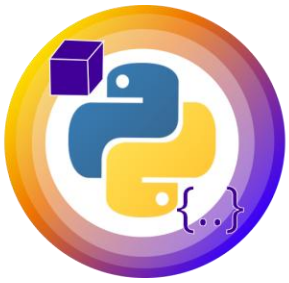

Python OOP: Methods

Key Takeaways

- **Methods**
  - They represent the **actions** that the instances of a class can perform. These actions are relevant to the context of the program.
    - ✓ For example: a BankAccount class could have a display_balance method. It could also have a deposit method to update the current balance.
  - You can think of methods as functions that belong to classes and that instances have special access to.
  - Methods are called by instances.
  - The first parameter of a method is *self*, which refers to the instance that calls the method.
  - You can use methods to modify or create instance attributes and to modify the value of class attributes.
  - Conventions:
    - ✓ Method names contain a verb because they represent actions.
      - ♦ For example: **display**_name, **make**_transfer, **move**_forward.

Methods

Key Takeaways

- **General Syntax**

Keyword

Separated by commas

```
def <method_name>(self, <params>):
    # Code
```

- **Example**

```
class BankAccount:

    accounts_created = 0

    def __init__(self, number, client):
        self.number = number
        self.client = client
        self.balance = balance
        BankAccount.accounts_creat

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

Shared by all instance s

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

Methods

Key Takeaways

- **General Syntax to Call a Method**

"Skipping" self

```
<instance>.<method>(<arguments>)
```

Separated by Commas

- **Example**

```
class BankAccount:

    accounts_created = 0

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

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

my_account = BankAccount("5621", "Gino Navone", 2343.32)

my_account.display_balance()
```

'self' refers to the instance that calls the method

If the method has only one parameter (self) use an empty set of parentheses to call it