgundy"
passenger.yell_name() # "RON BURGUNDY"
```

```
RON BURGUNDY
```

Great work!

Summary

In this lab, you practiced defining classes, creating instances of said classes, and using methods that made calls to object attributes.

Releases

No releases published

Packages

No packages published

Contributors 5

Languages

● **Jupyter Notebook** 100.0%