



*Lugar
para
você*

Introdução a Python

Altino Dantas

PARTE I



INSTITUTO DE
INFORMÁTICA
UFG



UFG
UNIVERSIDADE
FEDERAL DE GOIÁS

Objetivos



Descobrir como executar um programa Python

Aprender o básico sobre Python – Conceitos iniciais

Conhecer os tipos de dados em Python



1989

Criação do Python – Guido Van Rossum

1991

Classes e Herança

1994

Programação Funcional: lambda, map, filter, reduce

2001

Surgimento da Python Software Foundation

2008

Lançamento do Python 3

2012

Raspberry PI – Influência por Python

Características



Linguagem Orientada a Objetos

Alto Nível

Legível



pythonTM

Interpretada

Pode-se utilizar
Operadores

Tipagem Dinâmica

Suporta Módulos



Obtendo um interpretador

- Por ser uma linguagem interpretada, Python requer um interpretador para execução de um programa nesta linguagem;
- Instalando uma versão do Python localmente e invocando-o via **linha de comando**;
- Instalando uma versão do Python localmente e utilizando-o por uma **IDE**;
- Utilizando gerenciadores de ambientes Python, por exemplo, **Anaconda**;
- Utilizando um interpretador **online**, onde provavelmente haverá algumas limitações.



GPU gratuita

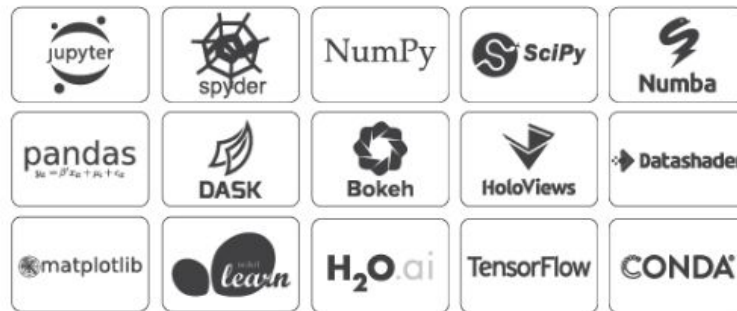


Ambientes online

<http://colab.research.google.com>



ANACONDA®



Ambientes locais

<http://www.anaconda.com/distribution>





O Projeto **Jupyter** desenvolve software de código, padrões e serviços abertos, para computação interativa em diversas linguagens de programação.



Console interativo

- O Python possui um **modo interativo** que permite “conversar” diretamente com o interpretador e definir variáveis, funções e classes;

```
>>> print (“Agora você está no modo interativo  
:.)”)
```

Agora você está no modo interativo :)

```
>>> exit()
```




Identação

- Python separa os blocos de código por identação e (:) para iniciar um novo nível.

```
print ("Hello World!")
nome = input("Qual é a melhor linguagem de programação: ")
if nome == "python":
    print ("Você acertou!")
else:
    print ("Errou! =(")
```



Comentários

- Podemos adicionar **comentários** em nossos códigos. Eles são ignorados pelo Python, mas tornam o código organizado, facilitando sua compreensão.

Comentário de uma Linha

*''' Comentário em
várias Linhas
'''*

```
#coding: utf-8
```

```
'''
```

```
toda informações contida entre 3 aspas SIMPLES  
é considerada como caracteres que devem ser ignorados.
```

```
'''
```



Convenção de estilo

- A Identação deve ser de 4 espaços (Sem Tabs).
- Limite de 79 caracteres por linhas.
- Linhas muito longas são quebradas por \ .
- Alinhar os parênteses em caso de quebra de linha.
- As funções devem estar sempre 2 linhas abaixo da de cima.
- Não usar espaço depois de abrir ou fechar um parênteses.



Convenções para nomes de variáveis

- Nunca inicie o nome de variáveis por números.
- Nunca utilize caracteres especiais, somente o underline (_).
- Nunca utilize espaços em branco.
- Crie variáveis com nomes abreviados.
- Não crie variáveis com nomes sem sentido dentro do código.
- Não utilize letras maiúsculas em variáveis.



Tipos de variáveis

```
inteiro = 1
```

```
decimal = 0.003
```

```
texto = "Big Data"
```

```
booleano = True
```

```
diccionario = {"nome": "Guido", "idade": 28}
```

```
lista = ["item1", "item2", 3, "quatro", 3.14]
```

```
tupla = (1, 2, 3, "Python")
```



Tipos numéricos

- Os tipos numéricos no Python são: números inteiros, números de ponto flutuante, números complexos e booleanos.

Inteiros

```
x = 10
```

```
type (x)
```

Flutuantes

```
x = 3.12
```

```
type (x)
```

Complexos

```
x = 1 - 2j
```

```
type (x)
```

Booleano

```
x = True
```

```
type (x)
```



Tipos de sequência

Os tipos de sequência no Python são: tupla, lista e dicionário.

Tupla

```
>>>x=(“a”,“b”)
>>> print (x)
('a', 'b')
>>> type (x)
<class'tuple'>
```

Lista

```
>>> x=[1,“a”]
>>> print (x)
[1, 'a']
>>> type (x)
<class 'list'>
```

Dicionário

```
>>>x={“num”:10}
>>> print (x)
{'num': 10}
>>> type (x)
<class 'dict'>
```



Tipos de sequência

Os dados textuais no Python são identificados por “strings”.

```
>>> x = “Python”
```

```
>>> print (x)
```

```
Python
```

```
>>> type (x)
```

```
<class 'str'>
```


Exercício

- Obtenha um interpretador Python que possa ser executado na sua máquina;
- Verifique o modo interativo invocando o Python por linha de comando;
- Crie seu primeiro programa Python declarando uma variável de cada tipo visto na aula;
- Atribua valores para as variáveis e imprima-os no console através do comando `print()`.

Obrigado

altinobasilio@inf.ufg.br

Dúvidas ou sugestões?

