

Data Science & Big Data



Funções e módulos

Prof. André Grégio



Funções

Uma função é uma sequência de instruções para realizar uma dada tarefa (similar a uma função matemática)

- ▶ Como vimos, um módulo é composto por funções
- ▶ É possível selecionar funções de um módulo, por exemplo:
 - ▶ Importação completa → `import MODULO`
 - ▶ Importação seletiva → `from MODULO import func1, func2`

▶ **PODEMOS CRIAR NOSSAS PRÓPRIAS FUNÇÕES!**

Definição de Funções

Podemos adicionar novas funções em Python criando-as!

- ▶ Definição de função: especifica o nome da função e as instruções a serem executadas quando a função é chamada
- ▶ Palavra reservada: **def**
- ▶ Exemplo:
 - ▶ `def funcao():`
 - Instrucao1
 - ...
 - InstrucaoN

Definição de Funções

- ▶ `def funcao():` → cabeçalho!
 - ▶ `Instrucao1 ... InstrucaoN` → corpo da função

Lembrar:

- ▶ Colocar “:” após nome da função no cabeçalho
- ▶ Indentar o corpo da função corretamente

Exemplo de função simples

- Função fatorial()
 - Deve receber um inteiro “n”
 - Calcula o fatorial de “n”, se possível
 - Retorna o fatorial de “n” calculado

Chamada de função

- ▶ Uma função é chamada por seu nome
- ▶ A **entrada** de uma função é seu argumento
- ▶ A **saída** é o valor de retorno ou resultado
- ▶ Exemplo:
 - ▶ $A = 42 \rightarrow$ variável com um inteiro atribuído a ela
 - ▶ $\text{type}(A) \rightarrow$ type é o nome da função, A é o argumento
 - ▶ $\langle \text{type 'int'} \rangle \rightarrow$ valor de retorno, i.e., resultado da aplicação da função

Chamada de função

- ▶ Python provê funções para conversão de tipo
 - ▶ `int('42')` → converte o argumento para inteiro, se possível
 - ▶ `int(3.1415)` → valor de retorno é 3
 - ▶ `float(42)` → valor de retorno?
 - ▶ `str(42)` → valor de retorno?


Exemplo da função fatorial

```
def fatorial(n):  
    res = 1  
    for i in range(2,n+1):  
        res *= i  
    return res
```



Exemplo da função fatorial

```
def fatorial(n):  
    res = 1  
    for i in range(2,n+1):  
        res *= i  
    return res
```

Entrada
da função



Saída da
função



Exemplo da função fatorial

```
def fatorial(n):  
    res = 1  
    for i in range(2,n+1):  
        res *= i  
    return res
```

```
n = int(input("Digite valor de n: "))  
print(fatorial(n))
```

Recebe valor
de variável do
usuário

Imprime o que
a função
retornar

Composição de funções

```
def fatorial(n):
```

```
    ...
```

```
def imprimeRetornoDoFatorial(n):
```

```
    print(fatorial(n))
```

```
n = int(input("Digite valor de n: "))
```

```
imprimeRetornoDoFatorial(n)
```

Em resumo...

- Elementos: Entrada, instruções/corpo, VALOR DE RETORNO
- Saída DEVE retornar um resultado para quem chamou a função, caso esperado, ou pode IMPRIMIR (sem retorno). Exemplo:
- $\text{fat} = \text{fatorial}(n)$
- Variável “fat” espera um valor de retorno!
- Saída da função == comando return

Argumentos de funções

Ex.: def func(ENTRADA)

- ENTRADA pode ser 0 ou mais “variáveis”
- Argumentos podem ser mandatórios ou opcionais

```
def pergunta(confirma, tentativas = 3, lembrete = "Tente novamente!"):
    while True:
        ok = input(confirma)
        if ok in ('s', 'S'):
            return True
        if ok in ('\n', 'N'):
            return False
        tentativas = tentativas - 1
        if tentativas < 0:
            raise ValueError("Resposta inválida!")
    print(lembrete)
```

Argumentos de funções

- Neste caso, 1 argumento é mandatório

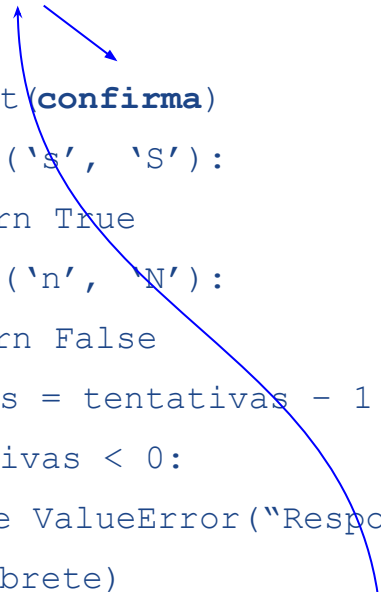
```
def pergunta(confirma, tentativas = 3, lembrete = "Tente novamente!"):
    while True:
        ok = input(confirma)
        if ok in ('s', 'S'):
            return True
        if ok in ('n', 'N'):
            return False
        tentativas = tentativas - 1
        if tentativas < 0:
            raise ValueError("Resposta inválida!")
    print(lembrete)
```

Chamada da função: `pergunta("Deseja confirmar? [S/N]")`

Argumentos de funções

- Neste caso, 1 argumento é **mandatório**

```
def pergunta(confirma, tentativas = 3, lembrete = "Tente novamente!"):
    while True:
        ok = input(confirma)
        if ok in ('s', 'S'):
            return True
        if ok in ('n', 'N'):
            return False
        tentativas = tentativas - 1
        if tentativas < 0:
            raise ValueError("Resposta inválida!")
        print(lembrete)
```



A blue arrow originates from the word **confirma** in the function definition and points to the **confirma** argument in the function call `pergunta("Deseja confirmar? [S/N]")` located at the bottom of the slide. Another blue arrow points from the `input()` function call inside the `while` loop to the **confirma** parameter in the function definition.

Chamada da função: `pergunta("Deseja confirmar? [S/N]")`

Argumentos de funções

- Neste caso, 2 argumentos são **opcionais** (e possuem valor *default*)

```
def pergunta(confirma, tentativas = 3, lembrete = "Tente novamente!"):  
    while True:  
        ok = input(confirma)  
        if ok in ('s', 'S'):  
            return True  
        if ok in ('n', 'N'):  
            return False  
        tentativas = tentativas - 1  
        if tentativas < 0:  
            raise ValueError("Resposta inválida!")  
    print(lembrete)
```


Argumentos de funções

- Neste caso, 2 argumentos são **opcionais** (e possuem valor *default*)

```
def pergunta(confirma, tentativas = 3, lembrete = "Tente novamente!"):  
    while True:  
        ok = input(confirma)  
        if ok in ('s', 'S'):  
            return True  
        if ok in ('n', 'N'):  
            return False  
        tentativas = tentativas - 1  
        if tentativas < 0:  
            raise ValueError("Resposta inválida!")  
    print(lembrete)
```

Exemplo de criação do próprio módulo

ARQUIVO fatorial.py

```
def fat(n):  
    if n >= 0:  
        res = 1  
        for i in range(2,n+1):  
            res *= i  
        return res  
    else:  
        return "Número inválido!"
```

Exemplo de criação do próprio módulo

```
# ARQUIVO main_fat.py
```

```
import sys
```

```
import fatorial
```

```
n = int(sys.argv[1])
```

```
with open(sys.argv[2], 'a') as arq_saida:
```

```
    arq_saida.write(str(fatorial.fat(n))+"\n")
```

Exemplo de criação do próprio módulo



Chamadas:

```
$ python3 main_fat.py 4 res.txt
```

```
$ python3 main_fat.py 0 res.txt
```

```
$ python3 main_fat.py -9 res.txt
```

```
$ python3 main_fat.py 5 res.txt
```



Exemplo de criação do próprio módulo



\$ less res.txt

24

1

Número inválido!

120

