

# Introdução à Programação em Python

Nelson Seixas dos Santos

Faculdade de Ciências Econômicas  
Universidade Federal do Rio Grande do Sul

March 13, 2019

## Sumário

- 1 Introdução
- 2 Entrada de dados
- 3 Processamento de dados
- 4 Saída de dados
- 5 Exemplos de programas
- 6 Paradigmas de Programação em Python
  - Programação estruturada
  - Programação procedural
  - Programação modular
- 7 Estilo de escrita de código
- 8 Exercícios
- 9 Referências

- Introdução
- Entrada de dados
- Processamento de dados
- Saída de dados
- Exemplos de programas
- Paradigmas de Programação em Python
- Estilo de escrita de código
- Exercícios
- Referências

# 1. Introdução

## A estrutura básica do programa em python

Um programa é composto pelas seguintes partes:

- entrada de dados
- processamento dos dados
- saída dos dados.

## 2. Entrada de dados

## Entrada de dados

- Entrada de dados é o procedimento que consiste em inserir no interpretador/compilador Python os valores das variáveis que desejamos processar.
- Essas variáveis são armazenadas em um endereço de memória do computador.
- Naturalmente, só se pode entrar com dados de tipos que possam ser reconhecidos pelo interpretador Python.
- Python é capaz de reconhecer diversos tipos de dados os quais serão especificados posterior.
- A entrada de dados pode ser feita diretamente pelo programador (atribuição de variáveis), solicitada ao usuário no teclado ou por leitura de arquivos

## Tipos de dados: definição

### Definição (Tipos de dados)

*Tipo de dado é a especificação da forma como o computador armazenará um dado na memória em conjunto com as operações podem ser realizadas com ele. Cada linguagem de programação define os tipos de dados que ela é capaz de manipular.*

## Tipos de dados: exemplos

### Exemplos

- booleano (ou lógico)
- inteiro ou ponto fixo
- real ou ponto flutuante
- complexo
- caracter

## Tipos de dados em Python

- 1 booleano - bool
- 2 inteiro ou ponto fixo - int
- 3 real ou ponto flutuante - float
- 4 complexo - complex
- 5 string - str

## Tipos de dados compostos ou estruturas de dados

### Definição

*São tipos de dados que armazenam mais de um valor de variável simultaneamente. São formados a partir dos tipos de dados existentes.*

## Tipos de dados compostos ou estruturas de dados: exemplos clássicos

- vetores - conjunto ordenado unidimensional de termos de mesmo tipo (inteiros, reais, complexos ou caracteres)
- matrizes - vetores bidimensionais de termos de mesmo tipo (inteiros, reais, complexos ou caracteres)
- arranjos - vetores de mais de 2 dimensões de termos de mesmo tipo (inteiros, reais, complexos ou caracteres)

## Tipos de dados compostos ou estruturas de dados: exemplos clássicos (cont.)

- listas - conjunto ordenado unidimensional de termos de tipos possivelmente diferentes.
- registros - conjunto ordenado multidimensional de termos de tipos possivelmente diferentes. Basicamente, é uma lista de listas.

## Estruturas de dados mais importantes em Python

- listas -  $x = [1,2,3]$
- tuplas -  $x = (1,2,3)$
- conjuntos - `set()`
- dicionários - `dict()`

## Comandos básicos de entrada de dados

- atribuição -  $x = 2$
- leitura do teclado - `input()`
- leitura de arquivos - `open()` / `close()`

### 3. Processamento de dados: estruturas de controle de fluxo de execução

## Processamento de dados: estruturas de controle de fluxo de execução

- O processamento consistirá na execução do algoritmo que soluciona o problema.
- O procesamento é onde são calculados os valores das variáveis de interesse do programador.
- O processamento será composto por instruções de sequência, repetição e decisão tal como previsto no Teorema de Bohn-Jacopini.

## Estruturas de repetição em python

- **for** <condicao >:  
<comandos>
- **while** <condicao >:  
<comandos>

## Estruturas de decisão em python



```
if <condicao >:  
    <comandos>  
elif :  
    <comandos>  
elif :  
    <comandos>  
else :  
    <comandos>
```

## 4. Saída de dados

## Saída de dados

- A saída de dados é parte do programa que envia os resultados obtidos pela execução do programa para um dispositivo de interação com o usuário.
- A saída de dados pode se dar: em tela; em uma impressora ou em arquivos.

## Saída de dados em Python

- `print()`
- `open(,w)`.

## 5. Exemplos de programas

## Soma dois números em Python: estruturado

```
# Entrada de dados  
x = int(input('Informe o primeiro número'))  
y = int(input('Informe o segundo número'))  
  
# Processamento de dados  
soma = x + y  
  
# Saída de dados  
print('A soma dos números eh igual a: ', soma)
```

## Cálculo da média

```
# Entrada de dados  
soma = 0  
i = 0  
x = [1, 7, 15, 31]  
# Processamento de dados  
while i < len(x):  
    soma = soma + x[i]  
    i = i + 1  
    media = soma/len(x)  
#Saída de dados  
print('A media dos numeros eh igual a:', media)
```

## 6. Paradigmas de Programação em Python

## Paradigmas de programação: definição

### Definição

Um paradigma de programação é um conjunto de regras definidos por uma linguagem de programação para organização do código fonte de um programa, tendo em vista as funcionalidades oferecidas pela linguagem.

# Paradigmas de programação em Python

A linguagem Python suporta os seguintes paradigmas de programação total ou parcialmente:

- estruturada;
- procedural;
- modular;
- orientação a objetos, e
- funcional.

## 6.1. Programação estruturada

# Programação estruturada

## Definição

É o paradigma de programação onde os comandos que compõem o código estão ordenados sequencialmente com repetições de trechos do código ou decisões sobre o fluxo de execução (isto é, utiliza apenas as estruturas de controle de Bohm e Jacopini).

Características:

- A sequência de execução do programa nunca é quebrada. Por isso, não se utilizam instruções que possam provocar a quebra nesta sequência de execução.
- É o paradigma de programação trabalhado até agora.

## 6.2. Programação procedural

# Funções

## Definição

Uma função em Python é - tal qual na matemática - uma regra que fornece um valor de saída a partir valores de entrada.

## Funções (cont.)

### Usos

- *Servem para armazenar trechos de código que serão reutilizados em diversas partes do mesmo.*
- *Facilitam a leitura do programa.*
- *Evita a repetição do código da mesma em diversas partes do programa.*
- *Implementa o princípio de programação denominado pela sigla em inglês DRY ("Don't Repeat Yourself")*

## Escrevendo uma função em Python: sintaxe

```
def f(x, y):  
    z = f(x, y)  
    return z
```

## Escrevendo uma função em Python: exemplos

- Função que soma os valores da sua entrada

```
def soma(a, b):  
    soma = a + b  
    return soma
```

- Função que soma o produto dos valores des entrada

```
def somaproduto(x, y, z, w):  
    somaproduto = x*y + z*w  
    return somaproduto
```

## Soma dois números em Python: procedural

```
# Definicao de funcoes  
def soma(x,y):  
    soma = x + y  
    return soma  
# Entrada de dados  
a = int(input( 'Informe o primeiro numero' ))  
b = int(input( 'Informe o segundo numero' ))  
# Processamento de dados  
z = soma(a,b)  
# Saida de dados  
print( 'A soma dos numeros eh igual a: ', z)
```

## 6.3. Programação modular

# Programação modular em Python

## Definição

É um paradigma de programação voltado para escrita de programas grandes onde o código fonte, além de ser estruturado, é dividido em diversos arquivos (denominados módulos), que guardam cada um deles funções correlatas entre si.

## Programação modular em Python (cont.)

### Proposição (Características)

- *É um aprofundamento do paradigma de programação procedural.*
- *O algoritmo de solução do problema posto é escrito em um arquivo principal (denominado módulo principal) que chama as funções definidas nos demais módulos.*
- *Facilita ainda mais a leitura, entendimento e manutenção do código.*
- *Cada módulo é chamado pelo módulo principal pelo comando `import nomedoarquivo`.*

## Módulos

- Módulo em Python é qualquer arquivo com extensão `.py`.
- Módulos servem para dividir um código grande em vários arquivos pequenos a fim de tornar o código do programa mais inteligível e, ao mesmo tempo, facilitar adaptações e correções posteriores (manutenção). Um programa escrito em diversos módulos é chamado de modularizado.
- Para se incluir um módulo em um programa em python, basta chamá-lo por meio do comando `import NomeDoModulo` no início do programa principal.

## Soma dois números em Python: módulo de funções

```
# arquivo modsoma.py  
# Definição de funções  
def soma(x,y):  
    soma = x + y  
return soma
```

## Soma dois números em Python: módulo principal

```
# Importacao de modulos  
import modsoma  
  
# Entrada de dados  
a = int(input( 'Informe o primeiro numero' ))  
b = int(input( 'Informe o segundo numero' ))  
  
# Processamento de dados  
z = soma(a , b)  
  
# Saida de dados  
print( 'A soma dos numeros eh igual a: ', z)
```

## Pacotes

- Pacotes são simplifadamente módulos que devem ser instalados no interpretador Python. Tipicamente pacotes são mais rápidos que módulos.
- A escrita de pacotes e sua compilação está fora do escopo dessa introdução ao python.

## 7. Estilo de escrita de código

## Estilo de Escrita de Código em Python

- Inclusão de comentários (usando o símbolo `#`) com o nome do programa, o autor e a descrição do programa.
- Se o comentário possuir várias linhas, use três aspas simples antes do comentário e depois que terminá-lo.
- Se necessário, carregar módulos com a instrução `import nomedopacote`
- Definição de funções
- Declaração de variáveis em tipos de dados válidos

## Estilo de Escrita de Código em Python (cont)

- Entrada de dados
- Algoritmos (sequencia, repetição e decisão)
- Saída de dados
- O arquivo do programa deve ter extensão .R
- O estilo oficial de escrita de de código Python, usado em equipes profissionais de desenvolvimento de software é dado por Style Guide for Python Code.

## 8. Exercícios

## Menezes (2014, pp.71, 77)

- 1 Faça um programa que peça dois números inteiros e ponha na tela a soma deles.
- 2 Faça um programa que converta leia um valor em metros e o converta para milímetros
- 3 Faça um programa que pergunte a velocidade do carro do usuários. Caso ultrapasse 80km/h, exiba uma mensagem dizendo que o usuário foi multado. Nesse caso, exiba o valor da multa, cobrando R \$ 5,00 por km acima de 80km/h.

## Outros exercícios

- 1 Faça um programa que peça os coeficiente a, b e c de um polinômio de segundo grau e determine as raízes deste polinômio.
- 2 Faça um programa que converta leia um valor em metros e o converta para milímetros
- 3 Faça um programa que pergunte a velocidade do carro do usuário. Caso ultrapasse 120km/h, exiba uma mensagem dizendo que o usuário foi multado. Nesse caso, exiba o valor da multa, cobrando R \$ 5,00 por km acima de 120km/h.

## Referências

- Curso de Python
- Python Software Foundation
- Python for Finance
- Quantitative Economics
- **MENEZES**, Nilo Ney C. *Introdução à programação com PYTHON: algoritmos e lógica de programação para iniciantes*, 2a. Ed. São Paulo: NOVATEC, 2014
- **FOROUZAN**, B. e **MOSHARRAF**, F. *Fundamentos da Ciência da Computação. Tradução da 2a. ed. internacional*. São Paulo: CENGAGE Learning, 2011.
- Simpósio de Ciência da computação e Teoria Econômica