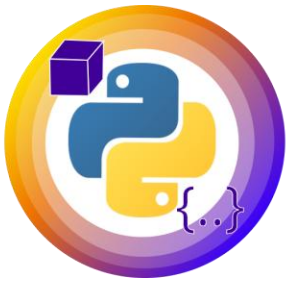

Python OOP: Inheritance
(Methods)

Inheritance (Methods)

Key Takeaways

- **Inheritance (Methods)**
  - Take advantage of natural hierarchies between objects and concepts by creating classes that "inherit" functionality and behaviors from other classes.
    - ✓ For example: a SavingsAccount is a type of account, so it could inherit the functionality of the Account class.
  - All instances of a subclass have access to the methods of the superclass. They can call them, passing the corresponding arguments.
  - *self* will still refer to the instance that calls the method, even if the method belongs to the parent class.
  - When an instance calls a method, the first method that is found in the hierarchy with that name is executed.
  - Method Overriding:
    - Occurs when there are two methods with the same name, one in the superclass and another one in the subclass.
    - The method in the subclass is executed.
    - You can explicitly call the method from the parent class using <parent_class>.<method>(<arguments>)

Key Takeaways

- **General Syntax (First Step – Parent & Child)**

```
class <Superclass>:
    # Body

class <Subclass>(<Superclass>):
    # Body
```

- **Example**

```
class Account:
    # Body

class SavingsAccount(Account):
    # Body
```

Inheritance (Methods)

Key Takeaways

- **General Syntax (Second Step - Methods)**

```
class Account:

    accounts_created = 0

    def __init__(self, number, client, balance):
        self.number = number
        self.client = client
        self.balance = balance
        Account.accounts_created += 1

    def display_balance(self):
        print(self.balance)

class SavingsAccount(Account):

        def __init__(self, number, clie
                Account.__init__(self, numb
                self.interest_rate = interest_

        def display_interest_rate(self):
                print(self.interest_rate)

my_savings_account = SavingsAccount("5621", "G  Navone", 452.34, 0.02)

my_savings_account.display_balance()
```

An instance of the subclass calls a method of the superclass

```
>>> my_savings_account = SavingsAccount("5621", "Gino Navone", 452.34, 0.02)
>>> my_savings_account.display_balance()
452.34
```

Method Call