

Programação Orientada a Objetos

```
if __name__ == '__main__':
```

```
If __name__ == '__main__':
```

Um módulo pode descobrir se ele está ou não sendo executado no escopo principal, verificando seu próprio `__name__`, que é um método especial do Python que retorna o nome do escopo em que o código está sendo executado.

A validação “`If __name__ == '__main__':`” é utilizada quando queremos que parte do código seja executada apenas se o mesmo for executado à partir do escopo principal, não à partir de outro módulo que tenha importado a classe.

If `__name__ == '__main__':`

arq1.py

```
class Arquivo1(object):  
    def __init__(self):  
        print("__name__:", __name__)  
  
    def somar(self, x, y):  
        return x + y  
  
a = Arquivo1()  
print("Soma em arq1:", a.somar(10, 25))
```

```
__name__: __main__  
Soma em arq1: 35
```

Ao executar o arq1.py, quando o objeto “a” foi criado, foi impresso o código do init e, em seguida foi feita a soma e impresso o resultado.

arq2.py

```
from arq1 import Arquivo1  
  
a = Arquivo1()  
print("Soma em arq2", a.somar(10, 10))
```

```
__name__: arq1  
Soma em arq1: 35  
__name__: arq1  
Soma em arq2 20
```

Ao executar o arq2.py, ao criar o objeto “a” o código de arq1 é executado, sendo assim, temos na ordem de execução:

- 1 – Execução do init do objeto “a” de arq1.
- 2 – Execução do print da soma de arq1.
- 3 – Execução do init do objeto “a” de arq2.
- 4 – Execução do print da soma de arq2.

If `__name__ == '__main__':`

```
arq1.py
class Arquivo1(object):
    def __init__(self):
        print("__name__:", __name__)

    def somar(self, x, y):
        return x + y

if __name__ == '__main__':
    a = Arquivo1()
    print("Soma em arq1:", a.somar(10, 25))
```

```
__name__: __main__
Soma em arq1: 35
```

Ao executar o arq1.py, quando o objeto “a” foi criado, foi impresso o código do init e, em seguida foi feita a soma e impresso o resultado. Executado igual anteriormente sem a validação do `__main__`.

```
arq2.py
from arq1 import Arquivo1

a = Arquivo1()
print("Soma em arq2", a.somar(10, 10))
```

```
__name__: arq1
Soma em arq2 20
```

O código que está dentro da validação de `__name__` não é executado quando importamos a classe Arquivo1 do módulo arq1.

FIM