



# Computational Thinking With Python

Prof. Gilberto Alexandre das Neves  
[profgilberto.neves@fiap.com.br](mailto:profgilberto.neves@fiap.com.br)

# Funções

Funções são as estruturas que permitem ao usuário separar seus programas em blocos.

A implementação de funções em programas é fundamental, pois tem o objetivo de otimizar o código-fonte e evitar replicação de código, ou seja, definimos uma única vez e podemos usar sempre.

Um função é utilizada quando seu nome é chamado no programa principal. Podemos ter funções onde passamos informações, chamados de parâmetros ou argumentos.

Uma função também pode retornar informações.

A sintaxe de uma função é definida por três partes: nome, parâmetros e corpo.

```
def nome_da_função(parâmetros):  
    <corpo da função>  
    return “valor de retorno”
```

- A palavra **def** define o início de uma função.
- **parâmetros** podem existir ou não, pois são as informações que a função pode receber para serem processadas.
- **corpo da função** onde é realizada a sequência de instruções, como entradas, processamentos e/ou saídas.
- **return** deverá ser usado quando existir a necessidade de retornar alguma informação para quem invocou a função.

**Identação** deve possuir 4 espaços em branco e pular 2 linhas para o próximo bloco de instruções (próxima função ou programa principal).

# Exemplo 1

Veja exemplo: uma função simples para exibir “Hello World” e outra função que retorna “Hello World”:

```
1 def frase1() :  
2     print("Hello World!")  
3  
4 def frase2() :  
5     return "Hello World!"  
6  
7 frase1()  
8 texto = frase2()  
9 print(texto)
```

## Exemplo 2

Função para somar 2 números (ele recebe dois números como parâmetros e retorna o resultado da soma dos mesmos)

```
1 def soma(n1, n2) :  
2     result = n1 + n2  
3     return result  
4  
5 print("Digite 2 números inteiros")  
6 num1 = int(input())  
7 num2 = int(input())  
8 resultado = soma(num1,num2)  
9 print(f"A soma é: {resultado}")
```

## Exemplo 3

Função que verifica estado civil (recebe como parâmetro o estado civil e exibe mensagem relativo ao estado civil passado)

```
1 def situacao(estado) :  
2     if estado.lower() == "solteiro" :  
3         print("Você é solteiro")  
4     else :  
5         print("Você é casado")  
6  
7 print("Digite seu estado civil")  
8 ecivil = input()  
9 situacao(ecivil)
```

# Exercícios





1. FUPQ peça para o usuário digitar dois números inteiros. Exiba ao final o resto da divisão (crie uma função com retorno).



2. FUPQ peça para o usuário digitar um número inteiro qualquer. Exiba a raiz quadrada (crie uma função com retorno).



3. FUPQ peça para o usuário digitar vários números inteiros e deve continuar digitando até que digite o valor 0 para encerrar. Exiba para cada número que ele digitar se o número é par ou ímpar (crie uma função sem retorno).



4. FUPQ que peça para o usuário digitar quatro números reais diferentes. Exiba os números digitados em ordem decrescente com 3 casas decimais (crie uma função sem retorno)





Introdução à programação com Python. Nilo Menezes. Novatec, 2019.

Curso Intensivo de Python: Uma introdução prática e baseada em projetos à programação. Eric Matthes. Novatec, 2016.

## Até breve!