



O que são namespaces?

No Python, todos os nomes de métodos e variáveis precisam ser únicos para evitar conflitos. Imagine se pudéssemos nomear duas variáveis com o mesmo nome, como decidir qual variável seria impressa através do comando print? Haveria um conflito, pois o Python não iria conseguir determinar qual variável você está querendo imprimir. O mesmo ocorre para os métodos, não podemos nomear métodos com o mesmo nome sem que isso gere um conflito.

Sendo assim, o Python utiliza os namespaces para evitar este tipo de conflito. Basicamente, o namespace é um sistema para certificar que todos os nomes em um programa são únicos e podem ser acessados de qualquer classe sem que haja conflito.

O Python implementa os namespaces como dicionários, onde os nomes são as chaves e os objetos seus valores. Sendo assim, é possível utilizar nomes iguais, porém em namespaces diferentes. Sendo assim, é possível implementar nomes de variáveis ou métodos iguais, desde que estejam em namespaces diferentes. Abaixo são listados alguns namespaces presentes no Python:

- **Namespace local:** Inclui nomes locais dentro de uma função. Criado sempre que uma função é chamada, só dura até que a função se encerre.
- **Namespace global:** Inclui nomes de vários módulos importados em um projeto. É criado quando algum módulo é importado em um projeto e dura até o script se encerrar.
- **Namespace interno:** Inclui funções e exceções internas.

Por exemplo, imagine que temos dois métodos chamados `calcular_idade` em duas classes diferentes, porém precisamos utilizar esses dois métodos em uma determinada classe. Para que não haja qualquer conflito entre estes dois métodos, utilizamos o prefixo com o nome do módulo, exemplo:

```
1 classe1.calcular_idade()  
2 classe2.calcular_idade()
```

2%

Python - Orientação a objetos ▾

