

Object Oriented Programming with Python

Gramsci Hermozo

Session 03

Content

- Functions
- UML Class Diagram
- Classes in Python

Functions

Defining a Function

```
def function_name(parameters):  
    pass
```

Functions

Function Parameters (1/3)

Parameters could be one or more

```
def greet(name1, name2):  
    prin('Hello', name1, name2)
```

```
greet("Jhon", "Snow")
```

Unknown Number of Arguments

```
def greet(*names):  
    print('Hello', names[0], ",", names[1])
```

```
greet("Jhon", "Snow")
```

Functions

Function Parameters (2/3)

Keyword Arguments

```
def greet(name, lastname):  
    print('Hello', name, lastname)
```

```
greet(lastname="Snow", name="Jhon")
```

__Keyword Argument **kwargs__

```
def greet(**kwargs):  
    print("Hello", kwargs["name"], kwargs["lastname"])
```

```
greet(firtsname="Jhon", name="Snow")
```

Functions

Function Parameters (3/3)

Parameter with default value

```
def greet(name="My Friend"):
    print("Hello", name)
```

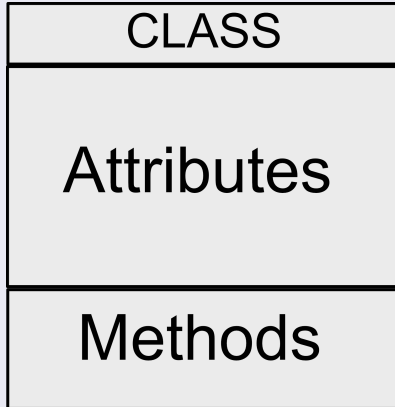
```
greet()
greet("Jhon")
```

Function with Return value

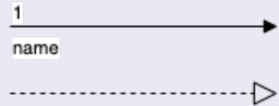
```
def get_name():
    return "GHC"
```

```
name = get_name()
print('Hello', name)
```

Class Diagram



Relationships



Visibility

- Private
+ Public
Protected

Class

Definiton

```
class class_name:  
    pass
```

Class

Class Attributes

```
class Student:  
    school = 'Hogwarts'
```

Class

Class Constructor

```
class class_name:  
    __init__(self):  
        pass
```

Class

Class Properties

```
class class_name:
    __init__(self, prop1, prop2):
        self.prop1 = prop1
        self.prop2 = prop2
```