**Singleton**

A Singleton pattern in python is a design pattern that allows you to create just one instance of a class, throughout the lifetime of a program. Using a singleton pattern has many benefits. A few of them are:

* To limit concurrent access to a shared resource.
* To create a global point of access for a resource.
* To create just one instance of a class, throughout the lifetime of a program

**Classic Singleton:**

Classic Singleton creates an instance only if there is no instance created so far; otherwise, it will return the instance that is already created.

class SingletonClass(object):

def \_\_new\_\_(cls):

if not hasattr(cls, 'instance'):

cls.instance = super(SingletonClass, cls).\_\_new\_\_(cls)

return cls.instance

singleton = SingletonClass()

new\_singleton = SingletonClass()

print(singleton is new\_singleton)

singleton.singl\_variable = "Singleton Variable"

print(new\_singleton.singl\_variable)

**Output**

True

Singleton Variable