Instance Methods

Instance Methods in Python:

In Python, instance methods are functions that are associated with an instance of a class. They operate on the instance's data and have access to both the instance and its attributes. Instance methods are defined within a class and are called on instances of that class. The first parameter of an instance method is typically named `self` and represents the instance on which the method is called.

Syntax:
```python
class ClassName:
    def instance_method(self, parameter1, parameter2, ...):
        # method implementation
```

Example:

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

    def bark(self):
        print(f"{self.name} says Woof!")

    def celebrate_birthday(self):
        self.age += 1
        print(f"{self.name} is now {self.age} years old!")

# Creating an instance of the class
my_dog = Dog("Buddy", 3)

# Calling instance methods
my_dog.bark()                  # Output: Buddy says Woof!
my_dog.celebrate_birthday()    # Output: Buddy is now 4 years old!
```

In this example, `bark` and `celebrate_birthday` are instance methods of the `Dog` class. They can access the instance attributes (`name` and `age`) using the `self` parameter.

Explanation:

1. **`__init__` Method:**
   - The `__init__` method is a special instance method called a constructor. It is automatically invoked when a new instance of the class is created.

   - It initializes the object's attributes based on the arguments passed during instantiation.

2. **Instance Attributes:**
   - Attributes like `name` and `age` are instance attributes, and they are specific to each instance of the `Dog` class.
   - These attributes are accessed using the `self` parameter inside the instance methods.

3. **`bark` Method:**
   - `bark` is a simple instance method that prints a message using the `name` attribute of the instance.

4. **`celebrate_birthday` Method:**
   - `celebrate_birthday` is another instance method that increments the `age` attribute and prints a birthday message.

5. **Calling Instance Methods:**
   - Instance methods are called on instances of the class using the dot notation (`instance_name.method()`).
   - The instance itself is implicitly passed as the first argument (`self`), so you don't need to pass it explicitly.

Instance methods are essential in encapsulating the behavior of objects and promoting code reusability. They enable objects to perform actions and interact with their own data, contributing to the overall flexibility and maintainability of object-oriented code in Python.