

FUNÇÃO

Docente: Simone Neves

POR QUE USAR FUNÇÕES?

 Vantagens:

Reutilização de código

Organização e legibilidade

Facilidade na manutenção

Evita repetições

COMO CRIAR UMA FUNÇÃO?

 Sintaxe:

```
def nome_da_funcao():  
    # bloco de código
```

 Exemplo:

```
def saudacao():  
    print("Olá! Bem-vindo ao curso de Python.")
```

CHAMANDO UMA FUNÇÃO



Como executar uma função:

```
saudacao()
```

Saída:

```
Olá! Bem-vindo ao curso de Python.
```

FUNÇÕES COM PARÂMETROS



Sintaxe:

```
def saudacao(nome):  
    print(f"Olá,  
{nome}!")
```

```
saudacao("Maria")
```

Saída:

Olá, Maria!

FUNÇÕES COM RETORNO (RETURN)



Sintaxe:

```
def soma(a, b):  
    return a + b
```

Saída:

8



Exemplo:

```
resultado = soma(3, 5)  
print(resultado)
```

DIFERENÇA ENTRE PRINT E RETURN

`print:`

apenas exibe o resultado

`return:`

**devolve o resultado para ser usado em outro
lugar do código**

FUNÇÃO COM VALOR PADRÃO (DEFAULT)

 Sintaxe:

```
def saudacao(nome="visitante"):
    print(f"Olá, {nome}!")
```

 Exemplo:

```
saudacao()
saudacao("João")
```

Saída:

```
Olá, visitante!
Olá, João!
```


FUNÇÕES COM MÚLTIPLOS PARÂMETROS



Exemplo:

```
def calcular_media(n1, n2, n3):  
    return (n1 + n2 + n3) / 3
```

```
print(calcular_media(7, 8, 9))
```

Saída:

8.0

FUNÇÃO DENTRO DE OUTRA



Exemplo:

```
def quadrado(x):  
    return x * x
```

```
def soma_quadrados(a, b):  
    return quadrado(a) + quadrado(b)
```

Saída:

13

CONCLUSÃO

Funções tornam o código mais limpo e organizado.

Podem receber parâmetros e retornar valores.

Evitam repetição de código.

São fundamentais na programação moderna!