Introduction to Object-Oriented Programming: Takeaways

Syntax

DEFINING A CLASS

- To define a class:

```
class Dataset:

    def __init__(self):

        self.type = "csv"
```

- To initialize a class:

```
new_dataset = Dataset()
```

PASSING ADDITIONAL ARGUMENTS TO THE INITIALIZER

- Adding a variable to the initializer:

```
class Dataset:

    def __init__(self, data):

        self.data = data
```

- Initializing a class and accessing an attribute:

```
csv_dataset = Dataset(csv_data)

print(csv_dataset.data[:10])
```

ADDING BEHAVIORS

- Adding a method to our class:

```
class Dataset:

    def __init__(self, data):

        self.data = data

    def print_data(self):

        # New method **remember to add self**.

        print(self.data[:10])
```

- Using the method from our class:

```
nfl_dataset = Dataset(nfl_data)

nfl_dataset.print_data()  # Prints the first 10 rows.
```

ENHANCING THE INITIALIZER

- Adding a header variable:

```
def extract_header(self):

    self.header = self.data[0]

    self.data = self.data[1:]  # set data
```

- Accessing this header variable:

```
nfl_dataset = Dataset(nfl_data)

nfl_header = nfl_dataset.header
```

MAKING OBJECTS READABLE

- To use a special method to print:

```
class Dataset:

    def __init__(self, data):

        self.header = data[0]

        self.data = data[1:]

    def __str__(self):

        data_string = self.data[:10]

        return str(data_string)
```

Concepts

- Python is an **object-oriented programming** language. Everything we use in python is created from a class. Think of a class as a **blueprint** used to construct objects.
- Each blueprint shares the same functions. A function defined in a class is called a **method**.
- A class bundles up logically grouped functions and variables. These groups are called **attributes**. This promotes code **abstraction** which helps us avoid repeating the same code.
- The initialization method creates the instance of the new object and sets those attributes to the instance.
- Whenever we instantiate a new class, we will need to define a **self** variable. `self` is required to define the instance of class.

Resources

- Python Documentation on Classes
- A list of all the built-in methods in Python Classes.