index

February 28, 2022

1 Instance Methods - Lab

1.1 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.

1.2 Objectives

In this lab you will:

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

1.3 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?"`

[1]:
```
# 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!"`

[2]:
```
# Define Passenger class here
class Passenger:
```

```
    def reply_greeting(self):
        return "I am doing well!"
    def in_a_hurry(self):
        return "Punch it! They're on our trail!"
```

1.4 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`.

[3]:
```
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`

[4]:
```
polite_greeting = daniel.greeting()
print(polite_greeting)
```

```
Hey, how are you?
```

[5]:
```
no_time_to_talk = niky.in_a_hurry()
print(no_time_to_talk)
```

```
Punch it! They're on our trail!
```

1.5 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!).

[7]:
```
# Create Lion class
class Lion:
    def speak(self):
        return "rour, rour"
```

[8]:
```
# Create Tiger class
class Tiger:
    def speak(self):
```

```
        return "meow"
```

[9]:
```
# Create Elephant class
class Elephant:
    def speak(self):
        return "WOOOO"
```

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

[10]:
```
simba = Lion()
tony = Tiger()
dumbo = Elephant()
```

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

[11]:
```
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!

[13]:
```
for i in zoo:
    print(i.speak())
```

```
rour, rour
meow
WOOOO
```

1.6 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.