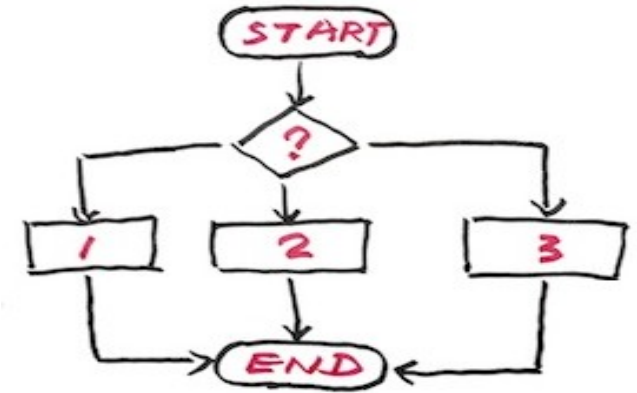


Introdução à Programação

Aula 14

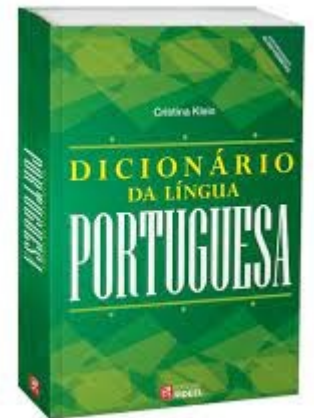
Funções



Aula anterior

Dicionários

- Definição
- Acesso
- Adicionar
- Remover
- Loop
- Copy
- Aninhamento
- Métodos
- Exercícios



Aula de hoje

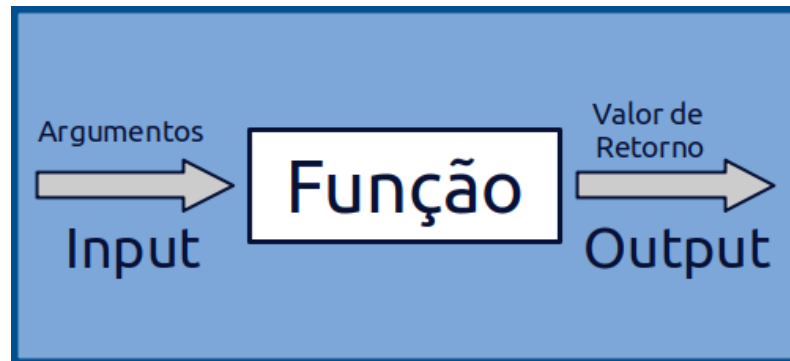
Funções

Introdução

Uma função é um bloco de código que apenas executa quando é chamado.

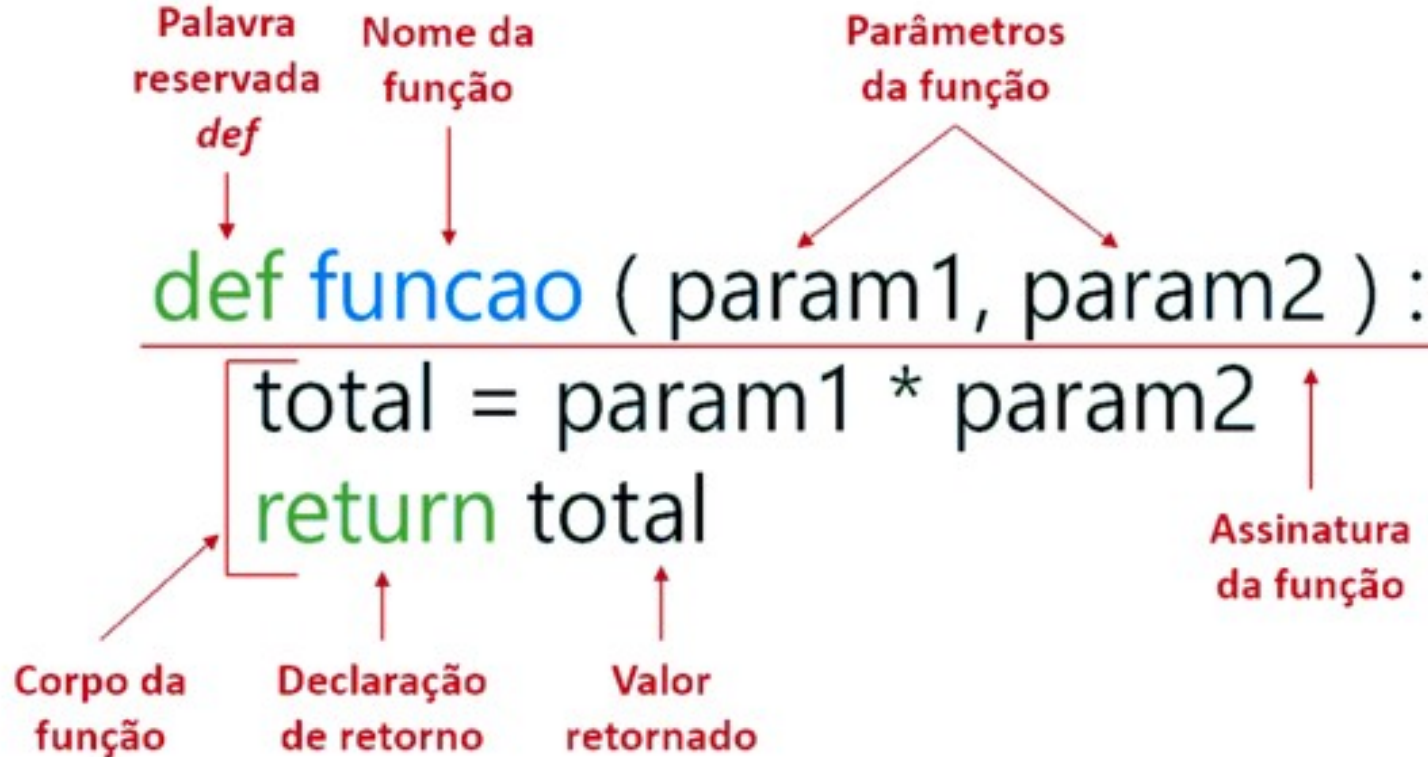
Com funções pode-se:

- Passar dados, conhecidos como parâmetros;
- Retornar dados como um resultado;



Funções

Como declarar uma função



Funções

Criando uma função

Em Python uma função é definida usando a palavra reservada **def**

Example

```
def my_function():  
    print("Hello from a function")
```

Para chamar uma função, use o nome da função seguido por parênteses

Example

```
def my_function():  
    print("Hello from a function")  
  
my_function()
```

Funções

Parâmetros

Informações podem ser passadas para dentro da função como parâmetro

Example

```
def my_function(fname):  
    print(fname + " Refsnes")  
  
my_function("Emil")  
my_function("Tobias")  
my_function("Linus")
```

1. Parâmetros são especificados depois do nome da função (entre parênteses)
2. Pode-se adicionar quanto parâmetros você quiser, apenas separe cada parâmetro por vírgulas.

Atividade 01

Crie uma função que receba o nome de uma pessoa. Exiba dentro a função a mensagem:

Olá [nome] , seja bem vindo!

Chame a função duas vezes!

Example

```
def my_function(fname):  
    print(fname + " Refsnes")  
  
my_function("Emil")  
my_function("Tobias")  
my_function("Linus")
```


Funções

Parâmetros

Informações podem ser passadas para a função como parâmetro

```
def my_function(country = "Norway"):  
    print("I am from " + country)  
  
my_function("Sweden")  
my_function("India")  
my_function()  
my_function("Brazil")
```

1. O exemplo acima mostra como usar um parâmetro com valor padrão.
2. Se a função foi chamada sem parâmetro, ele usa o valor padrão.

Funções

Parâmetros

Você pode enviar qualquer tipo de parâmetro para a função (string, number, list, dictionary e etc)

```
def my_function(food):  
    for x in food:  
        print(x)  
  
fruits = ["apple", "banana", "cherry"]  
  
my_function(fruits)
```

Se você enviar uma lista como um parâmetro, ele ainda será uma lista quando “entrar” na função.

Atividade 02

Crie uma função que receba o nome e cidade de uma pessoa usando um dicionário. Exiba dentro da função a mensagem:

Olá [nome]. [cidade] está 38°!

Chame a função três vezes com parâmetros diferentes.

Funções

Retornos de valores

Para permitir que uma função retorne um valor, use a declaração **return**

```
def my_function(x):  
    return 5 * x  
  
print(my_function(3))  
print(my_function(5))  
print(my_function(9))
```

O valor de retorno da função será exibido pelo comando de saída (print)

Atividade 03

Crie uma função que faça **operação** a **soma de dois números inteiros**. A função deve **receber** dois valores e **retornar** a soma deles.

```
def my_function(x):  
    return 5 * x  
  
print(my_function(3))  
print(my_function(5))  
print(my_function(9))
```

Funções

Argumentos por palavra chaves

Você pode enviar argumento com a sintaxe *chave = valor*

```
def my_function(child3, child2, child1):  
    print("The youngest child is " + child3)  
  
my_function(child1 = "Emil", child2 = "Tobias", child3 = "Linus")
```

Dessa maneira, a ordem dos argumentos não importa.

Funções

Argumentos arbitrários

Se você não sabe a quantidade de argumentos que será passado para a função, add um `*` antes do nome do parâmetro

```
def my_function(*kids):  
    print("The youngest child is " + kids[2])  
  
my_function("Emil", "Tobias", "Linus")
```

Dessa maneira, a função receberá uma tupla de argumentos, e pode acessar os itens respectivamente.

Funções

Argumentos arbitrários

Se você deseja passar o nome da variável para dentro da função, no ato que chama a função, add dois ****** antes do parâmetro.

```
def my_function(**kid):  
    print("His last name is " + kid["lname"])  
  
my_function(fname = "Tobias", lname = "Refsnes")
```

Dessa maneira, a função receberá uma tupla de argumentos, e pode acessar os itens respectivamente.

Atividade 04

Crie uma função que recebe nomes de **idades**. A função deve exibir:

Cidade: cidade[1]

Cidade: cidade[...]

Cidade: cidade[n]

A função não sabe exatamente a quantidade de cidades que será informada.

Funções

Recursão

Python também aceita função recursiva, que significa uma função chama ela mesma.

```
def tri_recursion(k):  
    if(k>0):  
        result = k+tri_recursion(k-1)  
        print(result)  
    else:  
        result = 0  
    return result  
  
print("\n\nRecursion Example Results")  
tri_recursion(6)
```

O desenvolvedor deve ter muito cuidado com recursão. Fácil acontecer um loop infinito, uso grande de memória ou processador.

Funções

pass statement

Definição de uma função não pode ser vazia, mas por alguma razão precisar de uma função sem conteúdo, use a declaração pass para evitar erros.

```
def myfunction():  
    pass
```

O desenvolvedor deve ter muito cuidado com recursão. Fácil acontecer um loop infinito, uso grande de memória ou processador.

Exercícios

SeuNome_Aula14.py

Ex01: Escreva uma função que receba dois números como parâmetros e retorne a soma deles.

Ex02: Escreva uma função que receba uma lista como parâmetro e retorne o maior elemento da lista.

Ex03: Escreva uma função que receba uma string como parâmetro e retorne a quantidade de vogais presentes nessa string.

Ex04: Escreva uma função que receba uma lista de números como parâmetro e retorne a média dos elementos da lista.

Ex05: Escreva uma função que receba um número como parâmetro e verifique se esse número é primo.

SeuNome_Aula14.py

Ex06: Escreva uma função que receba uma string como parâmetro e verifique se ela é um palíndromo.

Ex07: Escreva uma função que receba uma lista de números como parâmetro e retorne uma nova lista contendo apenas os números pares.

Ex08: Escreva uma função que receba uma lista de strings como parâmetro e retorne uma nova lista contendo apenas as strings que possuem mais de 5 caracteres.

Ex09: Escreva uma função que receba um número como parâmetro e exiba a tabuada desse número de 1 a 10.

Ex10: Escreva uma função que receba uma lista de números como parâmetro e retorne o produto de todos os elementos da lista.