


 [learn-co-curriculum](#) / [dsc-instance-methods-lab](#) Public 0 stars  165 forks Star Watch ▼

<> Code

 Issues Pull requests Actions Projects Security Insights solution ▼

...

This branch is [8 commits ahead](#), [12 commits behind](#) master.

sumedh10 update readme ...

on Nov 13, 2019

 9[View code](#) README.md

Instance Methods - Lab

Introduction

In the last lesson, you learned about instance methods -- what they are and how to define them. In this lab, you are going to flesh out the `Driver` and `Passenger` classes by writing your own instance methods for these classes.

Objectives

In this lab you will:

- Create an instance of a class
- Define and call an instance method

Define classes and instance methods

You will now define classes and associated instance methods in the cell below:

Remember: *as we learned in the previous lesson, we need to define our instance methods with at least one argument (`self`) in order to call them on an instance object.*

Define a class `Driver` with two instance methods:

- `greeting` : this should return the string `"Hey, how are you?"`
- `ask_for_destination` : this should return the string `"Where would you like to go today?"`

```
# Define Driver class here
class Driver:

    def greeting(self):
        return "Hey, how are you?"

    def ask_for_destination(self):
        return "Where would you like to go today?"
```

Define a class `Passenger` with two instance methods:

- `reply_greeting` : this should return the string `"I am doing well!"`
- `in_a_hurry` : this should return the string `"Punch it! They're on our tail!"`

```
# Define Passenger class here
class Passenger:

    def reply_greeting(self):
        return "I am doing well! Thanks for picking me up today!"

    def in_a_hurry(self):
        return "Punch it! They're on our tail!"
```

Instantiate classes and methods

Great! You've now defined classes and the associated instance methods. You will now actually use them:

Start by instantiating a driver and a passenger. Assign the driver to the variable `daniel` and assign the passenger to `niky` .

```
daniel = Driver() # driver
niky = Passenger() # passenger
```

Alright, you have the passengers and drivers! Now you need to put those instance methods to use. Try them out and assign the return values to the variables below.

- Have `daniel` greet his passenger, who is going to be `niky`. Assign the greeting to the variable `polite_greeting`
- Have `niky` respond by calling `in_a_hurry()`, and assign the return value to the variable, `no_time_to_talk`

```
polite_greeting = daniel.greeting()  
print(polite_greeting)
```

Hey, how are you?

```
no_time_to_talk = niky.in_a_hurry()  
print(no_time_to_talk)
```

Punch it! They're on our tail!

Feel like doing more?

In the cells below, you'll create three different classes that represent animals in a zoo -- lions, tigers, and elephants. Each animal should have a method, `speak()`, which returns a string containing the sound they make (feel free to have some fun with this -- we don't know how to spell the sound an elephant makes any better than you do!).

```
# Create Lion class  
class Lion:  
  
    def speak(self):  
        return "Roar"  
  
  
# Create Tiger class  
class Tiger:  
  
    def speak(self):  
        return "Meow"
```

```
# Create Elephant class
class Elephant:

    def speak(self):
        return "woo-I'm-an-elephant!"
```

Now, in the cell below, create an instance of each animal:

```
simba = Lion()
tony = Tiger()
dumbo = Elephant()
```

Now, add each of them into the list `zoo` in the cell below:

```
zoo = [simba, tony, dumbo]
```

Now, loop through the `zoo` list and call out the `.speak()` method for every animal in the zoo. Make sure you print this in order to see the output!

```
for animal in zoo:
    print(animal.speak())
```

```
Roar
Meow
woo-I'm-an-elephant!
```

Summary

In this lab, you practiced defining classes and instance methods. You then instantiated instances of your classes and used them to practice calling your instance methods.

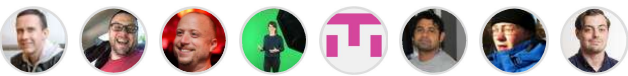
Releases

No releases published

Packages

No packages published

Contributors 8



Languages

