

"self" keyword and "__init__" function

In Python, the `self` keyword and `__init__` method are fundamental concepts related to object-oriented programming (OOP). They are used within classes to refer to the instance of the class and initialize the object's attributes, respectively.

1. `self` Keyword:

- Usage:

- `self` is a convention in Python to represent the instance of the class.
- It is the first parameter in the method definition of instance methods within a class.
- It is used to access the instance variables and call other methods within the class.

- Syntax:

```
class ClassName:  
    def method_name(self, other_parameters):  
        # Method body  
        # Access instance variables using self.attribute_name
```

- Example:

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

```
print(f'{self.name} says Woof!')

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

# Accessing instance variable and calling method using self
print(f'My dog's name is {my_dog.name} and age is {my_dog.age} years.')
my_dog.bark()
```

2. `__init__` Method:

- Usage:

- `__init__` is a special method (constructor) in Python classes.
- It is automatically called when an instance of the class is created.
- It is used to initialize the attributes of the object.

- Syntax:

```
class ClassName:

    def __init__(self, parameter1, parameter2, ...):
        # Initialization code

        # Assign values to instance variables using self.attribute_name
```

- Example:

```
class Car:

    def __init__(self, make, model, year):
        self.make = make
        self.model = model
```

```

        self.year = year
        self.is_running = False

    def start_engine(self):
        if not self.is_running:
            print(f"The {self.year} {self.make} {self.model}'s engine is now
running.")
            self.is_running = True
        else:
            print("The engine is already running.")

# Creating an object of the class
my_car = Car("Toyota", "Camry", 2020)

# Accessing attributes and calling method using self
print(f"My car is a {my_car.year} {my_car.make} {my_car.model}.")
my_car.start_engine()

```

In the above examples:

- **`self`** is used to refer to the instance of the class (``my_dog`` and ``my_car``).
- **`__init__`** is used to initialize the attributes of the object when the instance is created.

Understanding these concepts is crucial in Python OOP as they allow you to work with instance-specific data and behaviors within a class.