

Python - Aula 6

Funções

Python - Funções

1) FUNÇÕES

Funções são usadas em programação para agrupar um conjunto de instruções repetitivas ou que por sua complexidade ficam melhores contidas e invocadas quando necessário.

Python - Funções

1) FUNÇÕES

Funções são usadas em programação para agrupar um conjunto de instruções repetitivas ou que por sua complexidade ficam melhores contidas e invocadas quando necessário.

Geralmente são utilizadas para realizar uma **tarefa específica**.

Python - Funções

1) FUNÇÕES

Em Python existem 3 tipos de funções:

- Funções integradas;
- Funções definidas pelo usuário (UDFs);
- Funções anônimas.

Python - Funções

1) FUNÇÕES INTEGRADAS

São funções fornecidas ao programador pela própria linguagem.

Python - Funções

1) FUNÇÕES INTEGRADAS

São funções fornecidas ao programador pela própria linguagem.

Exemplos: `help()`

`print()`

`open()`

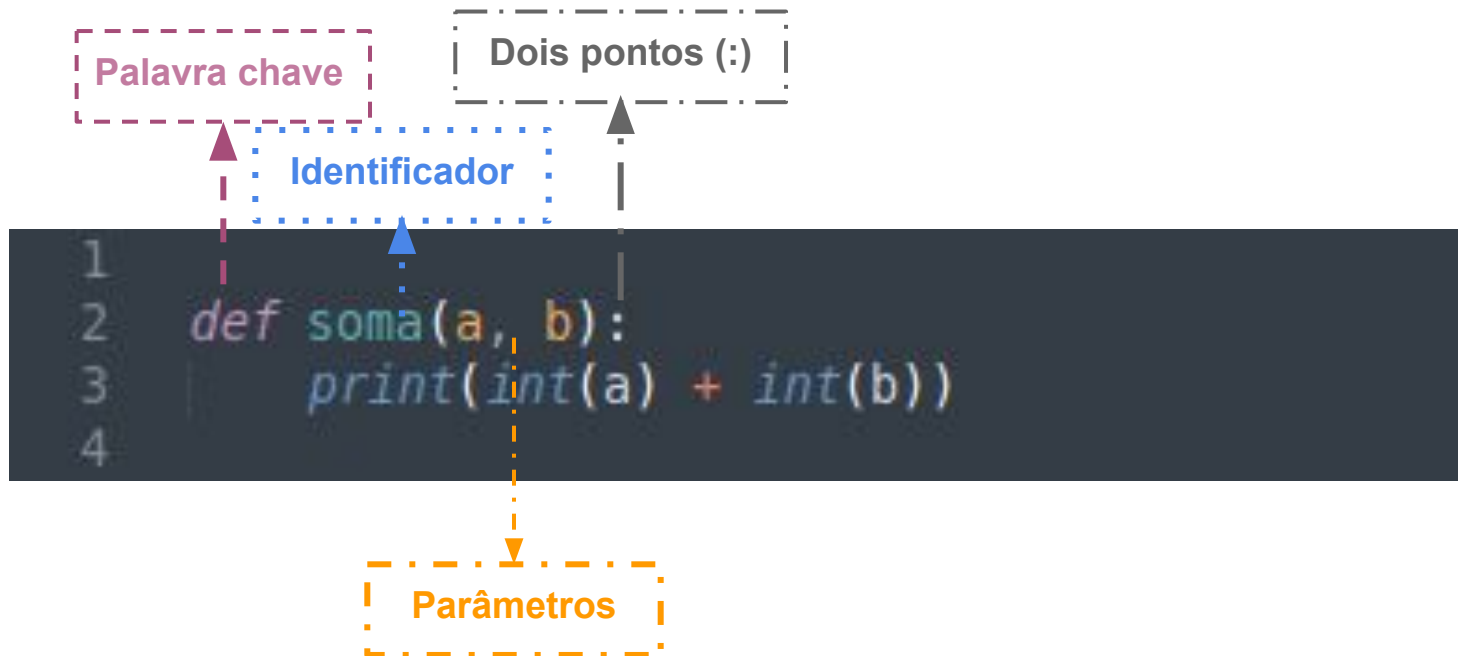
Python - Funções

2) FUNÇÕES (UDF)

```
1  
2  def soma(a, b):  
3      print(int(a) + int(b))  
4
```

Python - Funções

2) FUNÇÕES (UDF)



Python - Funções

2) FUNÇÕES (UDF) - Chamada da função

```
1
2  def soma(a, b):
3      print(int(a) + int(b))
4
5  soma(4, 5)
6
```

Python - Funções

2) FUNÇÕES (UDF) - Parâmetro default

```
1  
2  def soma(a = 0, b = 0):  
3      print(int(a) + int(b))  
4
```


Python - Funções

2) FUNÇÕES (UDF) - Retorno

```
1  
2  def soma(a = 0, b = 0):  
3      return int(a) + int(b)  
4  
5  resultado = soma(4, 5)  
6
```

Python - Funções

2) FUNÇÕES (UDF) - Docstring



```
1
2  def soma(a = 0, b = 0):
3      """Retorna o resultado da soma"""
4      return int(a) + int(b)
5
6  print(soma.__doc__)
7
```

Python - Funções

3) FUNÇÕES - Escopo de variáveis

De forma geral, variáveis podem ter dois tipos de escopo em Python:

a) Local

b) Global

Python - Funções

3) FUNÇÕES - Escopo de variáveis

De forma geral, variáveis podem ter dois tipos de escopo em Python:

- a) **Local** -> **Existe somente no bloco onde foi criada**
- b) **Global** -> **Pode ser acessada por quaisquer funções no código**

Python - Funções

3) FUNÇÕES - Escopo de variáveis

```
1
2  def numero():
3      x = 10
4      print("Valor dentro da função: {}".format(x))
5
6  x = 20
7  numero()
8  print("Valor fora da função: {}".format(x))
9
```

Python - Funções

3) FUNÇÕES - Escopo de variáveis

```
1
2  def numero():
3      x = 10
4      print("Valor dentro da função: {}".format(x))
5
6  x = 20
7  numero()
8  print("Valor fora da função: {}".format(x))
9
```

```
Valor dentro da função: 10
Valor fora da função: 20
>>>
```


Python - Funções

4) FUNÇÕES - Quantidade de argumentos variáveis

```
1  
2  def soma(*args):  
3      return sum(*args)  
4  
5  resultado = soma(1,6,8)  
6
```

Python - Funções

#EXERCÍCIOS

- 1) Faça um programa, com uma função que necessite de um argumento. A função retorna o valor de caractere 'P', se seu argumento for positivo, e 'N', se seu argumento for zero ou negativo.

Python - Funções

#EXERCÍCIOS

- 1) Faça um programa, com uma função que necessite de um argumento. A função retorna o valor de caractere 'P', se seu argumento for positivo, e 'N', se seu argumento for zero ou negativo.
- 2) Faça um programa que use a função valorPagamento para determinar o valor a ser pago por uma prestação de uma conta. O programa deverá solicitar ao usuário o valor da prestação e o número de dias em atraso e passar estes valores para a função valorPagamento, que calculará o valor a ser pago e devolverá este valor ao programa que a chamou. O programa deverá então exibir o valor a ser pago na tela. Após a execução o programa deverá voltar a pedir outro valor de prestação e assim continuar até que seja informado um valor igual a zero para a prestação. Neste momento o programa deverá ser encerrado, exibindo o relatório do dia, que conterá a quantidade e o valor total de prestações pagas no dia. O cálculo do valor a ser pago é feito da seguinte forma. Para pagamentos sem atraso, cobrar o valor da prestação. Quando houver atraso, cobrar 3% de multa, mais 0,1% de juros por dia de atraso.