

Funcionamento de uma função

Como funciona uma função

Como já sabemos, uma função refere-se a um bloco de código onde queremos encapsular e reaproveitar esse bloco!

E para criarmos uma função em **Python**, utilizaremos o termo **def**, onde refere-se a criação de uma **function**.

```
# Criando funcao
def funcao_teste():
    print('criando funcao')
```

Vale destacar também que todo o código no escopo da função está edentado abaixo da declaração da função.

Outro ponto que é importante destacar, é que quando criamos **functions** no **python**, geralmente separamos por **2 linhas** no **restante do código**, pelas definições de boas práticas!

```
03 - Funcoes > 00 -Codigo Fonte > 01 - funcao.py > ...
1  # Criando funcao
2  def funcao_teste():
3      print('criando funcao')
4
5
6  funcao_teste()
7  |
```

Um ponto **MUITO IMPORTANTE** a se destacar, é que quando estamos criando variáveis no script **Python**, estamos criando **variáveis globais**. Já quando criamos variáveis dentro de uma **function**, estamos criando **variáveis no escopo da function**, logo não conseguiremos ter acesso a essas variáveis fora da **function**!

```
def somar_dois_numeros():
    resultado = 10 + 20
    print(resultado)

print(resultado)
```

Passando parâmetros na function

E para customizar melhor o comportamento das nossas **function**, podemos utilizar parâmetros, que como podemos ver, é bem simples!

```
def somar_dois_numeros(numero1: float, numero2: float):
    resultado = numero1 + numero2
    print(f'{resultado:.2f}')

somar_dois_numeros(10, 20)
```

Veja que definimos os tipos dos parâmetros, para que sejamos obrigados a passar os tipos corretos na **function**, que nesse caso serão números do tipo **float**! Senão, os parâmetros serão definidos do tipo **any**, tornando o parâmetro do tipo mais dinâmico!

Parâmetros Default

Um recurso muito utilizado também referente aos parâmetros de uma **function**, são os **parâmetros default**. Onde nele definimos valores padrões neles, e caso ao consumir essa **function** e não passarmos valores nesses parâmetros, o valor **default** será assumido!

```
# Funcao com parametros DEFAULT
def funcao_default(valor='DEFAULT'):
    print(valor)

funcao_default()
```

Obs: Vale destacar que os parâmetros defaults sempre ficarão no final da definição da função!

Return em funções

Também podemos retornar objetos/dados em nossas funções. E claro, podemos especificar o **tipo** do dado que será retornado, ou podemos definir como **any**, deixando o retorno sendo **dinamicamente tipado**!

```
# Funcao com RETURN ANY
def funcao_return_any():
    return 'Tipado Dinamico'

# Funcao com RETURN TIPADO
def funcao_return_tipo() -> str:
    return 'Tipado'
```