Day 23: Python Inheritance

Python Inheritance

- Inheritance allows us to define a class that inherits all the methods and properties from another class.
- **Parent class** is the class being inherited from, also called base class.
- **Child class** is the class that inherits from another class, also called derived class.

Create a Parent Class

Any class can be a parent class, so the syntax is the same as creating any other class.

```
class Person:
  def __init__(self, fname, lname):
    self.firstname = fname
    self.lastname = lname

  def printname(self):
    print(self.firstname, self.lastname)

x = Person("John", "Doe")
x.printname()
```

Create a Child Class

- To create a class that inherits the functionality from another class, send the parent class as a parameter when creating the child class.

```
class Student(Person):
  pass
```

```
x = Student("Mike", "Olsen")
x.printname()
```

Add the __init__() Function

- So far we have created a child class that inherits the properties and methods from its parent.
- We want to add the `__init__()` function to the child class (instead of the `pass` keyword).

```
class Student(Person):
  def __init__(self, fname, lname):
    self.firstname = lname
    self.lastname = fname

# When you add the __init__() function,
# the child class will no longer inherit the parent's __init__() function.

class Student(Person):
  def __init__(self, fname, lname):
    Person.__init__(self, fname, lname)

# Python also has a super() function that will make the child class
# inherit all the methods and properties from its parent
class Student(Person):
  def __init__(self, fname, lname):
    super().__init__(fname, lname)
```

Add Properties

```
class Student(Person):
  def __init__(self, fname, lname):
```

```
    super().__init__(fname, lname)
    self.graduationyear = 2019

 # Making it dynamic
 class Student(Person):
  def __init__(self, fname, lname, year):
    super().__init__(fname, lname)
    self.graduationyear = year

x = Student("Mike", "Olsen", 2019)
```

Add Methods

```
class Student(Person):
  def __init__(self, fname, lname, year):
    super().__init__(fname, lname)
    self.graduationyear = year

  def welcome(self):
    print("Welcome", self.firstname, self.lastname, "to the class of", self.graduatior
```

📝 Test