

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Nova Cruz



Fundamentos de Lógica e Algoritmos

Aula 2.6 – Introdução a Programação

Prof. Dr. Bruno Moreno
bruno.moreno@ifrn.edu.br

O objetivo dessa disciplina é aprender uma linguagem de programação??

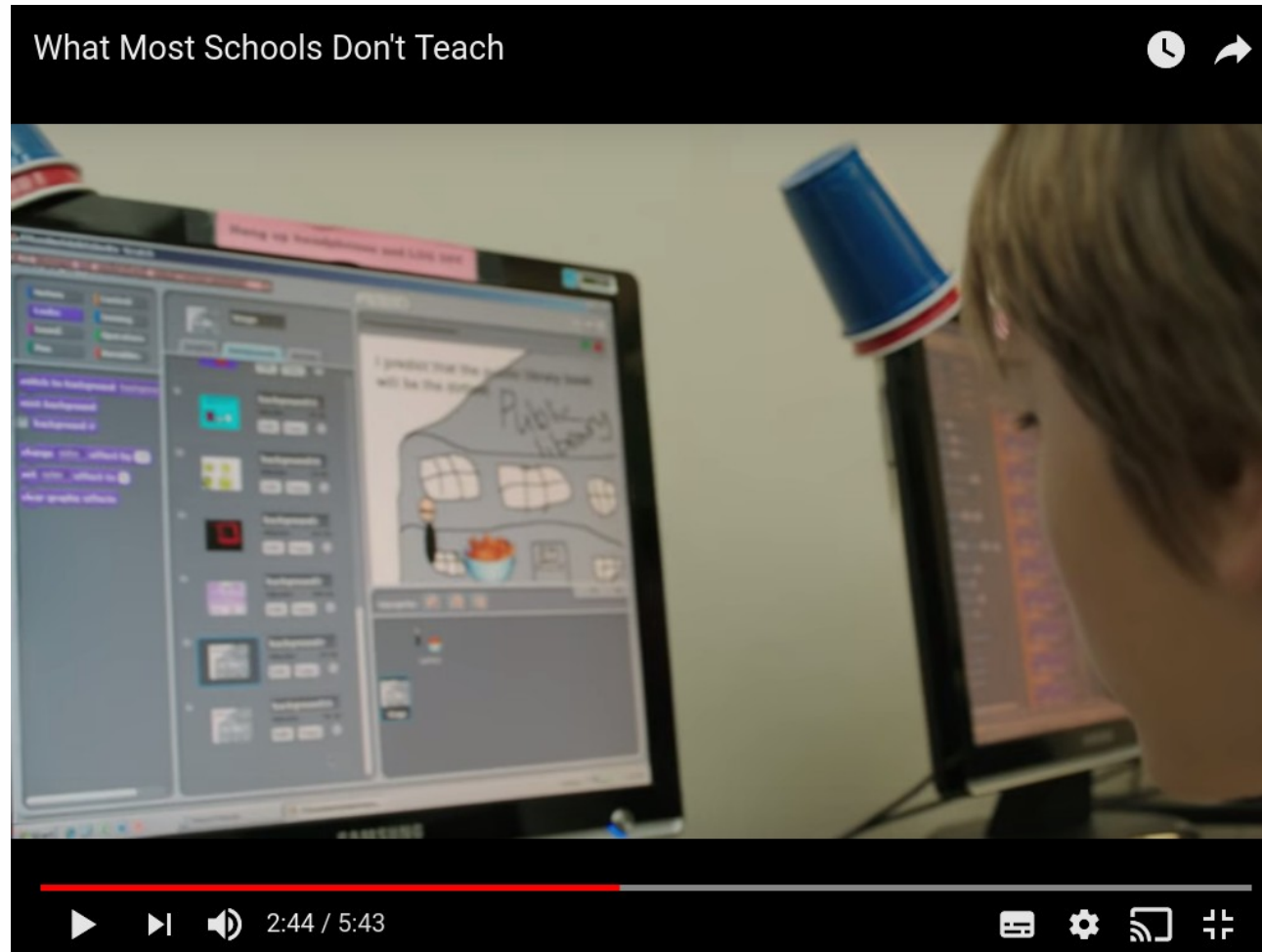


Objetivo da disciplina

- O objetivo da disciplina é ensinar a programar independentemente da linguagem;
- Aqui, Python será utilizado como ferramenta para esse aprendizado;
- O mais importante é aprender **lógica de programação**.

Aprender a programar!

Video do Code.org



<https://www.youtube.com/watch?v=nKlu9yen5nc>

Por que Python?



PÁI-TON!

Sobre Python























- Criada pelo holandês Guido van Rossum em 1989 e lançada em 1991;
- Suporta vários paradigmas de programação:
 - Programação Procedural/Estruturada
 - Programação Orientada a Objetos

Por que Python?

- É uma linguagem simples, versátil, clara e objetiva;
- É poderosa:
 - Desenvolvimento web;
 - Sistemas grandes e complexos;
- É software livre
 - Python Foundation (<http://www.python.org/>);
 - Possui inúmeros colaboradores.
- É multiplataforma: Linux, Microsoft Windows ou Mac;
- É popular

Por que Python?

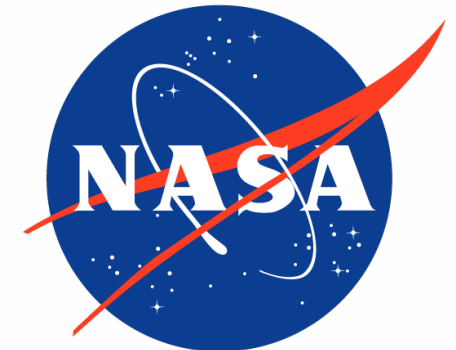
- Ranking (2017):

Language Rank	Types	Spectrum Ranking
1. Python	 	100.0
2. C	  	99.7
3. Java	  	99.5
4. C++	  	97.1
5. C#	  	87.7
6. R		87.7
7. JavaScript	 	85.6
8. PHP		81.2
9. Go	 	75.1
10. Swift	 	73.7

<https://spectrum.ieee.org/static/interactive-the-top-programming-languages-2017>

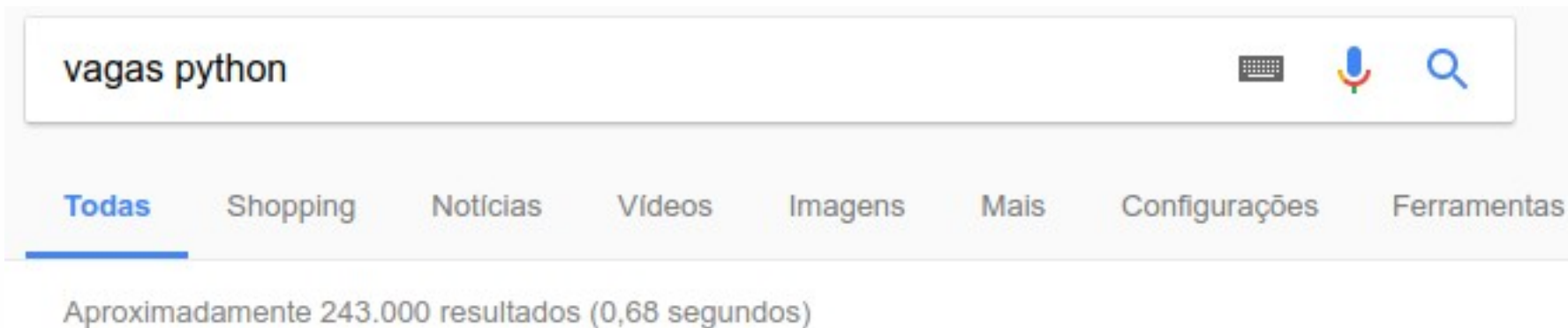
Por que Python?

- Algumas empresas:



Por que Python?

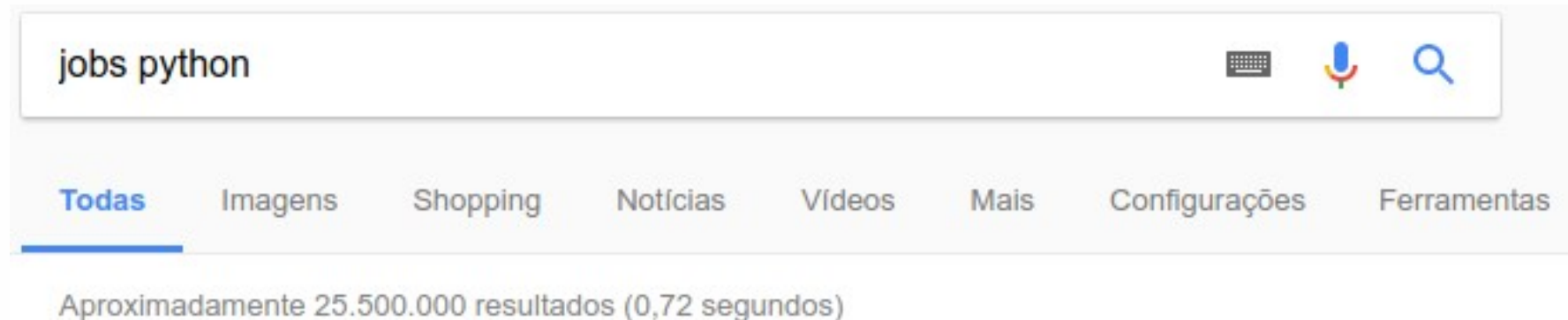
- Vagas de emprego:



vagas python

Todas Shopping Notícias Vídeos Imagens Mais Configurações Ferramentas

Aproximadamente 243.000 resultados (0,68 segundos)



jobs python

Todas Imagens Shopping Notícias Vídeos Mais Configurações Ferramentas

Aproximadamente 25.500.000 resultados (0,72 segundos)

Interpretador Python

- Diferente de outras linguagens interpretadas, o **interpretador Python é iterativo**
 - É possível executá-lo sem fornecer um programa;
 - Se resume em uma tela onde é possível executar comandos específicos e ver seus efeitos;
 - O interpretador é representado pelo sinal “>>>” antes dos comandos apresentados;
 - É muito útil para testar comandos e aprender mais sobre a linguagem.

Interpretador Python



- No Linux, o Python já vem pré-instalado:

```
bruno@bruno-Aspire-V3-571: ~  
File Edit View Search Terminal Help  
bruno@bruno-Aspire-V3-571:~$ python3  
Python 3.5.2 (default, Sep 14 2017, 22:51:06)  
[GCC 5.4.0 20160609] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> █
```

Instalando Python



- No Windows:
 - É necessário instalar o **interpretador** Python
- Como Python é software livre, pode ser baixado em www.python.org:

The screenshot shows the Python.org website in a Firefox browser. The main content area features a code editor on the left with the following code:

```
# Write Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610
```

On the right, the 'Functions Defined' section states: "The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. [More about defining functions](#)".

At the bottom of the page, there are four main navigation links:

- Get Started**: Whether you're new to programming or an experienced developer, it's easy to learn and use Python. Start with our [Beginner's Guide](#).
- Download**: Python source code and installers are available for download for all versions! Not sure which version to use? [Check here](#). Latest: Python 2.7.6 - [Python 3.3.4](#).
- Docs**: Documentation for Python's standard library, along with tutorials and guides, are available online. docs.python.org.
- Jobs**: Looking for work or have a Python related position that you're trying to hire for? Our community-run job board is the place to go. jobs.python.org.

Interpretador Python



- O interpretador Python permite que sejam executados comandos **iterativamente**;
- Por exemplo, digite:
 - `print("Olá Mundo")`
 - Observe o resultado!
- `print` é uma comando de Python chamado **função**;
- Toda **função** em Python é seguida por um par de **parênteses**;
- A função **print()** imprime o que estive dentro dos parênteses na tela
 - Essa função lembra a função **escreva** vista em pseudocódigo.

Interpretador Python



- O interpretador Python permite que sejam executados comandos **iterativamente**;
 - Por exemplo, digite:
 - `print("Olá Mundo")`
 - Observe o resultado!
- “Olá Mundo” é composto por duas palavras;
 - Cada palavra é composta por um conjunto de caracteres;
 - Em Python (e em outras linguagens também), um conjunto de caracteres é chamado de **string**;
 - Em Python, uma string **tem que ser delimitada** por **aspas simples** ou **aspas duplas**;

Interpretador Python



```
bruno@bruno-Aspire-V3-571: ~  
File Edit View Search Terminal Help  
bruno@bruno-Aspire-V3-571:~$ python  
Python 2.7.12 (default, Nov 19 2016, 06:48:10)  
[GCC 5.4.0 20160609] on linux2  
Type "help", "copyright", "credits" or "license" for more information.  
>>> print("Olá Mundo")  
Olá Mundo  
>>> exit() → Outra função de Python!  
bruno@bruno-Aspire-V3-571:~$
```

- Uma série de comandos pode ser utilizado no interpretador;
- Programas grandes ou mais complexos precisarão de outros recursos para serem executados.

Um programa em Python



- Programas escritos em Python armazenam código, ou seja, algoritmos escritos em Python;
- Esses programas são arquivos do tipo **py**
 - Por exemplo:
 - aloMundo.py
 - suap.py
- Para escrever códigos-fonte, utilizaremos (por enquanto) um editor de texto mais simples
 - Bloco de Notas, no Windows;
 - Gedit, no Linux;

Um programa em Python

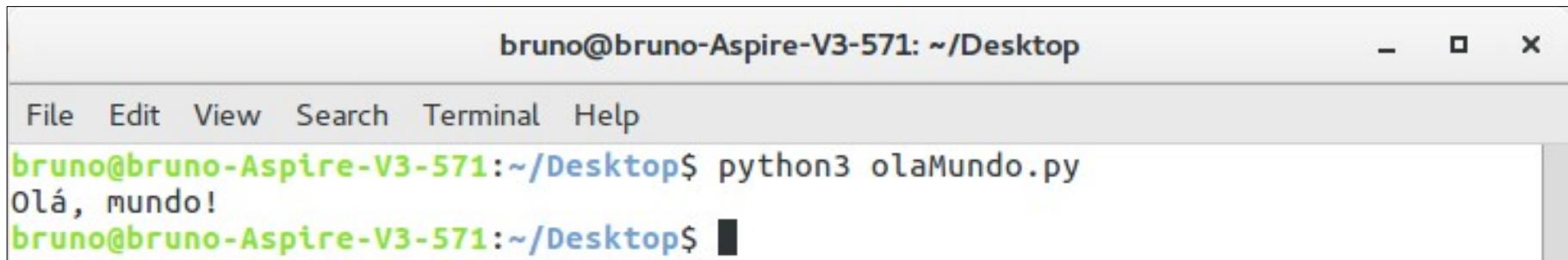


- Crie um programa tal qual o exemplo a seguir:



```
olaMundo.py
~/Desktop
Save
Python Tab Width: 8 Ln 1, Col 21 INS
print("Olá, mundo!")
```

- Execute o programa e observe o resultado:



```
bruno@bruno-Aspire-V3-571: ~/Desktop
File Edit View Search Terminal Help
bruno@bruno-Aspire-V3-571:~/Desktop$ python3 olaMundo.py
Olá, mundo!
bruno@bruno-Aspire-V3-571:~/Desktop$
```

1^{os} comandos de programação

Trabalhando com Python

ANATOMIA DO PROGRAMADOR

informaticus programatorum

Óculos estilo "Twilight Zone" necessários devido aos dias passados em frente ao monitor.

Barba e cabelos desleixados - não tem tempo para cuidar da pelagem.

Corpo com grande concentração de lipídios. Consome até 3 vezes o próprio peso em comida tipo congelada.

Camisa do torneio de Halo - o programador típico participa de, em média, 15 torneios de Halo ou outro FPS por ano.

Pelagem rala ou inexistente em certas regiões, exceto pela cabeça e rosto. No entanto, sabe-se que são mamíferos.

Pernas frágeis. Como sua locomoção é nula, este animal não as utiliza com frequência.

Mouse de estimação. O programador não tem tempo para animais vivos então adota os objetos queridos.



Recondição 20/5/02



Função de saída

- Saída do sistema (ou *output*) é toda informação que o sistema fornece para outras fontes
 - Uma mensagem na tela
 - Sons
 - Arquivos
 - Impressões
- A função **mais básica** de saída de Python é a função **print()**

Função `print` ()



- Digite:

```
>>> print(Olá Mundo)
```

- O interpretador entende que mensagens (strings) não podem ser tratadas sem aspas;

- Digite:

```
>>> print("Olá Mundo")
```

- Observe que Python pode tratar strings entre aspas duplas e aspas simples:

```
>>> print('Olá Mundo')
```


Função `print` ()



- Python é **case sensitive**

- Ou seja, letras maiúsculas são diferenciadas de minúsculas;
- Exemplo: digite o nome da função `print` com **Print** e observe o erro:

```
>>> Print("Olá Mundo")
```

Função `print` ()



- Digite:
 `>>> print(7+4)`
 - Por que não `7+4` não foi passado entre aspas?
- Se qualquer conjunto de caracteres estiver entre aspas, é uma string
- Se o conjunto de caracteres não estiver entre aspas e representar uma operação matemática, Python resolverá a operação

Função `print` ()



- Digite:

```
>>> print("7+4")
```

- Lembre-se: pode-se utilizar aspas duplas ou simples;

- Concatenação de Strings:

```
>>> print("7"+"4")
```

- O interpretador entende que 7 é uma string e que 4 também;

- O sinal + entre strings representa a operação de **concatenação de strings**

- Ou seja, as palavras são conectadas entre si;

1^{os} comandos no shell



- Como estamos trabalhando diretamente no shell, não é necessário utilizar a função **print**

```
>>> 7+4
```

```
11
```

```
>>> '7'+ '4'
```

```
74
```

```
>>> '7+4'
```

```
7+4
```

Função de entrada

- Entrada do sistema (ou *input*) é toda informação que o sistema recebe de outras fontes
 - Teclado
 - Mouse
 - Arquivos
 - Microfone
- A função mais básica de entrada em Python é a função `input()`
 - Essa função lembra o comando **leia** do pseudocódigo

Função `input()`

- Utilize a função `input` para ler dados do usuário:

```
input("Qual é o seu nome? ")  
input("Qual é sua idade? ")  
input("Qual é o seu peso? ")
```

**E se for necessário
manipular os valores
informados pelo usuário?**

**Para isso, utiliza-se
variáveis!**

Variáveis

- São formas de se armazenar dados para uso posterior;
- Toda variável pode **receber** valores:

```
nome = "Bruno"  
idade = 26  
peso = 60.3
```

nome **recebe** "Bruno"
idade **recebe** 26
peso **recebe** 60.3

- Em Python, variáveis podem ser classificadas em três **tipos básicos**
 - Variável inteira: valores inteiros;
 - Variável de ponto flutuante: valores decimais;
 - Variável do tipo **string**: mensagens;

Variáveis

- Agora, pode-se trabalhar com essas variáveis:

```
nome = input("Qual é o seu nome? ")  
idade = input("Qual é sua idade? ")  
peso = input("Qual é o seu peso? ")  
print(nome, idade, peso)
```

Por que utilizamos **vírgula** para imprimir o nome, a idade e o peso?