

Aula 5

Lógica de Programação e Algoritmos

Prof. Vinicius Pozzobon Borin

1

Conversa Inicial

2

- O objetivo deste estudo é aprendermos a construir rotinas de códigos, chamadas de funções

3

- Aprenderemos:
 - Função sem retorno
 - Função com retorno
 - Passagem de parâmetros
 - Escopo de variável
 - Tratamento de exceções
 - Função lambda

4

Funções

5

Definição

- Funções são rotinas de códigos que podem ser executadas quando têm seu nome invocado dentro do programa
- Sabia que você tem trabalhando com funções desde as aulas anteriores?
 - *Print, input, int, range* etc.

6

Print

Parte do código da função *print*

```
static PyObject *  
builtin_print(PyObject *self, PyObject *const *args, Py_ssize_t nargs, PyObject *kwnames)  
{  
    static const char * const _keywords[] = {"sep", "end", "file", "flush", 0};  
    static struct _PyArg_Parser _parser = {"([0000]:print", _keywords, 0);  
    PyObject *sep = NULL, *end = NULL, *file = NULL, *flush = NULL;  
    int i, err;  
  
    if (kwnames != NULL &&  
        !_PyArg_ParseStackAndKeywords(args + nargs, 0, kwnames, &_parser,  
                                     &sep, &end, &file, &flush)) {  
        return NULL;  
    }  
}
```

Fonte: GitHub, 2020.

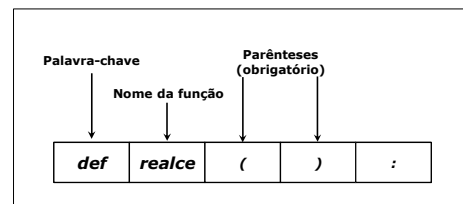
Motivação

As funções:

- Deixam nossos programas mais simples de compreender
- Confinam *bugs* para dentro delas
- Tornam programas mais portáteis
- Auxiliam no trabalho colaborativo

Primeiro, vejamos no Python um exemplo sem funções

A primeira função



A primeira função

Voltamos ao Python para ver como fica nossa primeira função

Fluxo de execução

Linha 1

```
1 def realce():  
2     print('|', ' ' * 10, '|')  
3     print('|', ' ' * 10, '|')  
  
#Programa principal  
1 realce()  
2 print('      MENU')  
3 realce()
```

Fluxo de execução

Linha 2

```

1 def realce():
2     print('|', ' ' * 10, '|')
3     print('|', ' ' * 10, '|')

#Programa principal
1 realce()
2 print('MENU')
3 realce()
  
```

Executa todo o código do print

13

Fluxo de execução

Linha 3

```

1 def realce():
2     print('|', ' ' * 10, '|')
3     print('|', ' ' * 10, '|')

#Programa principal
1 realce()
2 print('MENU')
3 realce()
  
```

14

Parâmetros em Funções

15

Definição

- Parâmetros: dados recebidos pelas funções
- O ato de enviar um dado para uma função é chamado de passagem de parâmetro

16

Função com parâmetros

Palavra-chave	Nome da função	Parênteses (obrigatório)			
def	realce	(s1)	:
			variável		

Fonte: Berti, 2024.

17

- Vamos praticar no Python

18

Parâmetros opcionais

- Podemos dar maior flexibilidade para nossas funções permitindo que nem sempre se use todos os parâmetros na chamada da função
- Vamos ver parâmetros opcionais em Python

Exercício

- Escreva uma rotina que crie uma borda ao redor de uma palavra, para destacá-la como sendo um título. A rotina deve receber como parâmetro a palavra a ser destacada. O tamanho da caixa de texto deverá ser adaptável, de acordo com o tamanho da palavra. Por exemplo:

```
+-----+ +---+
|Vinicius| |Olá|
+-----+ +---+
```

19

20

Escopo de Variáveis

Escopo de variáveis

- Um escopo é a propriedade que determina onde uma variável pode ser utilizada dentro de um programa

21

22

Escopo local

- Criado sempre que uma função é chamada
- Variáveis criadas, seja no campo de um parâmetro ou dentro do corpo da função, fazem parte do escopo local daquela função e são chamadas de variáveis locais. Essas variáveis só existem dentro daquela própria função

Escopo global

- Criado no programa principal
- Variáveis globais pertencem a um escopo global e são variáveis criadas dentro do programa principal. Uma variável global existe também em todas as funções invocadas ao longo do programa

23

24

- Vamos praticar o escopo em Python

25

A instrução global

- Força nosso programa a não criar uma variável local de mesmo nome e manipular somente a global dentro de uma função
- Vejamos em Python

26

Retorno de valores em funções

27

Função x procedimento

- Procedimento (*procedure*) – uma rotina sem retorno
- Função – uma rotina que retorna um dado a quem a invocou

28

- Vamos praticar retorno de funções em Python

29

Exercício

- Escreva uma função para validar uma string. Essa função recebe como parâmetro a string, o número mínimo e máximo de caracteres. Retorne verdadeiro se o tamanho da string estiver entre os valores de mínimo e máximo, e falso, caso contrário (elaborado com base em Menezes, s. d.)

30

Recursos avançados com funções

31

Erro de sintaxe

- Ocorre quando o programador comete algum erro de digitação, ou esquece de alguma palavra-chave, ou caractere, ou mesmo erra na indentação do código
- Veja um exemplo desse erro em Python

32

Exceção

- Neste tipo de erro, a sintaxe está correta, porém um erro durante a execução do programa ocorre, normalmente devido a um dado digitado de maneira inválida e não tratado durante o programa

33

Exceções comuns em Python

- `ZeroDivisionError` – erro de divisão por zero
- `ValueError` – erro de um dado não esperado sendo digitado
- `IndexError` – erro de índice inexistente sendo acessado
- Lista completa de exceções:
 - <https://docs.python.org/pt-br/3/library/exceptions.html#builtin-exceptions>

34

- Vejamos alguns exemplos e também como tratar uma exceção em Python

35

Função lambda

- Funções mais simples, sem nome, chamadas de funções lambda
- Elas podem ser escritas em uma só linha de código e dentro do programa principal
- Vejamos em Python

36

Função lambda: exercício

- Faça uma função lambda que recebe dois valores numéricos como parâmetro. Ao primeiro valor, sempre some 5. Em seguida, multiplique ambos e retorne o resultado

37

38