



INTRODUÇÃO À PROGRAMAÇÃO COM A LINGUAGEM PYTHON

PARTE I: INTRODUÇÃO

FABRÍCIO G. M. DE CARVALHO

Atualização: Junho 2019

1

SOBRE O MINISTRANTE



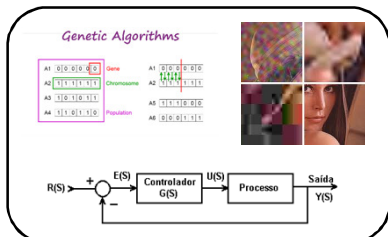
Prof. Fabrício Galende Marques de Carvalho

Formação:

Graduação em Engenharia Elétrica e em Análise e Desenvolvimento de Sistemas;
Mestrado em Engenharia Elétrica – Computação Aplicada;
Doutorado em Engenharia Eletrônica e Computação – Sistemas e Controle.

Áreas de Interesse:

Simulação de Sistemas;
Processamento de Imagens;
Otimização de Sistemas;
Engenharia de Software;
Segurança da Informação.



2

OBJETIVOS

- ✓ Apresentar conceitos básicos necessários ao desenvolvimento de programas computacionais.
- ✓ Apresentar os elementos básicos da linguagem de programação Python de modo a tornar o aluno apto a desenvolver programas simples e iniciar rapidamente o processo de aprendizado vinculado a algoritmos, estruturas de dados, etc.

3

ESTRUTURA DO CURSO

PARTE I: INTRODUÇÃO

- 1. INTRODUÇÃO À PROGRAMAÇÃO DE COMPUTADORES E À LÓGICA DE PROGRAMAÇÃO**
 - a) ALGORITMOS
 - b) BITS E BYTES, A LINGUAGEM FALADA PELOS COMPUTADORES
 - c) LINGUAGENS DE PROGRAMAÇÃO
- 2. CARACTERÍSTICAS DA LINGUAGEM PYTHON**
- 3. FERRAMENTAL NECESSÁRIO À CRIAÇÃO DE PROGRAMAS**

4

ESTRUTURA DO CURSO

PARTE II: PYTHON FUNDAMENTAL

4. DESENVOLVIMENTO DE PROGRAMAS SIMPLES

- a) ESTRUTURA BÁSICA DE UM PROGRAMA
- b) SAÍDA DE DADOS
- c) VARIÁVEIS E TIPOS
- d) ENTRADA DE DADOS
- e) ESTRUTURAS DE CONTROLE
- f) ESTRUTURAS DE REPETIÇÃO

5

ESTRUTURA DO CURSO

PARTE III: BIBLIOTECAS MÓDULOS E ORIENTAÇÃO A OBJETOS

5. BIBLIOTECAS E MÓDULOS

- a) Criando módulos
- b) Importando e fazendo uso de bibliotecas

6. ORIENTAÇÃO A OBJETOS

- a) Definindo classes e instanciando objetos

6

SUGESTÕES PARA LEITURA



BÁSICO - INTERMEDIÁRIO

AVANÇADO

7

1. INTRODUÇÃO À PROGRAMAÇÃO DE COMPUTADORES E LÓGICA DE PROGRAMAÇÃO

a) ALGORITMOS

- ✓ Em linhas gerais, um algoritmo pode ser descrito como uma sequência de passos a serem executados de modo a se resolver um problema.

Exemplo: Na preparação de um prato ou de uma bebida, em uma cozinha, é comum que sejam utilizados algoritmos que são expressos muitas vezes através de “receitas” ou instruções de preparação.

8

1. INTRODUÇÃO À PROGRAMAÇÃO DE COMPUTADORES E LÓGICA DE PROGRAMAÇÃO

a) ALGORITMOS (cont).

O vídeo mostrado no próximo slide, ilustra um algoritmo para se preparar um “shake” de proteínas e multivitamínico.

1. Acrescentar o “pó” do shake em um copo.
2. Acrescentar água.
3. Servir.

Esse algoritmo está correto? Qual o problema?

9

1. INTRODUÇÃO À PROGRAMAÇÃO DE COMPUTADORES E LÓGICA DE PROGRAMAÇÃO

a) ALGORITMOS (cont).



10

1. INTRODUÇÃO À PROGRAMAÇÃO DE COMPUTADORES E LÓGICA DE PROGRAMAÇÃO

a) ALGORITMOS (cont).

Algoritmo de preparação do “shake” corrigido.



11

1. INTRODUÇÃO À PROGRAMAÇÃO DE COMPUTADORES E LÓGICA DE PROGRAMAÇÃO

a) ALGORITMOS (cont).

- ✔ Conforme pode ser observado, o detalhamento dos passos foi alterado. Há mais elementos envolvidos e a sequência, desde a seleção inicial dos ingredientes até a sua entrega final, foi alterada de modo a evitar equívocos por parte de quem executará o algoritmo (nesse caso uma pessoa).

12

1. INTRODUÇÃO À PROGRAMAÇÃO DE COMPUTADORES E LÓGICA DE PROGRAMAÇÃO

a) ALGORITMOS (cont).

Algoritmo “corrigido” para o “shake” de proteínas.

1. Separar o multivitamínico, o <i>whey protein</i> e 350mL de água.	5. Ligar o liquidificador no modo contínuo, e misturar por 30 ou 40 segundos.
2. Em um copo de liquidificador, acrescentar 2 colheres de sopa do “pó” multivitamínico.	6. Desligar o liquidificador e, a seguir, abrir e retirar a tampa.
3. No mesmo copo de liquidificador, juntar 30g (2 medidas) de <i>whey protein</i> .	7. Derramar o “shake pronto” em um copo vazio. Parar antes que o copo transborde.
4. No mesmo copo de liquidificador, acrescentar os 350mL de água. A seguir, tampar o liquidificador.	6. Servir o “shake” pronto que foi colocado no copo.

13

1. INTRODUÇÃO À PROGRAMAÇÃO DE COMPUTADORES E LÓGICA DE PROGRAMAÇÃO

a) ALGORITMOS (cont).

Conclusões:

- ✔ Um algoritmo funciona tal como uma receita ou instrução de preparo. Os ingredientes, incluindo suas quantidades, devem ser bem especificados. Todos os passos, desde a inicialização até o término, devem ser detalhados.
- ✔ Para o caso de algoritmos computacionais, o rigor deve ser ainda maior, dado que a máquina que o executará não é dotada de habilidades tais como as dos seres humanos, não podendo inferir plenamente, realizar analogias, etc.

14

1. INTRODUÇÃO À PROGRAMAÇÃO DE COMPUTADORES E LÓGICA DE PROGRAMAÇÃO

b) BITS E BYTES, A LINGUAGEM FALADA PELOS COMPUTADORES



15

1. INTRODUÇÃO À PROGRAMAÇÃO DE COMPUTADORES E LÓGICA DE PROGRAMAÇÃO

b) BITS E BYTES, A LINGUAGEM FALADA PELOS COMPUTADORES

- ✓ Diferentemente dos seres humanos, as máquinas entendem exclusivamente sinais elétricos. Mais especificamente, no caso de computadores digitais, entendem sua ausência (0's) ou sua presença (1's).
- ✓ Os bits, representados por 0 ou 1, são os símbolos básicos que são “compreendidos” pelos computadores digitais.
- ✓ Conjuntos de 8 bits, denominados de bytes, ou seus múltiplos são tipicamente utilizados como o “blocos básicos” de comunicação com os computadores.

16

1. INTRODUÇÃO À PROGRAMAÇÃO DE COMPUTADORES E LÓGICA DE PROGRAMAÇÃO

b) BITS E BYTES, A LINGUAGEM FALADA PELOS COMPUTADORES

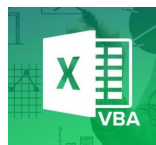
Conclusões:

- ✓ Programar computadores exige que algoritmos computacionais sejam expressos em bits/bytes.
- ✓ Sem a utilização de ferramentas apropriadas, o esforço necessário à programação de computadores seria gigantesco.
- ✓ Solução? Linguagens de Programação.

17

1. INTRODUÇÃO À PROGRAMAÇÃO DE COMPUTADORES E LÓGICA DE PROGRAMAÇÃO

c) LINGUAGENS DE PROGRAMAÇÃO



18

1. INTRODUÇÃO À PROGRAMAÇÃO DE COMPUTADORES E LÓGICA DE PROGRAMAÇÃO

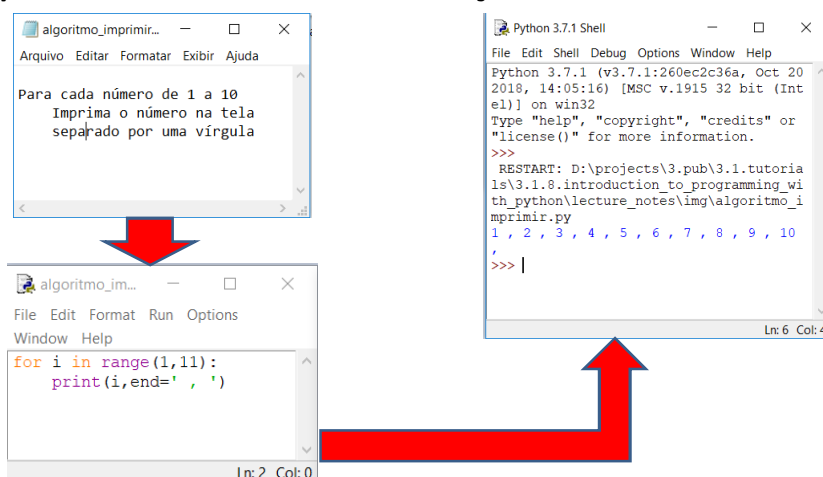
c) LINGUAGENS DE PROGRAMAÇÃO

- ✓ Uma linguagem de programação (LP) é uma tecnologia utilizada para facilitar a conversão de algoritmos compreendidos por seres humanos em algoritmos que podem ser executados por uma máquina (programas de computador).

19

1. INTRODUÇÃO À PROGRAMAÇÃO DE COMPUTADORES E LÓGICA DE PROGRAMAÇÃO

c) LINGUAGENS DE PROGRAMAÇÃO

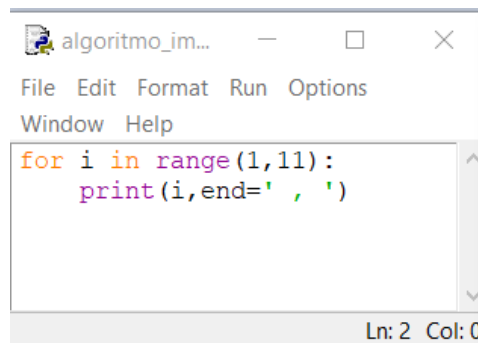


20

1. INTRODUÇÃO À PROGRAMAÇÃO DE COMPUTADORES E LÓGICA DE PROGRAMAÇÃO

c) LINGUAGENS DE PROGRAMAÇÃO

- ✓ Regras de escrita em uma LP: Sintaxe.
- ✓ Sentido dos elementos da LP: Semântica.



```

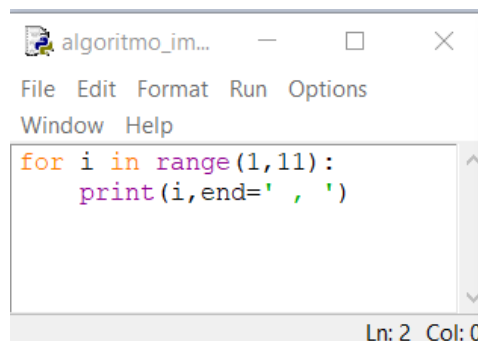
algoritmo_im...
File Edit Format Run Options
Window Help
for i in range(1,11):
    print(i,end=' , ')
Ln: 2 Col: 0
  
```

21

1. INTRODUÇÃO À PROGRAMAÇÃO DE COMPUTADORES E LÓGICA DE PROGRAMAÇÃO

c) LINGUAGENS DE PROGRAMAÇÃO

- ✓ Regras de escrita em uma LP: Sintaxe.
- ✓ Sentido dos elementos da LP: Semântica.



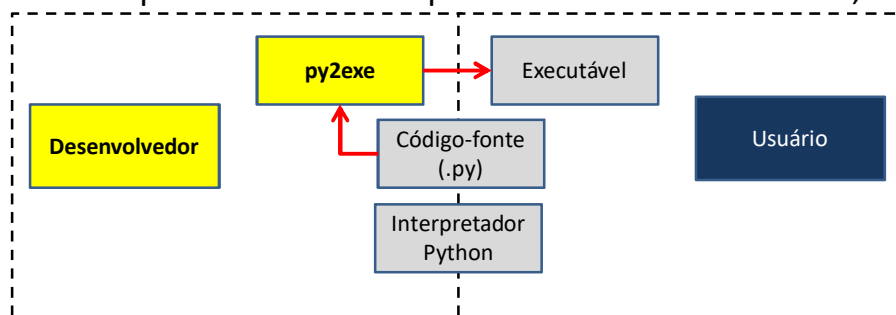
```

algoritmo_im...
File Edit Format Run Options
Window Help
for i in range(1,11):
    print(i,end=' , ')
Ln: 2 Col: 0
  
```

22

2. CARACTERÍSTICAS DA LINGUAGEM PYTHON

- ✓ Foi criada por volta de 1991 por Guido van Rossum (holandês);
- ✓ Python é uma linguagem de programação interpretada mas pode também gerar executáveis independentes do interpretador do lado do usuário;



23

2. CARACTERÍSTICAS DA LINGUAGEM PYTHON

- ✓ Possui sintaxe simplificada e de alto nível (requer inglês básico).
- ✓ É uma linguagem multiparadigma (vários estilos de programação), fornecendo a possibilidade de desenvolvimento orientado a objetos, imperativo (i.e., “procedural”), etc.
- ✓ É uma linguagem que não faz checagem estática de tipo.

24

3. FERRAMENTAL NECESSÁRIO À CRIAÇÃO DE PROGRAMAS

- ✔ Para desenvolver programas em linguagem Python, é necessário somente um editor de texto e um interpretador (ambos disponíveis de modo gratuito).
- ✔ Depois de criar o código-fonte, para executar a aplicação basta chamar o interpretador Python seguido do nome do arquivo principal a ser executado.
- ✔ A seguir, mostra-se como pode ser feito um teste para verificar a disponibilidade do ambiente de desenvolvimento e como pode ser executado um programa no modo interativo e no modo de execução de arquivo Python.

25

3. FERRAMENTAL NECESSÁRIO À CRIAÇÃO DE PROGRAMAS

Testando se o interpretador está instalado:

```
E:\PRJ-OPEN\3.pub\3.1.TUT\3.1.4\working\source\ch_01>python --version
Python 3.5.2
E:\PRJ-OPEN\3.pub\3.1.TUT\3.1.4\working\source\ch_01>
```

Executando o Python no modo interativo:

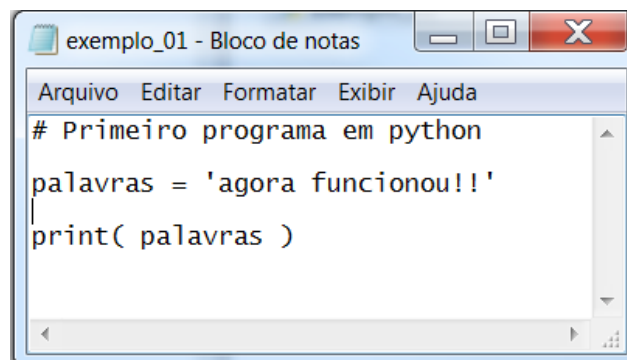
```
E:\PRJ-OPEN\3.pub\3.1.TUT\3.1.4\working\source\ch_01>python
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:18:55) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Meu nome é Fabrício G. M. de Carvalho')
Meu nome é Fabrício G. M. de Carvalho
>>> exit()
E:\PRJ-OPEN\3.pub\3.1.TUT\3.1.4\working\source\ch_01>
```

26

3. FERRAMENTAL NECESSÁRIO À CRIAÇÃO DE PROGRAMAS

Executando um arquivo Python:

Criar o seguinte arquivo (extensão .py)



27

3. FERRAMENTAL NECESSÁRIO À CRIAÇÃO DE PROGRAMAS

Executar o seguinte comando na mesma pasta onde o arquivo criado está localizado:

```
E:\PRJ-OPEN\3.pub\3.1.TUT\3.1.4\working\source\ch_01>python exemplo_01.py
agora funcionou!!

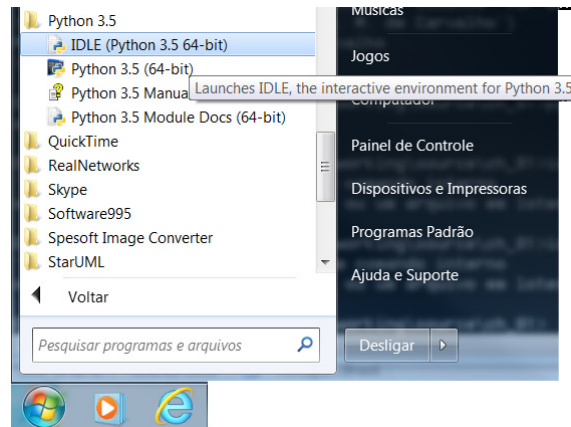
E:\PRJ-OPEN\3.pub\3.1.TUT\3.1.4\working\source\ch_01>
```

- ✓ Notar que agora o interpretador Python apenas executou o arquivo e depois terminou.

28

3. FERRAMENTAL NECESSÁRIO À CRIAÇÃO DE PROGRAMAS

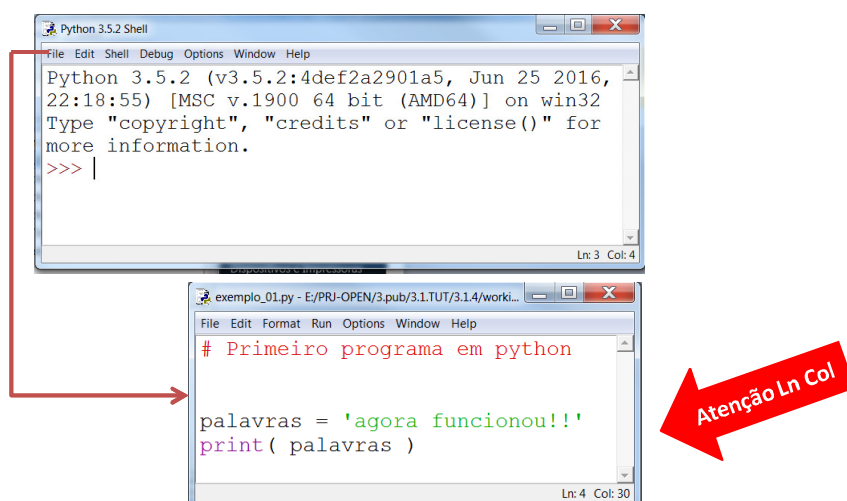
- ✓ Ambiente alternativo para criação de arquivos .py e para execução no modo interativo: Idle.



29

3. FERRAMENTAL NECESSÁRIO À CRIAÇÃO DE PROGRAMAS

Shell do Python e Editor:



30

FIM DA PARTE 1!!
VAMOS AOS
EXERCÍCIOS!