

Laboratório de Programação

Aula 02 – Funções

Prof.: Robson Gomes

Definição de Função

1. “Atividade natural ou característica de um órgão, aparelho, engrenagem etc”.
2. “Obrigação a cumprir, papel a desempenhar”.

- Oxford Languages

```
0 response = requests.get(url)
1
2 # checking response.status_code (if you get 502, try rerunning the code)
3 if response.status_code != 200:
4     print(f"Status: {response.status_code} - Try rerunning the code")
5 else:
6     print(f"Status: {response.status_code}\n")
7
8 # using BeautifulSoup to parse the response object
9 soup = BeautifulSoup(response.content, "html.parser")
10
11 # finding Post images in the soup
12 images = soup.find_all("img", attrs={"alt": "Post Image"})
13
14 # downloading images
15 counter = 0
16 images:
```

Blocos de código que realizam **determinadas tarefas** que normalmente precisam ser executadas **diversas vezes** dentro de uma aplicação.



Por que utilizar Funções?

Reuso de código e eliminação da redundância

- “É comum que um programador necessite executar certa tarefa diversas vezes no mesmo programa”.
- Separa o programa em partes menores que podem ser logicamente compreendidas de forma isolada.
- Ajuda a tornar o programa menor, ao eliminar a repetição do código.
- Facilita a manutenção do código.
- Se precisar realizar alguma alteração, necessário modificar em apenas um local.
- Funções podem ser úteis para muitos programas.
- Quando estudarmos módulos, verão como aproveitar funções e diversos códigos!!

A sintaxe de uma função pode ser definida por: **nome**, **parâmetros** e **corpo**.

```
def nomeDaFuncao (parâmetro1, parâmetro2, parâmetro3, ...N):  
    comando 1  
    comando 2  
    ...  
    comando N  
return algo
```

```
def imprimeMensagem():  
    print("Olá mundo")
```

```
imprimeMensagem()
```

Saída: Olá mundo

```
def imprimeMensagem(msg):  
    print(msg)
```

```
imprimeMensagem("Olá mundo")
```

Saída: Olá mundo

```
def imprimeMensagem(msg):  
    print(msg)
```

```
mensagem = input("Digite a mensagem: ")  
imprimeMensagem(mensagem)
```

Funções

Return

```
def retornaAlgo():
```

```
    a = 5
```

```
    return a
```

Saída: ??

```
retornaAlgo()
```

```
def retornaAlgo():
```

```
    a = 5
```

```
    return a
```

Saída: 5

```
print(retornaAlgo())
```

```
def retornaAlgo():
```

```
    a = 5
```

```
    return a
```

```
resultado = retornaAlgo()
```

```
print(resultado) Saída: 5
```

Toda função em Python retorna algo, até mesmo quando o retorno não é especificado.

```
def retornaAlgo():  
    a = 5  
    # return a
```

```
print(retornaAlgo()) Saída: None
```




Desafio!!!



Crie uma função em Python que receba dois números como parâmetros e retorna a soma dos dois.

Obs.: Os valores devem ser solicitados ao usuário

```
def somaDois(num1, num2):  
    soma = num1 + num2  
    return soma
```

```
num1 = int(input("Digite o primeiro número: "))  
num2 = int(input("Digite o segundo número: "))  
print(somaDois(num1, num2))
```

Desafio!!!

ACABA PELO AMOR DE DEUS



ACABA

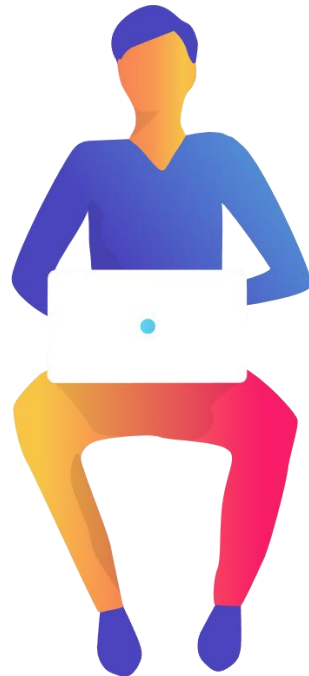
WWW.EACH-SKIN-STORIES.COM



Desafio!!!

Desafio!!!

Efetue o download do exercício resolvido da Aula 01 “Aula 01 – Exercício Resolvido.py”. Em seguida, implemente **uma função** para realizar os cálculos da MP e MF.



Escopo de Variáveis

```
def f():
```

```
    a = 5
```

```
    print("'a' dentro da função", a)
```

```
    return a
```

```
a = 3
```

```
print("'a' antes de executar a função", a)
```

```
print(f())
```

```
print("'a' depois de executar a função", a)
```

Saída:

3

5

5

3

Funções

Variáveis Locais

```
a = 30
```

```
print("'a' antes da declaração da função", a)
```

```
def f():  
    a = 5  
    print("'a' dentro da função", a)  
    return a
```

```
a = 3
```

```
print("'a' antes de executar a função", a)
```

```
print(f())
```

```
print("'a' depois de executar a função", a)
```

Saída:

30

3

5

5

3

```
a = 30
```

```
print("'a' antes da declaração da função", a)
```

```
def f():
```

```
    global a
```

```
    a = 5
```

```
    print("'a' dentro da função depois de modificar", a)
```

```
    return a
```

```
a = 3
```

```
print("'a' antes de executar a função", a)
```

```
print(f())
```

```
print("'a' depois de executar a função", a)
```

Saída:

30

3

5

5

5

Laboratório de Programação

Aula 02 – Funções

Prof.: Robson Gomes