**Methods in Python Programming**

In Python, **methods** are functions that are associated with an object and can be called on that object. Methods are similar to functions, but they operate on instances of a class and have access to the instance’s data.

**1. Types of Methods in Python**

Python has three types of methods:

1. **Instance Methods** (Most Common)
2. **Class Methods**
3. **Static Methods**

**1. Instance Methods**

Instance methods work with instance variables and can modify the object's state.

**Definition**

* Defined using def inside a class.
* The first parameter is always self, which refers to the instance of the class.

**Example**

class Car:

def \_\_init\_\_(self, brand, model):

self.brand = brand

self.model = model

def display\_info(self):

return f"Car: {self.brand} {self.model}"

# Creating an instance

my\_car = Car("Toyota", "Corolla")

# Calling the instance method

print(my\_car.display\_info()) # Output: Car: Toyota Corolla

**2. Class Methods**

Class methods work with the class itself rather than instances. They are used when you need to access or modify class-level variables.

**Definition**

* Use @classmethod decorator.
* The first parameter is cls, which refers to the class.

**Example**

class Vehicle:

wheels = 4 # Class variable

@classmethod

def set\_wheels(cls, count):

cls.wheels = count

# Changing class variable using class method

Vehicle.set\_wheels(6)

print(Vehicle.wheels) # Output: 6

**3. Static Methods**

Static methods are independent of both the class and instance. They behave like regular functions but are included in a class for better organization.

**Definition**

* Use @staticmethod decorator.
* No self or cls parameter.

**Example**

class MathOperations:

@staticmethod

def add(x, y):

return x + y

# Calling the static method

result = MathOperations.add(5, 3)

print(result) # Output: 8

**Key Differences Between Method Types**

| **Method Type** | **Uses self?** | **Uses cls?** | **Works With Instance Variables?** | **Works With Class Variables?** |
| --- | --- | --- | --- | --- |
| Instance Method | Yes | No | Yes | No |
| Class Method | No | Yes | No | Yes |
| Static Method | No | No | No | No |

**Method Overriding**

When a subclass provides a specific implementation of a method already defined in the parent class.

**Example**

class Parent:

def show(self):

return "This is the Parent class"

class Child(Parent):

def show(self):

return "This is the Child class"

# Creating instance of Child

c = Child()

print(c.show()) # Output: This is the Child class

**Conclusion**

* Methods are functions defined inside a class.
* **Instance methods** operate on instance variables and need self.
* **Class methods** operate on class variables and use cls.
* **Static methods** do not depend on instance or class variables.
* Methods can be overridden in subclasses for custom behavior.