Funções

Introdução a Programação

Objetivos de Aprendizagen

- Construir funções personalizadas
- Utilizar módulos

Agenda

- Funções personalizadas
- Módulos

Conceito

> Uma função é um bloco de código que é executada quando é chamada. Dados (**parâmetros**) podem ser passados para função. Uma função pode ainda **retornar** um resultado.

Criando funções

`def`

- Para declarar uma função utiliza-se `def` e indentação
- `def` também é uma palavra reservada

```
1  def my_function():
2      print("Hello ... ")
```

Chamando funções

- Funções apenas podem ser utilizadas após sua declaração

```
1 def my_function():
2     print("Hello ... ")
3
4 my_function()
```

Argumentos

ou Parâmetros

- Informação é passada às funções através de argumentos ou parâmetros
- Argumentos são sequenciais, ou seja, a ordem importa

```
1  def my_function(name, fname):
2      print("Hello, ", name.upper(), fname.upper())
3
4  my_function('john', 'lennon')
```

Argumentos Não Nomeados

`*args`

- Quando a quantidade de argumentos é indefinida adiciona-se `*` antes do parâmetro

```
1  def my_args(*nums):
2      for n in nums:
3          print(n)
4
5  my_args(1, 2, 4, 6, 8, 10)
```

Retornando valores

`return`

- Funções podem devolver valores, resultados das operações

```
1  def add(*nums):
2      sum = 0
3      for n in nums:
4          sum += n
5      return sum
6
7  print(f"Soma vale: {add(7,6,3)}")
```

Retornando valores

`return`

```
1  def area_ret(base, altura):
2      return base * altura
3
4  print("/|\ Área do Retângulo /|\ ")
5  b = int(input("Entre com a base do retângulo:"))
6  h = int(input("Entre com a altura do retângulo:"))
7  print(f"A área do retângulo vale {area_ret(b, h)}")
```

Argumentos por palavra chave

`**kwargs`

- Argumentos são passados através de pares `nome=valor`

```python
def show_values(**kwargs):
    print(f'Valores recebidos: {kwargs}')

    for valor in kwargs.values():
        print(f'{valor}')

show_values(var1=10, var2=20)
```

Argumentos por palavra chave

`**kwargs`

```python
def show_values(**kwargs):
    print(f'Nomes recebidos (keys): {kwargs}')

    for k in kwargs.keys():
        print(f'{k}')

show_values(var1=10, var2=20)
```

Argumentos por palavra chave

`**kwargs`

```python
def show_user_info(**data):
    for chave, valor in data.items():
        print(f"{chave.capitalize()}: {valor}")

show_user_info(user='jlennon@gmail.com', age=21, city='Lo
```

Valor padrão

```
1 def show_user_info(country='Brazil', **data):
2     print("Origem: ", country)
3     for chave, valor in data.items():
4         print(f"{chave.capitalize()}: {valor}")
5
6 show_user_info(user='jlennon@gmail.com', age=21, city='Lo
7 show_user_info(country='UK', user='jlennon@gmail.com', ag
```

Valor padrão

```python
def calcular_preco(preco, desconto=0):
    apagar = preco - (preco * desconto)
    return apagar

print(calcular_preco(100))
print(calcular_preco(100, 0.1))
```

Módulos

Módulos

> Permitem agrupar código para ser utilizado como uma biblioteca de funções, variáveis e dados em geral. Existem módulos do usuário e aqueles disponibilizados pela linguagem ou por terceiros.

Módulos do usuário

1. Crie um arquivo com a extensão `.py` com as definições de função abaixo
2. Faça o *upload* do arquivo na pasta do Google Colabs com o nome `geocalc.py`

```python
import math
def area_ret(base, altura):
    return base * altura

def area_circ(raio):
    return math.pi*raio*raio
```

Módulos do usuário

3. Imagine que o *script* a seguir foi criado para utilizar as funções contidas no módulo `geocalc.py`

```python
import geocalc
print("/|\ Calculadora de Áreas Online /|\ ")
opc = input("Digite 1 para retângulo ou 2 para círculo:")
if opc == '1':
    b = int(input("Base:"))
    h = int(input("Altura:"))
    print(f"Área do retângulo: {geocalc.area_ret(b, h)}")
else:
    r = int(input("Raio:"))
    print(f"Área do círculo: {geocalc.area_circ(r)}")
```

Módulos do usuário

Como executar no Google Colab

4. Se esse código for executado num ambiente local, ou seja na sua própria máquina, nenhum erro será exibido. Porém, como está sendo executado no ambiente do Google é necessário fazer com que o Colab "enxergue" um arquivo externo
5. Adicione as linhas de código abaixo e em seguida tente rodar novamente

```
from google.colab import drive
drive.mount('/content/gdrive')
import sys
sys.path.insert(0, '/content/gdrive/MyDrive/Colab Notebooks/IP')
```

Renomeando módulos

```python
import geocalc as gc
print("/|\ Calculadora de Áreas Online /|\ ")
opc = input("Digite 1 para retângulo ou 2 para círculo:")
if opc == '1':
    b = int(input("Base:"))
    h = int(input("Altura:"))
    print(f"Área do retângulo: {gc.area_ret(b, h)}")
else:
    r = int(input("Raio:"))
    print(f"Área do círculo: {gc.area_circ(r)}")
```

Importanto partes do módulo

`import from`

```
1  from geocalc import area_circ
2  print(f"A área de um círculo de raio 2 é {area_circ(2)}")
```

Importanto módulo inteiro

`import *`

```
1  from geocalc import *
2  print(f"A área de um círculo de raio 2 é {area_circ(2)}")
```

Exercícios

1

Crie uma função em Python que calcula o delta ($\Delta = b^2 - 4ac$) de uma equação do segundo grau ($ax^2 + bx + c = 0$) a partir dos coeficientes a, b, e c.

2

Crie um *script* em Python que utiliza a função criada no exercício anterior para calcular as raízes de uma equação do segundo grau.

Referências

- Funções e Módulos em Python
- Python Functions
- Python Modules

José Roberto Bezerra

jbroberto@ifce.edu.br

jbroberto76