

 [learn-co-curriculum](#) / [dsc-object-initialization-lab](#) Public [View license](#) 0 stars  107 forks Star Watch ▼[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#) solution ▼

...

This branch is [13 commits ahead](#), [12 commits behind](#) master.

sumedh10 update readme ...

on Dec 17, 2019  15[View code](#) README.md

# Object Initialization - Lab

## Introduction

In this lab, you'll practice defining classes with `__init__` methods. You'll define two classes, `Driver` and `Passenger` in the cells below.

## Objectives

In this lab you will:

- Create instance variables in the `__init__` method
- Use default arguments in the `__init__` method

# Initializing Instance Objects

Start off by defining the `Driver` class, similar to as you've done before. This time, define an `__init__` method that initializes a driver with the attributes `first`, `last`, and `occupation` for their first name, last name, and occupation. Provide a default argument of "driver" for `occupation`.

```
# Define Driver Class Here
```

```
class Driver():
```

```
    def __init__(self, first, last, occupation = 'driver'):  
        self.first = first  
        self.last = last  
        self.occupation = occupation
```

Now, initialize a driver with the first name "Dale" and last name "Earnhardt".



gif from [Nascar](#)

```
dale_earnhardt = Driver('Dale', 'Earnhardt') # Initialize Dale Earnhardt here
print(dale_earnhardt.first) # "Dale"
print(dale_earnhardt.last) # "Earnhardt"
print(dale_earnhardt.occupation) # "driving"
```

```
Dale
Earnhardt
driver
```

Next, define the `Passenger` class. Using the `__init__` method, ensure all instances contain the attributes `first`, `last`, `email`, and `rides_taken` for their first name, last name, email, and number of rides they have taken. Provide the `__init__` method with the default argument of `0` for the `rides_taken` attribute since new passengers should not have taken any rides.

```
# Define Passenger class here
class Passenger():

    def __init__(self, first, last, email, rides_taken=0):
        self.first = first
        self.last = last
        self.email = email
        self.rides_taken = rides_taken
```

Now that you've defined a `Passenger` class, check it out by initializing a new passenger with the first name "Jerry", the last name "Seinfeld", and the email "jerry.seinfeld@mailinator.com".

```
jerry = Passenger('Jerry', 'Seinfeld', 'jerry.seinfeld@mailinator.com') # Initialize
print(jerry.first) # "Jerry"
print(jerry.last) # "Seinfeld"
print(jerry.email) # "jerry.seinfeld@mailinator.com"
print(jerry.rides_taken) # 0
```

```
Jerry
Seinfeld
jerry.seinfeld@mailinator.com
0
```

Great work! Mr. Seinfeld is now in the system and ready to request a ride!

## Summary

---

In this lab, you defined `__init__` methods that allowed you to initialize new instances with a set of predetermined attributes and default attributes.

## Releases

No releases published

---

## Packages

No packages published

---

## Contributors 6



## Languages

● Jupyter Notebook 100.0%