Computação I - Python
Aula 2 - Função

Erros comuns

Apresentado por: Carolina G. Marcelino

Produção DCC-UFRJ

Metodologia de referência https://doi.org/10.5753/wei.2016.9683

Erros comuns

- Utilizando argumentos default
- Passagem de parâmetros
- Usando módulos
- Chamando funções decompostas

Utilizando argumentos default

Vejamos o seguinte problema físico.

Quando solto no vácuo, um objeto chega ao chão após um tempo de 1.5s, em um local onde a gravidade é constante e igual a 9.8 m/s^2. Queremos saber a alura aproximada que esse objeto foi solto. Bem, para realizar este cálculo podemos usar a fórmula de queda livre, da altura em relação ao tempo.

$$h = \frac{g \times s^2}{2}$$

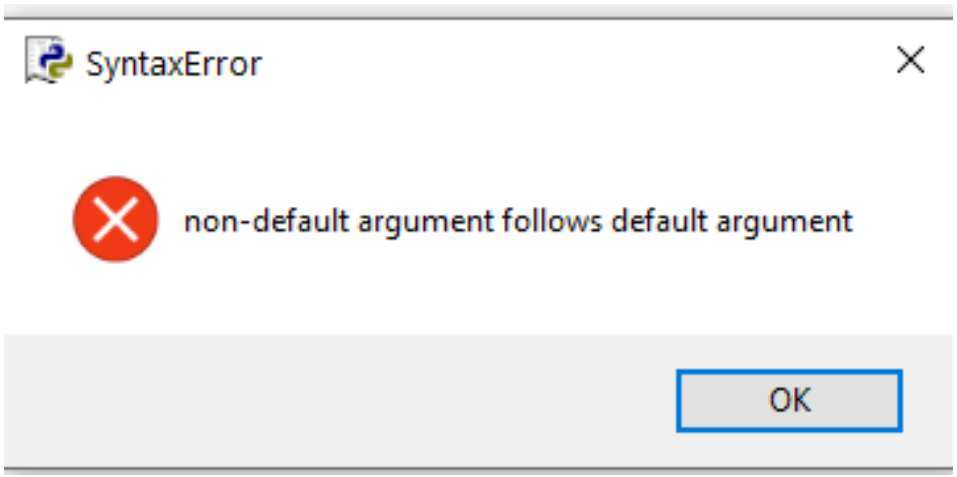
Vamos construir uma função para este cálculo?

Utilizando argumentos default

```
1 def quedaLivre(g=9.8, s):
2     """Funcao que calcula a altura de queda de um objeto,
3        dados a gravidade e o tempo"""
4     return (g*(s**2))/2
```

Utilizando argumentos default

- Ao tentar executar a função quedaLivre, o interpretador do Python gerou o seguinte erro:



Utilizando argumentos default

- Como vimos os argumentos default devem ser declarados após os argumentos sem valores defaults
- Agora vemos a função com a declaração de argumentos da forma correta!

```
1 def quedaLivre(s,g=9.8):
2     """Funcao que calcula a altura de queda de um objeto,
3     dados a gravidade e o tempo"""
4     return (g*(s**2))/2
5
6 >>> quedaLivre(1.5)
7     11.025
```

Passagem de parâmetros

Vamos a outro exemplo.

- Queremos calcular o valor de $x^2 + 2x$.

```
1 def potencia(a,b):
2     """Funcao que calcula a potenciacao de um numero
3         em relacao a outro"""
4     return a**b
5 def equacao(x,y):
6     """Funcao que calcula x^2 + x """
7     return potencia(a,b) + 2*x
```

```
1 >>> equacao(2,2)
2 Traceback (most recent call last):
3 File "<pyshell#16>", line 1, in <module>
4 equacao(2,2)
5 File "C:/Users/carol/equacao.py", line 12, in equacao
6     return potencia(a,b) + 2*x
7 NameError: name 'a' is not defined
```

Passagem de parâmetros

Vejamos a forma correta

- Queremos calcular o valor de $x^2 + 2x$.

```
1  def potencia(a,b):
2      """Funcao que calcula a potenciacao de um numero
3          em relacao a outro"""
4      return a**b
5  def equacao(x,y):
6      """Funcao que calcula x^2 + x """
7      return potencia(x,y) + 2*x
8
9  >>> equacao(2,2)
10 8
```

Passagem de parâmetros

Agora queremos saber o valor da área do perímetro de um círculo.

- Formula: $C = 2 \times \pi \times r$.

```
1  import math
2  def perimetro(r):
3      """Funcao que calcula o perimetro de
4          um circulo"""
5      return 2* pi *r
6
7  >>perimetro(2)
8
9  Traceback (most recent call last):
10 File "<pyshell#27>", line 1, in <module>
11 perimetro(2)
12 File "C:/Users/carol/perimetro.py", line 16, in perimetro
13 return 2* pi *r
14 NameError: name 'pi' is not defined
```

- O que há de errado?

Usando módulos

Agora queremos saber o valor da área do perímetro de um círculo.

- Notou a diferença?

```
1 import math
2 def perimetro(r):
3     """Funcao que calcula o perimetro de
4         um circulo"""
5     return 2*math.pi*r
6
7 >>perimetro(2)
8 12.5
```

Chamando funções

Neste último exemplo vamos simular o início da contrução de uma solução computacional que realiza operações matemáticas simples.

```
1  def soma(x,y):
2      """ Funcao que realiza soma de
3      dois numeros"""
4      return x+y
5
6  def sub(x,y):
7      """ Funcao que realiza a diferenca
8      de dois numeros"""
9      return x-y
10
11 def calculadoraSimples(x,y):
12     """ Funcao que realiza operacoes
13     matematicas simples"""
14     return soma(Sub(x,y),soma(x,y))
```

Chamando funções

Vamos testar a função calculadoraSimples

```
>> calculadoraSimples(5,5)

Traceback (most recent call last):
File "<pyshell#37>", line 1, in <module>
calculadoraSimples(5,5)
File "C:\Users\carol\calculadora.py", line 14, in
    calculadoraSimples
return soma(Sub(x,y),soma(x,y))
NameError: name 'Sub' is not defined
```

Erro!

Chamando funções

Vamos testar a função calculadoraSimples

```
1  def soma(x,y):
2      """ Funcao que realiza soma de
3      dois numeros"""
4      return x+y
5
6  def sub(x,y):
7      """ Funcao que realiza a diferenca
8      de dois numeros"""
9      return x-y
10
11 def calculadoraSimples(x,y):
12     """ Funcao que realiza operacoes
13     matematicas simples"""
14     return soma(sub(x,y),soma(x,y))
15
16 >> calculadoraSimples(5,5)
17 >>10
```

Resumo

Resumo:

- Neste vídeo vimos alguns erros comuns relacionados a definição e uso de funções.

Autores

- João C. P. da Silva ▸ Lattes
- Carla Delgado ▸ Lattes
- Ana Luisa Duboc ▸ Lattes

Colaboradores

- Anamaria Martins Moreira ▸ Lattes
- Fabio Mascarenhas ▸ Lattes
- Leonardo de Oliveira Carvalho ▸ Lattes
- Charles Figueiredo de Barros ▸ Lattes
- Fabrício Firmino de Faria ▸ Lattes

Computação I - Python
Aula 2 - Função

Erros comuns

Apresentado por: Carolina G. Marcelino

Produção DCC-UFRJ

Metodologia de referência https://doi.org/10.5753/wei.2016.9683