

Python – Funções

Introdução à Programação

SI2

Funções

- O que são?
- Que exemplos já vimos?
- Qual a função de uma função?

Imagine o código

```
matriz1 = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]  
matriz2 = [[2, 4, 3], [2, 1, 7], [4, 3, 2]]  
matriz3 = [[4, 3, 2], [1, 1, 6], [2, 1, 4]]
```

```
#Imprime matriz 1  
for nlinha in range(3):  
    linha = ""  
    for ncol in range(3):  
        linha = linha + str(matriz1[nlinha][ncol])  
    print linha
```

```
#Imprime matriz 2  
for nlinha in range(3):  
    linha = ""  
    for ncol in range(3):  
        linha = linha + str(matriz2[nlinha][ncol])  
    print linha
```

```
#Imprime matriz 2  
for nlinha in range(3):  
    linha = ""  
    for ncol in range(3):  
        linha = linha + str(matriz3[nlinha][ncol])  
    print linha
```

Imagine o código

```
matriz1 = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]  
matriz2 = [[2, 4, 3], [2, 1, 7], [4, 3, 2]]  
matriz3 = [[4, 3, 2], [1, 1, 6], [2, 1, 4]]
```

```
#Imprime matriz 1
```

```
for nlinha in range(3):  
    linha = ""  
    for ncol in range(3):  
        linha = linha + str(matriz1[nlinha][ncol])  
    print linha
```

```
#Imprime matriz 2
```

```
for nlinha in range(3):  
    linha = ""  
    for ncol in range(3):  
        linha = linha + str(matriz2[nlinha][ncol])  
    print linha
```

```
#Imprime matriz 2
```

```
for nlinha in range(3):  
    linha = ""  
    for ncol in range(3):  
        linha = linha + str(matriz3[nlinha][ncol])  
    print linha
```

Repetição
De
Código!

Qual a função da função?

- Evita repetição de código
- Deixa o código Menor
- Mais legível
- Mais modularizado

Função Simples: Soma

```
def soma(a,b) :  
    print(a+b)  
soma(2,9)  
soma(7,8)  
soma(10,15)
```

E se quisermos armazenar em uma variável o valor da soma?

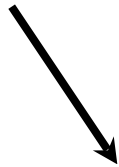
Função Simples: Soma

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

```
x = soma(7,8)
```

Definindo Funções

Definição da função inicia com
"def"




```
def funcao_que_faz_algo(arg1, arg2):  
    """Texto de documentação"""  
    linha1  
    linha2  
    return alguma_coisa
```


Definindo Funções

Definição da função inicia com
"def"

Nome da função



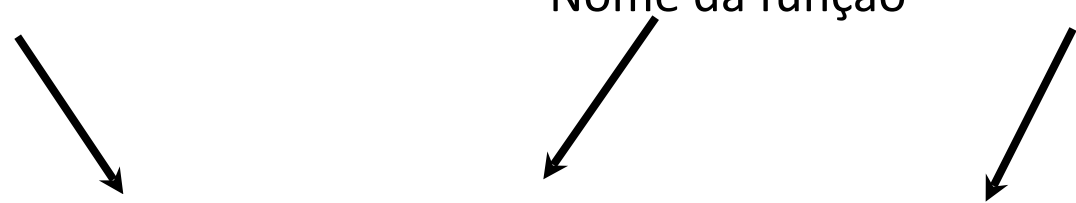
```
def funcao_que_faz_algo(arg1, arg2):  
    """Texto de documentação"""  
    linha1  
    linha2  
    return alguma_coisa
```

Definindo Funções

Definição da função inicia com
"def"

Nome da função

Argumentos



```
def funcao_que_faz_algo(arg1, arg2):  
    """Texto de documentação"""  
    linha1  
    linha2  
    return alguma_coisa
```

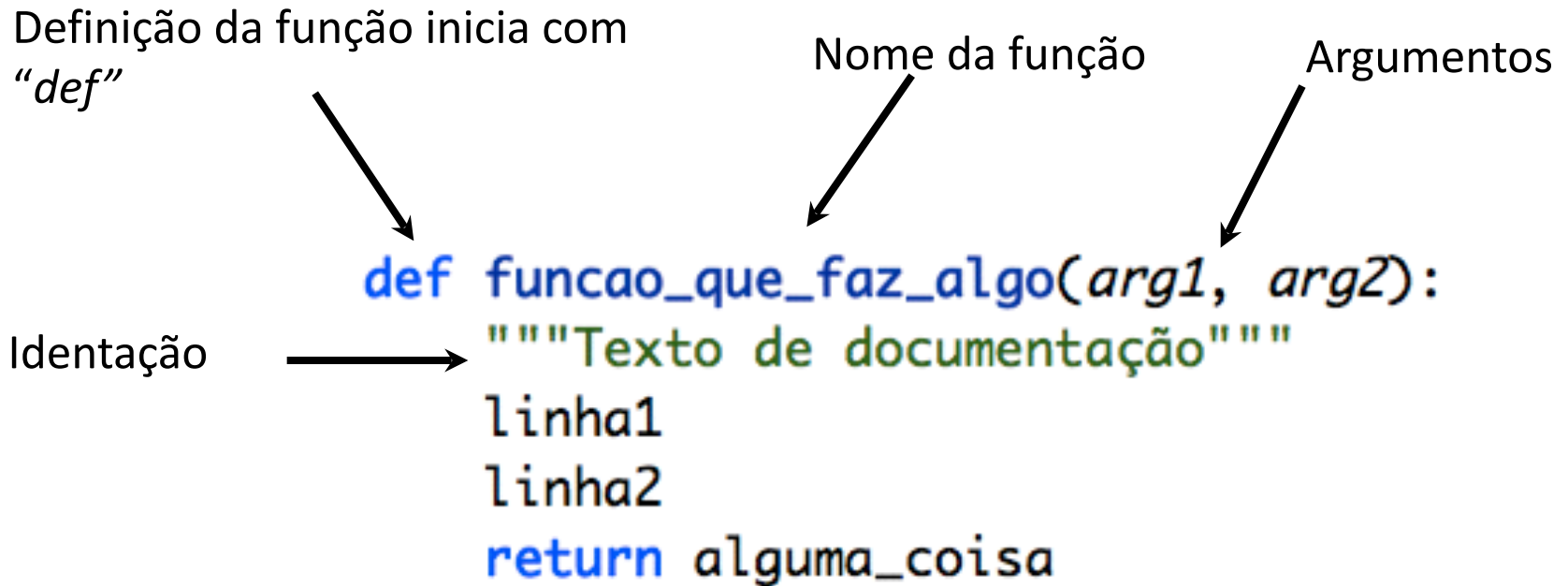
Definindo Funções

Definição da função inicia com
"def"

Nome da função

Argumentos

Identação



```
def funcao_que_faz_algo(arg1, arg2):  
    """Texto de documentação"""  
    linha1  
    linha2  
    return alguma_coisa
```

The diagram illustrates the syntax of a Python function definition. It shows a code snippet with four labels and arrows pointing to specific parts: 'Definição da função inicia com "def"' points to the 'def' keyword; 'Nome da função' points to 'funcao_que_faz_algo'; 'Argumentos' points to '(arg1, arg2)'; and 'Identação' points to the indented lines of the function body. The code snippet is as follows:

Definindo Funções

Definição da função inicia com
"def"

Nome da função

Argumentos

Identação

```
def funcao_que_faz_algo(arg1, arg2):  
    """Texto de documentação"""  
    linha1  
    linha2  
    return alguma_coisa
```

"return" indica o
retorno da função

Resultado de Funções

- Uma função tipicamente computa **um** ou **mais valores**
- Para indicar o valor a ser devolvido como o resultado da função, usa-se o comando **return**, que tem o formato

return expressão

– onde a *expressão* é opcional e designa o valor a ser retornado

Funções

- **return** → termina a função retornando um valor
- O valor default de return → **None**
- Se a função chegar ao fim sem o uso explícito do **return**, então também será retornado o valor **None**

Função para determinar se um valor é par ou não

```
def par(x) :  
    return (x%2==0)
```

```
print(par(2))  
print(par(3))  
print(par(16))
```

Exemplo 2

```
#Declaracao da funcao
def imprime_msg(nomePessoa):
    print "O usuario" + nomePessoa + "escreveu uma mensagem"

#Chamando a funcao
nome = raw_input("Digite o seu nome: ")
imprime_msg(nome)
```


Exemplo 3

```
import random

#Declaracao de funcao que cria lista aleatoria
def criaListaAleatoria (tamanho):
    lista = []
    for i in range(tamanho):
        valor = random.randint(1,10)
        lista.append(valor)
    return lista, max(lista), min(lista)

#Chamada funcao
tam = 5
li, maxli, minli = criaListaAleatoria(tam)
```

Exercício

- Fazer uma função que recebe três argumentos, e que retorne o produto desses três argumentos.

Produto

```
def produto(x,y,z):  
    return (x*y*z)
```

Exercício

- Defina uma função que retorna a palavra “par” ou “ímpar”, de acordo com o valor passado. Reuse a função par()
- Escreva uma função que retorne o maior de dois números passados como argumentos
- Escreva uma função que receba dois números e retorne True se o primeiro número for múltiplo do segundo
- Escreva uma função que receba o lado de um quadrado e retorne sua área ($A = \text{lado}^2$)
- Escreva uma função que receba a base e a altura de um triângulo e retorne sua área ($A = (\text{base} \times \text{altura}) / 2$)

Par ou Ímpar

```
def par(x):  
    return (x%2==0)
```

```
def par_impár(x):  
    if par(x):  
        return "par"  
    else:  
        return "ímpar"
```

```
print (par_impár(4))  
print (par_impár(5))
```

Maior de Dois Números

```
def maior(x,y):  
    if x > y:  
        return x  
    else:  
        return y
```

```
print (maior(6,5))  
print (maior(2,1))  
print (maior(7,7))
```

Múltiplos

```
def multiplo(x,y):  
    if (x%y==0):  
        return True  
    else:  
        return False  
  
print (multiplo(8,4))  
print (multiplo(7,3))  
print (multiplo(7,7))
```

Área Quadrado

```
def area_quadrado(1) :  
    return(1*1)
```

```
print (area_quadrado(8) )  
print (area_quadrado(2) )  
print (area_quadrado(5) )
```


Área Triângulo

```
def area_triângulo(b,a):  
    return (b*a)/2
```

```
print (area_triângulo(8,5))  
print (area_triângulo(2,3))  
print (area_triângulo(5,2))
```

Exercício

- Fazer uma função que receba como parametro um numero inteiro e retorne o fatorial desse numero (não usar recursividade).
- Defina uma função que pesquisa um valor em uma lista. Se o valor for encontrado, devemos retornar a posição deste na lista. Caso não seja encontrado, devemos retornar None
- Calcule a média de valores em uma lista