**Self in Python:**

* **Introduction**

“**self**” is a special parameter that refers to the instance of a class. It is used within a class definition to refer to the class’s own instance variables and methods. When defining a class, the first parameter in its constructor and instance methods must be “self.” This allows the instance of the class to reference its own attributes and methods. The use of “self” helps in maintaining the uniqueness and identity of each instance of a class, allowing for better organization and encapsulation of code.

* **What is the Need of Self in Python?**

1. **Accessing instance variables and methods:**

“self” is used to access the instance variables and methods within a class. Without “self,” it would be impossible to differentiate between instance variables and class variables or methods.

1. **Creating new instance variables:**

“self” can be used to create a new instance variables.

**\_\_init\_\_() function:**

The name “\_\_init\_\_” is a convention in Python that stands for “initialize.”  \_\_init\_\_ is an instance method that initializes a newly created object. It takes the object as its first argument followed by additional arguments. The method takes the object as its first argument (self), followed by any additional arguments that need to be passed to it.

All classes have a function called \_\_init\_\_(), which is always executed when the class is being initiated.

* **Examples of Using Self in Python:**

Example: Initializing instance variables with “self”

**# A Sample class with init method**

class Person:

**# init method or constructor**

def \_\_init\_\_(self, name):

self.name = name

**# Sample Method**

def say\_hi(self):

print('Hello, my name is', self.name)

p = Person(akash)

p.say\_hi()

**OUTPUT:**

Hello, my name is akash