



Aula 07

Programação Orientada a Objetos

Funções

Yuri Max

Natal - RN
20 de setembro de 2022

Conteúdo

- Funções def
- Funções lambda
- Funções auxiliares
- Exercícios

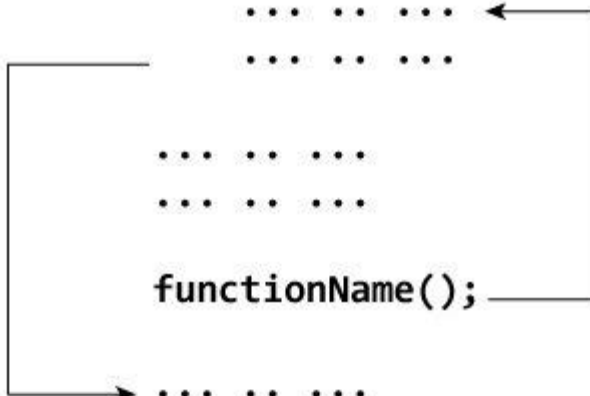
Função def

- É bloco de códigos que só roda quando é chamado;
- Pode possuir argumentos ou não;
- Muito utilizado para deixar o código mais limpo;
- As funções podem ser declaradas no mesmo arquivo do código principal, entretanto, evite;
- É caracterizada por def;

```
def funcaonada():  
    return None  
  
c = funcaonada()  
  
print(c)
```

None

```
def functionName():  
    ... ..  
    ... ..  
  
    ... ..  
    ... ..  
  
functionName();
```



Função def

- Pode receber um ou mais argumentos;
- * recebe uma tupla indefinida de argumentos
- ** prepara a função para receber um dicionário ou chaves e valores;

```
def soma(a, b, c):  
    return int(a+b+c)  
  
def soma2(*a):  
    soma = 0  
    for i in range(len(a)):  
        soma += a[i]  
    return soma  
  
def soma3(**a):  
    soma = 0  
    for i in a.keys():  
        soma += a[i]  
    return soma
```

```
from somas import *  
  
print(soma(1,2,3))  
print(soma2(1,2,3))  
print(soma3(n0 = 1, n1 = 2, n2 = 3))
```

6
6
6

Função def

- Pode especificar um valor default na entrada;
- Uma função pode não ter return;
- É possível entrar com uma lista, tupla, dicionário...;
- Pode-se usar pass quando uma função não estiver implementada;

```
def meupais(pais = 'Brasil'):
    print('Meu país é', pais)

def minhacidade():
    pass
```

```
from pais import meupais

meupais('Holanda')
meupais()
```

```
Meu país é Holanda
Meu país é Brasil
```

Função def

- Python aceita recursão;
- Uma função chama a si própria para analisar todos os dados;
- Podemos adicionar docstrings para dizer o que a função faz;
- Pode-se obter informação da função através de `help()` ou `.__doc__`;

```
def soma4(numero):  
    """  
    Essa função faz a soma recursiva  
    de um inteiro, acumulando até o zero  
    """  
    if numero > 0:  
        soma = numero + soma4(numero-1)  
    else:  
        soma = 0  
    return soma
```

```
from somas import soma4  
  
print(soma4(3))  
  
print(soma4.__doc__)
```

6

Essa função faz a soma recursiva
de um inteiro, acumulando até o zero

Função lambda

- Não são declaradas de forma convencional (def);
- Conhecida como função anônima;
- Só podem acessar variáveis da lista de parâmetros ou globais;
- Pode simplificar muitas linhas de código em apenas uma;
- Aceita mais de uma entrada;

```
x = lambda n, m: n*m  
  
print(x(2,3))  
  
multix = x  
  
print(multix(2,3))
```

```
6  
6
```

Função lambda

- Podem ser usadas dentro de funções convencionais (def);
- Podem ser usadas para cálculos internos ou criar novas funções dentro do código principal;

```
def mult(a):  
    return lambda b: a*b
```

```
from multiplicador import *  
  
duasvezes = mult(2)  
tresvezes = mult(3)  
dezvezes = mult(10)  
  
print(duasvezes(10))  
print(tresvezes(10))  
print(dezvezes(10))
```

```
20  
30  
100
```


Função auxiliares

- `map()` realiza a aplicação de cada elemento de uma lista no segundo argumento no primeiro argumento;
- `filter()` similar ao `map()`, mas só retorna quando o resultado é verdadeiro;
- Podem ser usadas em funções `def` e `lambda`.

```
x = list(map(lambda a: 2*a, range(10)))  
  
y = list(filter(lambda a: a%2 == 0, range(20)))  
  
print(x)  
print(y)
```

```
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18]  
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
```

Funções

- Exemplo:
 - ▶ Crie uma função que calcule o fatorial de um número inteiro dado.
 - ▶ Faça uma função que receba um dicionário. Esse dicionário deve conter nome, sobrenome e cidade de nascimento. A função deve imprimir uma mensagem de boas vindas.
 - ▶ Crie uma função que calcule o quadrado de um número. Depois crie uma função que faça a soma dos quadrados.
 - ▶ Crie uma função que recebe uma string e retorna uma lista com os valores ascii de cada elemento.

Exercícios

1. Escreva uma função que lê uma frase do usuário e retorne em quais posições a letra 'a' apareceram.
2. Crie uma função que recebe do usuário uma string e só encerra quando todos os elementos digitados estiverem entre a-z.
3. Faça uma função que diga quantos números negativos há em uma lista.
4. Crie uma função que receba duas listas aleatórias de 5 elementos e retorne a concatenação delas.
5. Crie uma função que receba dois inteiros, então diga se o primeiro argumento é menor, maior ou igual o segundo argumento.
6. Crie uma função que só encerra quando o usuário digita uma única palavra.
7. Crie uma função que simule um circuito digital entre n entradas. O circuito deve receber 1 para fazer a função AND, 2 para OR, 3 para NAND, 4 para NOR. A função NOT deve estar disponível quando o circuito tiver apenas uma entrada lógica e deve ser acessada por 0.
8. Crie uma função que receba uma matriz e um inteiro. A função deve informar a linha e coluna das entradas da matriz coincidente com segundo argumento, caso existam.
9. Crie uma função que receba uma matriz e arredonde todas as entradas com duas casas decimais.