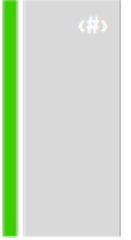




Programação I
Aula 09
Funções

Professor: Francisco Airton
Curso: Sistemas de Informação

Funções



...Seu programa tendo cara de
programa!



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

Resultado de Funções

- Ao encontrar o comando `return`, a função termina imediatamente e o controle do programa **volta ao ponto** onde a função foi chamada
- Se uma função chega a seu fim sem nenhum valor de retorno ter sido especificado, o valor de retorno é `None`

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
```


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



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

Sem declaração do tipo de retorno da função ou do tipo dos argumentos

Chamando uma função

- Sintaxe básica para chamar uma função:

```
>>> def myfun(x, y):  
        return x * y  
  
>>> myfun(3, 4)  
12
```

Passagem de parâmetros

- Várias métricas podem ser usadas na passagem dos parâmetros de uma função:

```
>>> def myfun(x, y):  
        return x * y  
>>> myfun(3, 4)  
12  
>>> myfun(y=3, x=4)  
12  
>>> myfun(3, y=4)  
12
```

Parâmetros com valores padrões

- Cada parâmetro pode ter um valor padrão.
- Utilizado quando um parâmetro é omitido na chamada da função

```
def myfun(x, y = 0.3, prompt = True):
```

```
    res = x * y
```

```
    if prompt:
```

```
        print Resposta é %f, res
```

```
    return res
```

Parâmetros com valores padrões

```
def myfun(x, y = 4, prompt = True):
```

```
    res = x * y
```

```
    if prompt:
```

```
        print Resposta é %i, res
```

```
>>> myfun(3)
```

```
Resposta é 12
```

```
12
```

```
>>> myfun(x=3, prompt = False)
```

```
12
```

```
>>> myfun(5, 2, False)
```

```
10
```

Exemplo

```
>>> def  
    f(nome,saudacao="Oi",pontuacao="!!"):  
        return saudacao+", "+ nome +  
        pontuacao
```

```
>>> print f("Joao")
```

```
Oi,Joao!!
```

```
>>> print f("Joao","Parabens")
```

```
Parabens,Joao!!
```

```
>>> print f("Joao","Ah","...")
```

```
Ah,Joao...
```

Parâmetros com valores padrões

- Valores default são avaliados apenas uma vez

```
>>>def push_buffer(it, buff=[]):  
...     '''Este método armazena na lista  
...     buff o que é passado no argumento it  
...     '''  
...     buff.append(it)  
...     print buff  
  
>>>push_buffer(1)  
[ 1]  
>>>push_buffer(3)  
[ 1, 3]  
>>>push_buffer(7)  
[ 1, 3, 7]
```


Múltiplos argumentos

- Podem receber um número arbitrário de 'keywords'

```
>>>def n_media(**notas):  
...     '''Este método calcula a média das  
...     notas passadas como argumento  
...     '''  
...     return sum(notas.values())/float(len(notas))  
>>>n_media(10,6)  
8  
>>>n_media(10, 6, 4, 8)  
7  
>>>n_media(10, 6, 4, 8, 3, 9, 6, 2)  
6
```

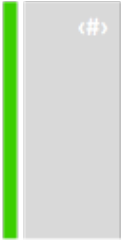
Valor de retorno

(#)

- Toda função em Python retorna algum valor;
- Mesmo funções que não possuem a cláusula **return**.
- Funções que não possuem **return** retornam **None**.
- *None* é uma constante especial definida na linguagem;
- Similar a **NULL**, **void**, ou **nil** em outras linguagens;
- É logicamente equivalente à **False**;



Observações 1/2

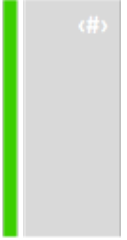


- Funções podem ser utilizadas da mesma maneira que outro tipo de dado em Python
- Elas podem ser:
- Argumentos para outras funções;
- Valores de retorno de outras funções;
- Atribuídas para outras variáveis;
- Partes de tuplas, listas, etc;

..



Observações 2/2



- A execução da função cria uma nova tabela de símbolos, específica para a função.
- Variáveis declaradas dentro da função são unicamente locais!
- São destruídas após a execução de toda a função.
- Funções tem acesso à variáveis com escopo global.
- Utilizando a cláusula `global` você pode acessar/modificar variáveis globais.

Exercícios

1. Fazer uma função que receba como parametro um numero inteiro e retorne o fatorial desse numero (não usar recursividade).
2. Fazer uma função que recebe três argumentos, e que retorne a soma desses três argumentos.
3. Faça uma função chamada somaImposto. A função possui dois parâmetros :
 - a) taxaImposto, que é a porcentagem de imposto sobre vendas
 - b) custo, que é o custo de um item antes do imposto.A função retorna o valor de custo alterado para incluir o imposto sobre vendas.

Exercícios

4. Faça uma função que converta da notação de 24 horas para a notação de 12 horas. Por exemplo, o programa deve converter 14:25 em 2:25 P.M; 6:44 em 6:44 A.M. A entrada é dada em dois inteiros. O programa deve ler várias entradas e chamar uma função para convertê-las e em seguida imprimir a saída.
5. Fazer uma função que recebe um argumento inteiro. A função retorna o valor de caractere 'P', se seu argumento for positivo, e 'N', se seu argumento for zero ou negativo.
6. Faça uma função que retorne o reverso de um número inteiro informado. Por exemplo: 127 -> 721.

Exercícios

7. Faça uma função que informe a quantidade de dígitos de um determinado número inteiro informado pelo usuário.
8. Faça um programa que permita ao usuário digitar o seu nome e em seguida o programa chama uma função que retorna o nome do usuário de trás para frente utilizando somente letras maiúsculas. Dica: lembre-se que ao informar o nome o usuário pode digitar letras maiúsculas ou minúsculas.

Exercícios

9. Faça um programa que solicite a data de nascimento (dd/mm/aaaa) do usuário e imprima a data com o nome do mês por extenso. O programa deve chamar uma função que retorna o mês convertido

Exemplo:

- Entrada - Data de Nascimento: 29/10/1973
- Saída - Você nasceu em 29 de Outubro de 1973.

Exercícios

10. Considere a seguinte fórmula para calcular o mdc (máximo divisor comum) de dois números inteiros positivos:

- $\text{mdc}(a, b) = b$, se b divide a (ou seja, $a \% b == 0$)
- $\text{mdc}(a, b) = \text{mdc}(b, a \% b)$, caso contrário

Escreva uma função em Python que, dados dois números, retorne o máximo divisor comum entre eles. **Usar recursividade.**

Exercício

- Faça o seguinte:
 - Faça uma função que receba um valor inteiro e retorne a divisão por 10 e o resto dessa divisão.
 - Faça uma função que receba um número inteiro entre 0 e 9 e retorne uma string com o nome do algarismo por extenso.
 - Faça um programa que imprima invertido os nomes do algarismos de um número inteiro. (Use suas funções!)
 - Ex: 234 saída: quatro três dois
- Faça uma função que calcule o peso ideal de uma pessoa. A função deve receber o valor da altura da pessoa e um valor numérico indicando o sexo (0 - masculino e 1 - feminino). A função deve calcular o peso ideal de acordo com as seguintes fórmulas: homens $\square (72.7 * altura) - 58$ e mulheres $\square (62.1 * altura) - 44.7$.
 - Faça um programa para ler a altura e o sexo de um conjunto de pessoas e imprimir o peso ideal até que seja digitado um valor negativo para altura.



Exercício 01

- Deve-se fazer uma aplicação que imprime caixas de texto. **Podem** ser especificados a largura (em caracteres) e a altura da caixa (em linhas). As mensagens são de linha única. Caso não sejam especificadas a largura e a altura, deve-se usar a largura igual ao tamanho da mensagem mais 10 e como altura 3.

```
>>>show_message("Hello World!!!",with=30,height=5)
```