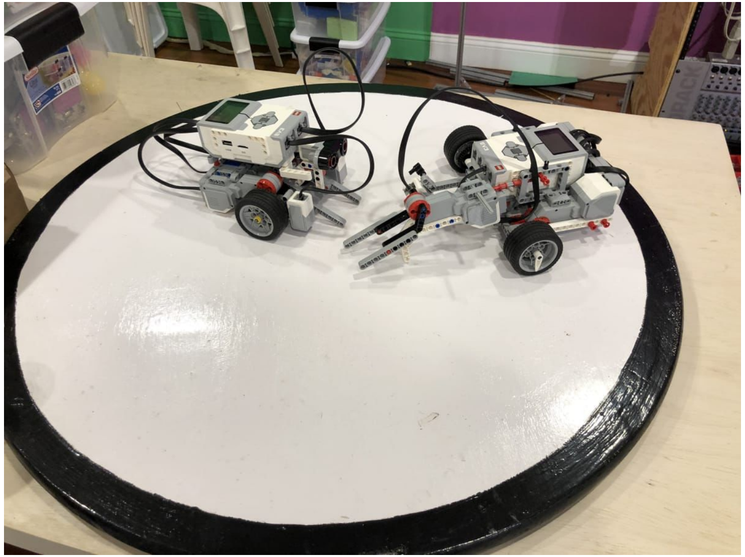
Python Bootcamp Session #3

The DA Programming Club

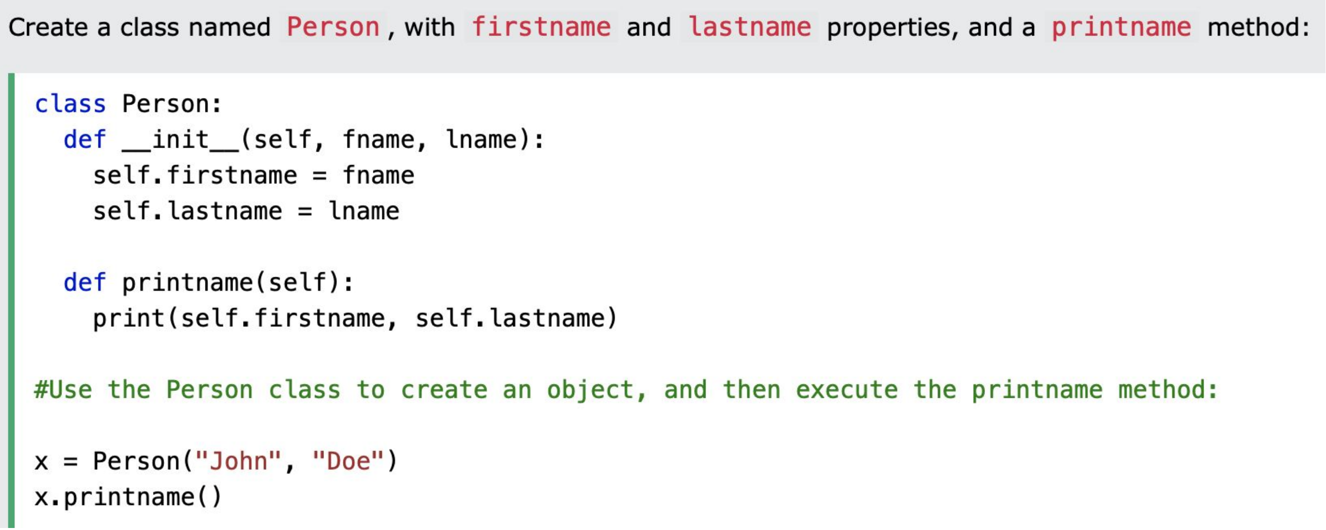
Club Showcase

- When? Next Thursday, the 21st, from 11:10-12:00 in room 245
- What? Our eye-catching attraction is a Battle Bots competition where kids will form teams, have 10 minutes to build an arm to attach to a prebuilt Lego Mindstorm bot, and then compete toe to toe on a Battle Bots board.
  - All robots have the same power and run the same code, so each team's goal is to build the arm that will best deflect other robots attacks such that they are the last bot on the board.



Classes and Inheritance: Parent Class

Any class can be a parent class, so you use the same syntax.

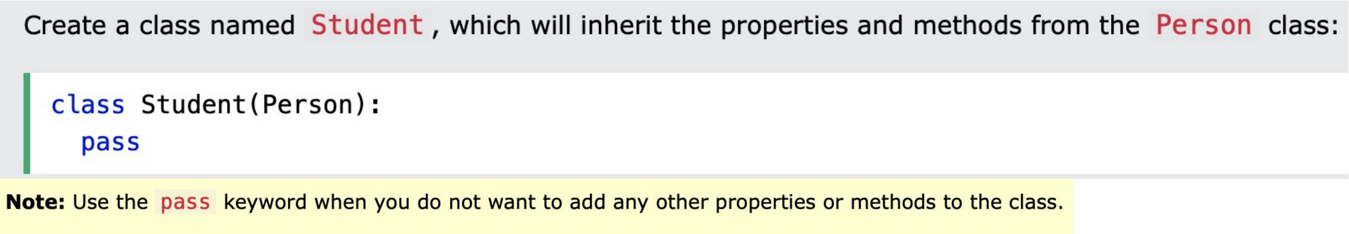


Create a class named `Person`, with `firstname` and `lastname` properties, and a `printname` method:

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

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

#Use the Person class to create an object, and then execute the printname method:

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

Classes and Inheritance: Child Class

To create a child class (a class that inherits functionality from another), send the parent class as a parameter when creating the child class:

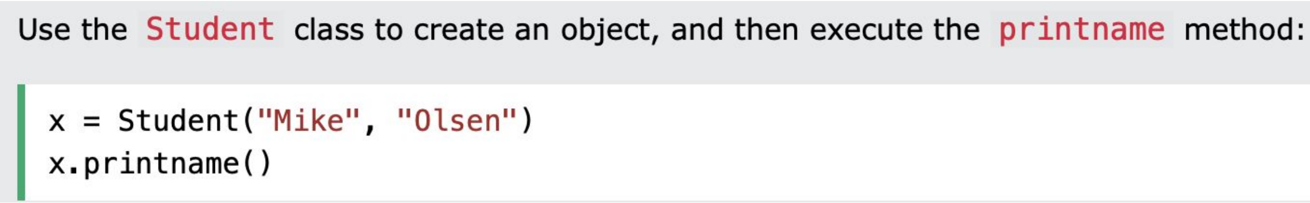


Create a class named Student, which will inherit the properties and methods from the Person class:

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

Note: Use the pass keyword when you do not want to add any other properties or methods to the class.

Now the student class has the same properties and methods as the Person class.



Use the Student class to create an object, and then execute the printname method:

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

Classes and Inheritance

More information on init(), super(), etc available here:

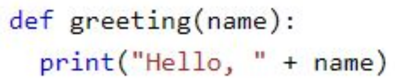
https://www.w3schools.com/python/python_inheritance.asp

Modules

A module is basically a code library. It is a file that has of functions you want to include in your application.

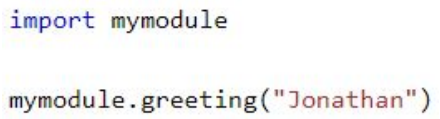
To create a module, just save the code you want in a file with the file extension .py:

Save this code in a file named module.py:



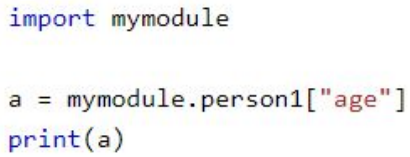
```python
def greeting(name):
  print("Hello, " + name)
```

To use the module you just use the import statement:



```python
import mymodule

mymodule.greeting("Jonathan")
```

To use variables in modules, you can save the variable, dictionary, list, etc. in the module. Then, to call it in a separate file, you can use the module name, then period and the variable name. For example:



```python
import mymodule

a = mymodule.person1["age"]
print(a)
```

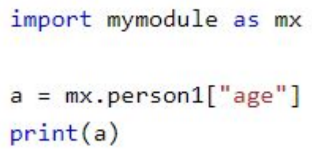
Modules Continued

Naming a Module

You can name the module file whatever you like, but it must have the file extension `.py`

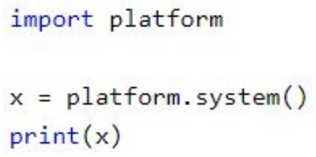
Renaming a Module instance

You can create an alias when you import a module, by using the `as` keyword:



```python
import mymodule as mx

a = mx.person1["age"]
print(a)
```

There are hundreds of built in modules in python. You can import any one you like, just make sure it is installed first. To install a module execute `pip install modulename` in the terminal. To use a built in module in Python the example:



```python
import platform

x = platform.system()
print(x)
```

Intro to Python Google Colab Reference

https://colab.research.google.com/drive/1WlPbtzK0-Kd36I0QNaRbs1y-u8mj70Q2?usp=sharing

New ACSL Content

- If you’re new to ACSL, start from the slideshows available on Blackboard — Contest 1 is the material that’s most pressing to prep for now.
- Check out the “ACSL Supplemental Content” module in Blackboard if you’re a returning competitor
    - That module will be updated regularly as we move along