

Python - Classe Singleton

Uma classe Singleton é uma classe da qual apenas um objeto pode ser criado. Isso ajuda a otimizar o uso da memória quando você executa alguma operação pesada, como criar uma conexão com o banco de dados.

Exemplo

```
class SingletonClass:
    _instance = None

    def __new__(cls):
        if cls._instance is None:
            print('Creating the object')
            cls._instance = super(SingletonClass, cls).__new__(cls)
        return cls._instance

obj1 = SingletonClass()
print(obj1)

obj2 = SingletonClass()
print(obj2)
```

É assim que funciona o código acima -

Quando uma instância de uma classe Python é declarada, ela chama internamente o método `__new__()`. Substituímos o método `__new__()` que é chamado internamente pelo Python quando você cria um objeto de uma classe. Ele verifica se nossa variável de instância é `None`. Se a variável de instância for `None`, ele cria um novo objeto e chama o método `super()` e retorna a variável de instância que contém o objeto desta classe.

Se vários objetos forem criados, fica claro que o objeto só será criado na primeira vez; depois disso, a mesma instância do objeto é retornada.

Creating the object

```
<__main__.SingletonClass object at 0x000002A5293A6B50>
<__main__.SingletonClass object at 0x000002A5293A6B50>
```

