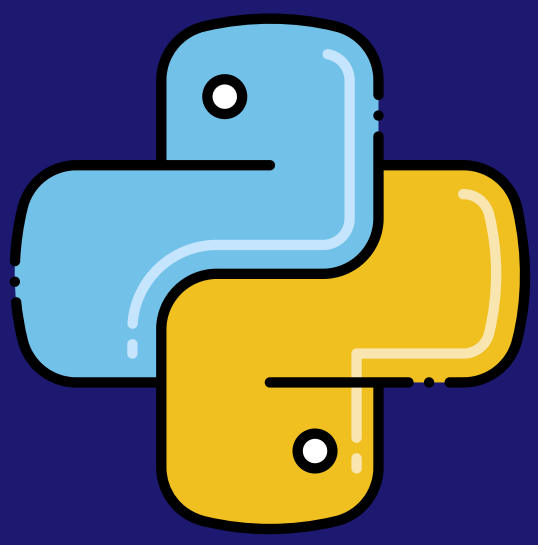
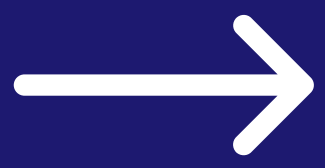
PROGRAMMINGVALLEY.COM



Python OOP

by Amr AbdElkarem

Let’s Swipe



What Is OOP?

Object-Oriented Programming organizes code into objects.
Each object represents a real-world thing with data (attributes) and actions (methods).

Why use OOP?

- Organize complex code
- Reuse code (inheritance)
- Avoid repetition
- Group logic with data

Let’s Swipe

Core Concepts

1. Class and Object

- Class = blueprint (e.g. Dog)
- Object = instance (e.g. my_dog)

```python
class Dog:
    def __init__(self, name):
        self.name = name

my_dog = Dog("Rex")
print(my_dog.name)
```

Let’s Swipe →

Core Concepts

2. __init__ Method

- Special method to initialize an object
- Runs automatically when you create an object

```python
class Car:
    def __init__(self, brand):
        self.brand = brand
```

Let’s Swipe

Core Concepts

3. Instance Attributes

- Belong to the object
- Defined with self. inside methods

```python
class User:
    def __init__(self, name):
        self.name = name
```

Let’s Swipe →

Core Concepts

4. Instance Methods

- Functions inside the class
- Always take self as first argument

```python
class User:
    def greet(self):
        print("Hello!")
```

Let’s Swipe

Core Concepts

5. Inheritance

- Create new class based on an existing one
- Use the parent class features

```
class Animal:
    def speak(self):
        print("Some sound")

class Dog(Animal):
    def speak(self):
        print("Bark")
```

Let’s Swipe

Core Concepts

6. super() Function

- Call parent class method in child class

```
class Dog(Animal):
    def __init__(self, name):
        super().__init__()
        self.name = name
```

Let’s Swipe

Core Concepts

7. Class Attributes

- Shared across all objects

```
class Circle:
    pi = 3.14  # class attribute
```

Let’s Swipe →

Core Concepts

8. Magic Methods

Built-in methods with double underscores
Examples:

- __str__ – string representation
- __len__ – define length
- __add__ – define + behavior

```
class Book:
    def __str__(self):
        return "Book title"
```

Let’s Swipe

Core Concepts

9. Encapsulation

- Keep data private inside class
- Use _ or __ for convention

```python
class Person:
    def __init__(self):
        self._age = 30
```

Let’s Swipe →

Core Concepts

10. Composition (Has-A Relationship)

- Use objects inside other objects

```python
class Engine:
    pass

class Car:
    def __init__(self):
        self.engine = Engine()
```

Let’s Swipe →

Tips

- Use self for instance access
- Use inheritance for related classes
- Use composition for loosely related logic
- Use class attributes sparingly
- Favor readability over clever tricks

Let’s Swipe

PROGRAMMINGVALLEY.COM

Follow us to get more information and tips like these.

by Amr AbdElkarem

@programmingvalley

Save