

Chamadas de função

Objetivos de aprendizagem

- Mostrar como são realizadas chamadas de função.
- Explicar o propósito de algumas funções pré-definidas do Python.
- Explicar o propósito dos argumentos de uma função.
- Criar código que utiliza funções pré-definidas.

1. Destaque as chamadas a funções no programa abaixo.

```
nome = input("Informe seu nome")
print("Nome: ", nome)
```

2. Execute os comandos do quadro abaixo e responda:

```
print(abs(-4.67))
print(pow(5, 3))
print(pow(49, 0.5))
print(int(34.8))
print(round(6.9))
```

- a. Qual é o resultado de cada um desses comandos?

- `print(abs(-4.67))` 4.67
- `print(pow(5, 3))` 125
- `print(pow(49, 0.5))` 7.0
- `print(int(34.8))` 34
- `print(round(6.9))` 7

- b. Qual é a diferença entre as funções `round()` e `int()` ?

A diferença entre as funções `round` vai arredondar o número para a casa mais próxima, e `int` vai tirar os pontos depois da vírgula.

3. Cada uma das funções a seguir retornam um valor. Qual é o valor retornado por cada comando? Verifique a resposta executando o respectivo código.

a. `abs(4.5)` 4.5

b. `int("678")` 678

c. `round(-5.6)` -6

d. `import random`
`random.randint(4,10)` Um número aleatório de 4 à 10.

Qual é o propósito do comando “`import random`”? O que acontece se você omitir essa linha? O propósito do `import random` é importar o módulo `random`, logo se você excluir esse o comando `random.randint(4,10)` vai parar de funcionar.

4. Destaque o argumento na chamada da função do programa abaixo:

```
número = 45.78
resposta = round(número)
```

5. `resultado = pow(4, 3)`. Quais são os argumentos da função desse código? Os argumentos da função é 4 e o 3.

6. Se uma função possui mais de um argumento, você acha que a ordem deles faz diferença? Explique sua resposta com um exemplo.

Sim, por exemplo na função `pow(5, 2)` seu resultado será 25, mas se por acaso for ao contrário `pow(2, 5)` o resultado será 32.

7. Execute o seguinte programa:

```
import math
x = 4.7
y = 5.3
z = -4.8
a = -3.2
print(math.ceil(x))
print(math.ceil(y))
print(math.ceil(z))
print(math.ceil(a))
print(math.floor(x))
print(math.floor(y))
print(math.floor(z))
print(math.floor(a))
```

a. Explique o propósito da função **ceil()**. Vai aproximar um valor acima dele.

b. Explique o propósito da função **floor()**. Vai aproximar para uma casa abaixo dele

c. Por que as chamadas às funções **floor()** e **ceil()** são precedidas por “**math.**”?
Porque eles estão dentro da biblioteca de `math`, sem o `math` os dois não funcionam.

8. Digite e execute o seguinte código:

```
num1 = int(input("Informe o primeiro número: "))
num2 = int(input("Informe o segundo número: "))
num3 = int(input("Informe o terceiro número: "))
num4 = int(input("Informe o quarto número: "))
numeroMáximo = max(num1, num2, num3, num4)
print("O maior dos quatro números é: ", numeroMáximo)
```

a. O que este programa faz?

O programa vai receber quatro variáveis, e o `numeroMáximo` vai escolher quais desses valores é o maior, por fim o programa vai mostrar o valor máximo dentre os 4 números digitados.

b. O que esta linha de código faz?

```
numeroMáximo = max(num1, num2, num3, num4)
```

Compara os argumentos para achar o maior dentre eles.

c. Experimente outros argumentos na função `max` para determinar se a função realmente precisa de quatro argumentos. Dê exemplos para sustentar sua conclusão quanto à quantidade de argumentos necessários. A função precisa de no mínimo mais de 2 argumentos porque senão não tera com quem comparar.

9. Escreva uma linha de código que mostra a parte inteira do número 21.45.

```
numero = 45.5
print(int(numero))
```

10. Escreva um programa que mostre um número aleatório entre 1 e 6.

```
import random  
numeroAleatorio = random.randint(1, 6)  
print(numeroAleatorio)
```