

FIAP GRADUAÇÃO

TDS

Computacional Thinking using Python
Funções

Prof. Dr. Daniel Trevisan Bravo

FUNÇÕES EM PYTHON

FUNÇÕES - DEFINIÇÃO

- Funções são blocos de código que executam funcionalidades específicas.
- Normalmente são utilizados para evitar que determinada parte do seu código seja escrito varias vezes.
- Em Python sua sintaxe é definida usando def e atribuindo um nome a ela.

```
def funcao():  
    print("Bloco de código")
```

FUNÇÕES - DEFINIÇÃO

- Observando essa função, podemos extrair algumas informações, iniciando com a palavra reservada para funções *def* o nome atribuído à função *funcao* e os parênteses () utilizado para definição dos dados de entrada da função, também chamados de parâmetros.
- Em seguida usa-se dois pontos : e abaixo o bloco de código a ser executado, que neste caso é apenas imprimir de uma string.
- Para “chamar” uma função, utilizamos o nome que foi definido.

```
def funcao():  
    print("Bloco de código")
```

```
funcao()
```

FUNÇÕES - PARÂMETROS

- Além de executar código, funções também podem receber e retornar dados.
- Podemos enviar dados para uma função através de seus parâmetros.

```
def imprime_nome(nome):  
    print(f"Nome: {nome}")  
  
imprime_nome("Erickson")  
imprime_nome("Renan")  
imprime_nome("Daniel")
```

- A utilização dos valores padrão serve para dar um valor quando quem chamou a função não passar nenhum valor para os parâmetros definidos.

```
def flor(flor='Rosa', cor='Vermelha'):  
    print(f"A cor da {flor} é {cor}")  
  
flor()  
flor("Orquídea", "Azul")
```

CHAMADA DE FUNÇÃO POSICIONAL VERSUS CHAMADA DE FUNÇÃO NOMEADA

- Quando chamamos uma função, podemos utilizar a localização dos parâmetros para fazer o casamento entre o que foi chamado e o que foi definido na função.

```
def monta_computador(cpu='', armazenamento=0, memoria=0):  
    print('A configuração é: \n\t- CPU: {cpu}\n\t- Armazenamento: {armazenamento}Tb\n\t- Memória: {memoria}')  
  
monta_computador('Intel Core i9', 4, 64)
```

- O programador que escreveu a chamada da função *monta_computador* está respeitando a posição dos parâmetros, ou seja:
 - ✓ O valor "Intel Core i9" é referente ao primeiro parâmetro (cpu)
 - ✓ O valor 4 é referente ao segundo parâmetro (armazenamento)
 - ✓ O valor 64 se refere ao terceiro parâmetro (memoria)
- Essa é uma chamada de **função posicional**, ou seja: que respeita a ordem dos parâmetros.

CHAMADA DE FUNÇÃO POSICIONAL VERSUS CHAMADA DE FUNÇÃO NOMEADA

- Outra forma de fazer essa chamada de função é utilizar os nomes dos parâmetros!
- Dessa forma, não é necessário respeitar a ordem de definição dos parâmetros!

```
monta_computador(memoria=64, armazenamento=4, cpu='Intel Core i9')
```


FUNÇÕES COM RETORNO DE DADOS

- As funções também podem retornar valores através da palavra reservada `return`.

```
def soma_dois_numeros(valor1, valor2):  
    soma = valor1 + valor2  
    return soma
```

```
valor_soma = soma_dois_numeros(32, 15)  
print(valor_soma)  
print(soma_dois_numeros(50, 10))
```

- Funções também podem retornar múltiplos dados.

```
def soma_dois_numeros_e_calcula_media(valor1, valor2):  
    soma = valor1 + valor2  
    media = (valor1 + valor2)/2  
  
    return soma, media
```

```
valor_soma = soma_dois_numeros_e_calcula_media(32, 15)  
print(valor_soma)  
print(soma_dois_numeros_e_calcula_media(50, 10))
```

ESQUELETO DE UM PROGRAMA EM PYTHON COM CHAMADAS DE FUNÇÕES

```
# função principal
def main():
    """
    Função principal, será a primeira a ser executado e
    será a responsável pela chamada de outras funções """
    # corpo da função main

# Declaração das funções
def f( parâmetros_de_f ):
    """
    docstring da função f
    """
    # corpo da função f

[...]

# início da execução do programa
#-----
if __name__ == '__main__': # chamada da funcao principal
    main() # chamada da função main
```

EXERCÍCIOS

- Escreva um programa em Python com as funções abaixo e, sem seguida, faça a chamada delas na função “main”:
 - Uma função com dois parâmetros (a e b) que verifique e retorne o maior deles;
 - Uma função com um parâmetro x que calcule e retorne o dobro de x.
- Escreva um programa em Python com as funções abaixo e, sem seguida, faça a chamada delas na função “main”:
 - Uma função com um parâmetro n que verifique se ele é par ou ímpar. Caso ele seja par, retorne 1; caso contrário, retorne 0;
 - Uma função que calcule e retorne o quadrado de um parâmetro b.