Python Chapter 5: Classes

Ezequiel Torres

June 23, 2024

Table of contents

Class Example

Here is the basics of a class. Classes you can think of like a blueprint for programming. It can store it's own functions and variables as an object.

```
class Dog:
        name = "Lucky"
myDog = Dog()
print(myDog.name)
```

Class Initalize

When wanting to create a class with an initalized variables, you can use the init function.

```
class Dog:
        def __init__(self, name):
                self.name = name
myDog = Dog("Lucky")
print(myDog.name)
```

Class Functions

When wanting to create a class with initalized variables, you can use the init function.

```
class Dog:
        def __init__(self, name, age):
                self.name = name
                self.age = age
        def increaseAge(self):
                self.age += 1
myDog = Dog("Lucky", 2)
myDog.increaseAge()
print(myDog.age)
```

Class Functions

When we think of classes as blueprints, we can begin to think of inheritance as well. We call inheritance a "is-a" relationship. So, we can create a second class stating a golden retriever is a dog.

```
class Dog:
        def __init__(self, name, age):
                self.name = name
                self.age = age
        def increaseAge(self):
                self.age += 1
class GoldenRetriever(Dog):          def __init__(self, name, age,
furColor):
                super().__init__(name, age)
                self.furColor = furColor
myDog = GoldenRetriever("Lucky", 2, "Gold")
myDog.increaseAge()
print(myDog.age)
print(myDog.furColor)
```

Polymorphism

Polymorphism is where a function (or operators) can be executed from the same base class with different outputs

```
class Dog:
    def __init__(self, name, age, furColor = NULL):
        self.name = name
        self.age = age
        self.furColor = furColor
    def increaseAge(self):
        self.age += 1
    def jump(self):
        print("Dog jump!")
```

Polymorphism Continued

```
class GoldenRetriever(Dog):
    def __init__(self, name, age, furColor):
    super().__init__(name, age, furColor)
def jump():
    print("Golden Retriever jump!")
class Husky(Dog):
    def __init__(self, name, age, furColor):
    super().__init__(name, age, furColor)
def jump():
    print("Husky jump!")
myDog = GoldenRetriever("Lucky", 2, "Gold")
theirDog = Husky("Snowy, 3, "White")
thierDog.jump()
```