

Processando

Sim. Mas, que  
Python é esse?



# Afinal, o que é Python?

- É uma linguagem de programação de alto nível\*
- De Script
- Multi paradigma
- Interpretada
- Imperativa
- Interativa
- Orientada a objetos
- De código-fonte aberto
- Funcional
- Dinamicamente tipada
- Fácil de usar e aprender.
- Roda em Windows, Linux, Mac, Palm, em celulares...

\*mais próximo da linguagem natural.

Linguagem criada em 1991 pelo holandês Guido Van Rossum, sob o ideal de "**Programação de Computadores para todos**".



# Afinal, o que é Python?

- É uma linguagem de programação de alto nível\*
- De Script
- Multi paradigma
- Interpretada
- Imperativa
- Interativa
- Orientada a objetos
- De código-fonte aberto
- Funcional
- Dinamicamente tipada
- Fácil de usar e aprender.
- Roda em Windows, Linux, Mac, Palm, em celulares...

\*mais próximo da linguagem natural.

É um “roteiro” seguidos por sistemas computacionais e trazem informações que são processadas e transformadas em ações efetuadas por um programa principal.



# Afinal, o que é Python?

- É uma linguagem de programação de alto nível\*
- De Script
- Multi paradigma
- Interpretada
- Imperativa
- Interativa
- Orientada a objetos
- De código-fonte aberto
- Funcional
- Dinamicamente tipada
- Fácil de usar e aprender.
- Roda em Windows, Linux, Mac, Palm, em celulares...

\*mais próximo da linguagem natural.

Ex.: Digamos que você precisa de um programa que realize a soma de notas escolares e como elas serão apresentadas. O script é a sequência de passos que o computador vai interpretar para somar e apresentar as notas na tela.



# Afinal, o que é Python?

- É uma linguagem de programação de alto nível\*

- De Script

- Multi paradigma

- Interpretada

- Imperativa

- Interativa

- Orientada a objetos

- De código-fonte aberto

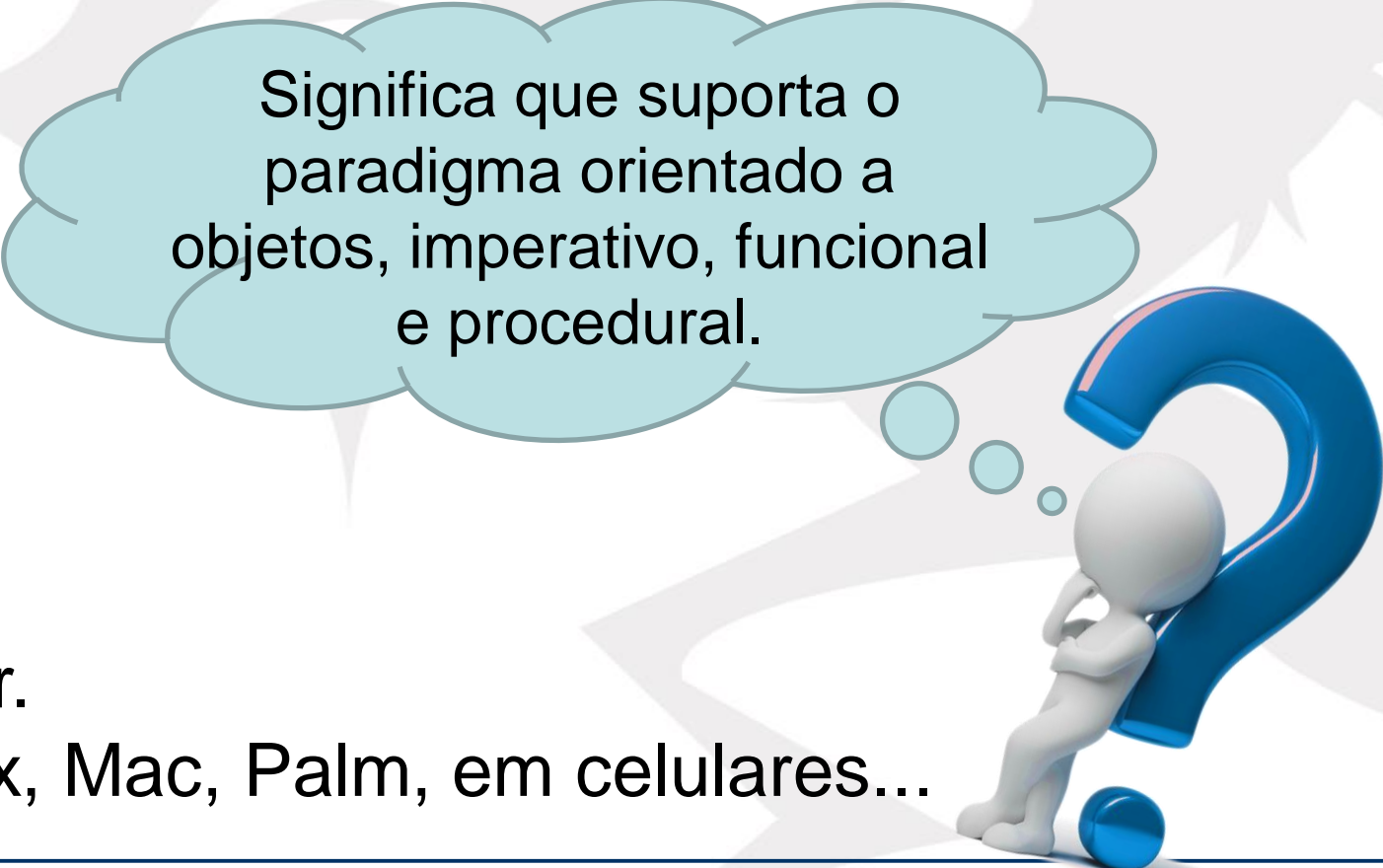
- Funcional

- Dinamicamente tipada

- Fácil de usar e aprender.

- roda em Windows, Linux, Mac, Palm, em celulares...

\*mais próximo da linguagem natural.



Significa que suporta o paradigma orientado a objetos, imperativo, funcional e procedural.



# Afinal, o que é Python?

- É uma linguagem de programação de alto nível\*
- De Script
- Multi paradigma
- Interpretada
- Imperativa
- Interativa
- Orientada a objetos
- De código-fonte aberto
- Funcional
- Dinamicamente tipada
- Fácil de usar e aprender.
- roda em Windows, Linux, Mac, Palm, em celulares...

\*mais próximo da linguagem natural.

Significa dizer que ao se escrever um programa, este não será **compilado** (traduzido para uma linguagem de máquina), mas sim lido por um outro programa (chamado de **interpretador**) que traduzirá para a máquina o que seu programa quer dizer.



# Afinal, o que é Python?

- É uma linguagem de programação de alto nível\*
  - De Script
  - Multi paradigma
  - Interpretada
  - Imperativa
  - Interativa
  - Orientada a objetos
- \*mais próximo da linguagem natural.

Significa dizer que ao se escrever um programa, este não será **compilado** (traduzido para uma linguagem de máquina), mas sim lido por um outro programa (chamado de **interpretador**) que traduzirá para a máquina o que seu programa quer dizer.



# Afinal, o que é Python?

- É uma linguagem de programação de alto nível\*

- De Script

- Multi paradigma

- Interpretada

- Imperativa

- Interativa

- Orientada a objetos

- De código-fonte aberto

- Funcional

- Dinamicamente tipada

- Fácil de usar e aprender.

- roda em Windows, Linux, Mac, Palm, em celulares...

\*mais próximo da linguagem natural.

Significa dizer que ao se escrever um programa, este não será **compilado** (traduzido para uma linguagem de máquina), mas sim lido por um outro programa (chamado de **interpretador**) que traduzirá para a máquina o que seu programa quer dizer.





# Afinal, o que é Python?

- É uma linguagem de programação de alto nível\*
- De Script
- Multi paradigma
- Interpretada
- Imperativa
- Interativa
- Orientada a objetos
- De código-fonte aberto
- Funcional
- Dinamicamente tipada
- Fácil de usar e aprender.
- roda em Windows, Linux, Mac, Palm, em celulares...

\*mais próximo da linguagem natural.

São orientadas a ações, onde a computação é vista como uma sequência de instruções que manipulam valores de variáveis.

**São caracterizadas por 3 conceitos:**

- Variáveis
- Atribuição
- Repetição



# Afinal, o que é Python?


- É uma linguagem de programação de alto nível\*
- De Script
- Multi paradigma
- Interpretada
- Imperativa
- Interativa
- Orientada a objetos
- De código-fonte aberto
- Funcional
- Dinamicamente tipada
- Fácil de usar e aprender.
- roda em Windows, Linux, Mac, Palm, em celulares...

\*mais próximo da linguagem natural.

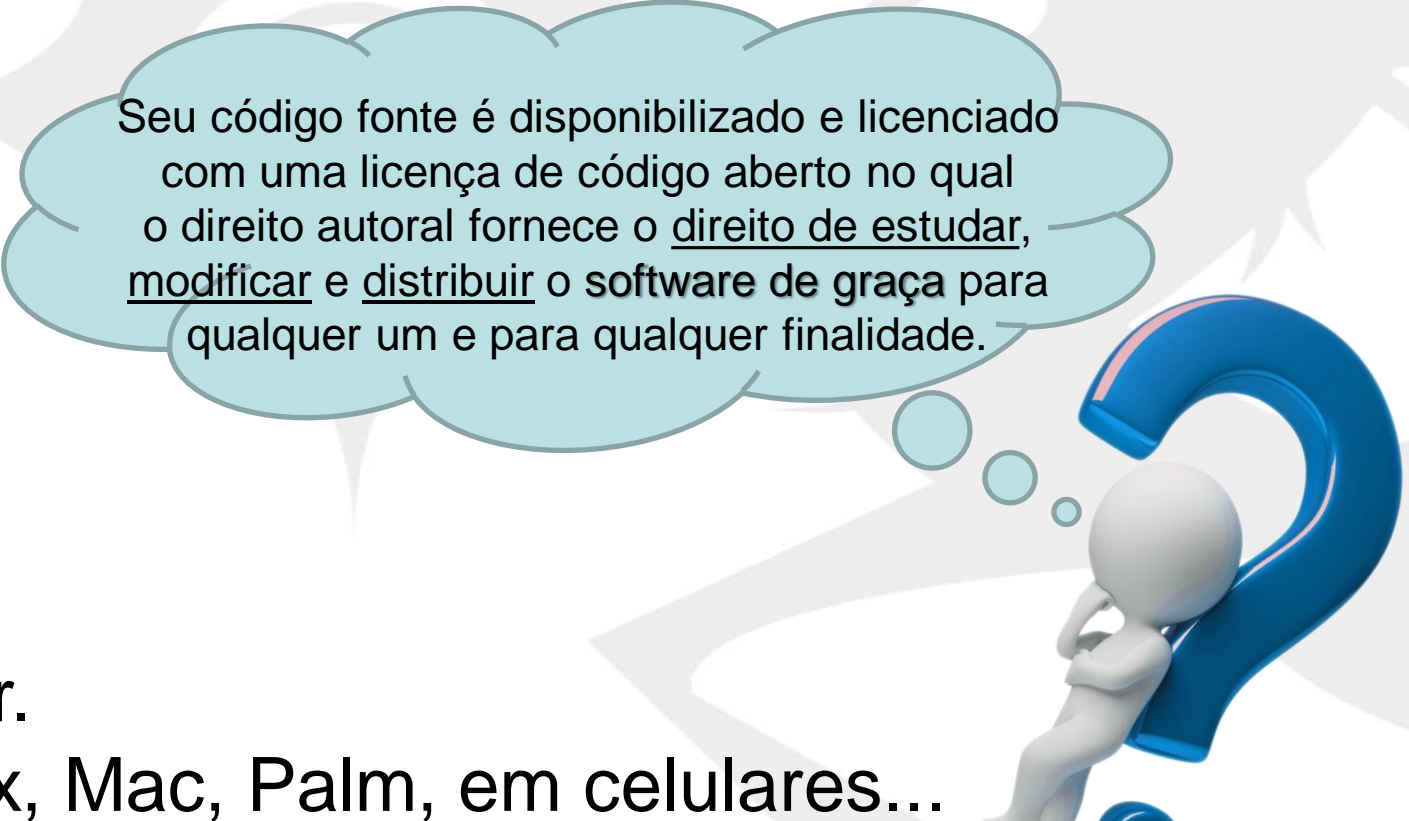
O interpretador disponibilizará uma interface interativa onde é possível inserir os comandos desejados um por um e ver o efeito de cada um deles.



# Afinal, o que é Python?

- É uma linguagem de programação de alto nível\*
  - De Script
  - Multi paradigma
  - Interpretada
  - Imperativa
  - Interativa
  - Orientada a objetos
  - De código-fonte aberto
  - Funcional
  - Dinamicamente tipada
  - Fácil de usar e aprender.
  - roda em Windows, Linux, Mac, Palm, em celulares...
- \*mais próximo da linguagem natural.
- Facilita o controle sobre a estabilidade dos projetos quando estes começam a tomar grandes proporções, permitindo que o usuário programe na forma procedural, se desejar.
- 

# Afinal, o que é Python?

- É uma linguagem de programação de alto nível\*
  - De Script
  - Multi paradigma
  - Interpretada
  - Imperativa
  - Interativa
  - Orientada a objetos
  - De código-fonte aberto
  - Funcional
  - Dinamicamente tipada
  - Fácil de usar e aprender.
  - roda em Windows, Linux, Mac, Palm, em celulares...
- \*mais próximo da linguagem natural.
- Seu código fonte é disponibilizado e licenciado com uma licença de código aberto no qual o direito autoral fornece o direito de estudar, modificar e distribuir o software de graça para qualquer um e para qualquer finalidade.
- 

# Afinal, o que é Python?

- É uma linguagem de programação de alto nível\*
- De Script
- Multi paradigma
- Interpretada
- Imperativa
- Interativa
- Orientada a objetos
- De código-fonte aberto
- Funcional
- Dinamicamente tipada
- Fácil de usar e aprender.
- roda em Windows, Linux, Mac, Palm, em celulares...

\*mais próximo da linguagem natural.

Numa linguagem funcional, tudo, até mesmo o próprio programa, **é uma função.**

A programação funcional modela um problema computacional como uma coleção de funções matemáticas, cada uma com um domínio de entrada e um resultado.



# Afinal, o que é Python?

- É uma linguagem de programação de alto nível\*
- De Script
- Multi paradigma
- Interpretada
- Imperativa
- Interativa
- Orientada a objetos
- De código-fonte aberto
- Funcional
- Dinamicamente tipada
- Fácil de usar e aprender.
- roda em Windows, Linux, Mac, Palm, em celulares...

\*mais próximo da linguagem natural.

Não é necessário **tipar** as variáveis para usá-las. Assim, para atribuirmos o valor 2, que é um inteiro, ao nome valor1, só precisamos digitar valor1=2, que automaticamente será tipado em Python como inteiro.





# Afinal, o que é Python?

- É uma linguagem de programação de alto nível\*
- De Script
- Multi paradigma
- Interpretada
- Imperativa
- Interativa
- Orientada a objetos
- De código-fonte aberto
- Funcional
- Dinamicamente tipada
- Fácil de usar e aprender.
- roda em Windows, Linux, Mac, Palm, em celulares...

\*mais próximo da linguagem natural.

Como não é preciso declarar a variável valor1 antes de utilizá-la, se você resolver atribuir o valor "Boa tarde" a essa variável, ela irá incorporar o tipo como sendo STRING.



# Por que usar Python?

- Conceitos fundamentais simples;
  - Sintaxe clara – muito próxima de um **pseudocódigo**;
  - Código curto e legível;
  - Tipos pré-definidos poderosos;
  - Possui um vasto repertório de bibliotecas;
  - Ciclo de desenvolvimento rápido;
  - Extremamente portátil: Unix/Linux, Windows, Mac, PalmOS, WindowsCE, OS2, PlayStation...
-

# Em que é utilizada essa linguagem?

- Inteligência artificial
  - Banco de dados
  - Biotecnologia
  - Animação 3D
  - Aplicativos móveis
  - Jogos
  - Aplicativos web.
- 



# Motivação para usar essa linguagem...

1) Use Python para **Web e Internet desenvolvimento**, pois é um excelente programa para certas áreas da Web e Internet desenvolvimento. Especificamente, Python é a melhor opção se o seu desenvolvimento exige a criação de scripts básicos CGI, frameworks Django e TurboGears, Zope ou outras soluções high-end e avançar de gerenciamento de conteúdo como o Plone .

---

# Motivação para usar essa linguagem...

2) Use Python para **banco de dados**.

Python é ideal para o costume e as interfaces **ODBC** para bancos de dados como **MySQL, Oracle, MS SQL Server, PostgreSQL e SybODBC**. Ele também é compatível para uso com um banco de dados padrão como **API, ZODB e Duro** .

---

# Motivação para usar essa linguagem...

3) Use Python para e **serviços orientados científicos e numéricos da GUI desktop**. Python é uma linguagem de programação excelente para o desenvolvimento de GUI, pois está incluído na maioria das distribuições binários para Python. Python também é um programa popular para **computação científica e numérico, incluindo bioinformática e física**.

---



# Motivação para usar essa linguagem...

4) Use Python para jogos e gráficos 3D. Muitos programadores usam Python no desenvolvimento de jogos comerciais e de lazer. **PyGame** e **PyKyra** são dois dos quadros mais populares do jogo que Python oferece.

---

# Quem utiliza Python?

## No mundo

- Nasa (vários projetos)
- Ubuntu, Redhat
- Amazon Web Service
- Disney
- Google, Instagram, Pinterest
- Apple, HP, Microsoft
- Netflix, Youtube, Yahoo!
- Nokia
- Dropbox
- Várias universidades como MIT e Stanford

## No Brasil

- Jornal do Brasil, Aol Brasil, Globo.com.
- JusBrasil, SERPRO, Magazine Luiza
- **Embratel**: monitoramento das interfaces de backbone e clientes de internet.
- **CPqD**: monitoramento de centrais telefônicas.
- **Conectiva**: Gerenciamento de pacotes da distribuição Linux e ferramentas de uso interno.
- **Async**: desenvolvimento de software de automação comercial.

O interpretador Python pode ser usado de forma interativa, na qual as linhas de código são digitadas em um **prompt** (linha de comando), semelhante ao **shell** do sistema operacional. Para evocar o modo interativo basta executar o interpretador. Ele estará pronto para receber comandos após o surgimento do sinal de espera “>>>” na tela:

**Shell** é o nome dado aos ambientes **interativos** para execução de comandos, que podem ser usados para testar pequenas porções de código.

Obs.: O modo interativo do Python também é chamado de **IDLE**.

## Curiosidades

A comunidade de usuários de Python criou algumas expressões para se referir aos assuntos relacionados à linguagem. Neste jargão, o termo **Pythonic** é usado para indicar que algo é compatível com as premissas de projeto do Python, e **Unpythonic** significa o oposto. Já o usuário da linguagem é chamado de **Pythonist**.

Obs.: Para verificar se o Python foi bem instalado em seu computador. Abra o prompt do DOS, em Executar , digite cmd, em seguida digite py (de Python).

```
C:\Users\Jacqueline>py
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

# Primeiros passos: Usando o Python como calculadora

Além de poder ver todos os cálculos anteriores, você também pode usar os resultados deles em cálculos futuros, simplesmente copiando e colando os resultados. Os Operadores Algébricos usados no interpretador são os mesmos que você já usava em outros programas do seu sistema operacional: +, -, \*, /. Abaixo segue exemplo de diferentes operadores.

## Exemplo:

```
>>> a=1
```

```
>>> b=2
```

```
>>> a == b    # == testa se a é igual a b
```

```
False
```

```
>>> a != b    # != testa se a é diferente de b
```

```
True
```

```
>>> 2*a >= b   # testa se o dobro de a é maior  
True          ou igual a b
```

```
>>> a > b      # > testa se a é maior que b
```

```
False
```

```
>>> a < b      # < testa se a é menor que b
```

```
True
```

## Tipos de expressões

**Aritméticas** (operadores Numéricos) (+, -, \*, /)

**Relacionais** (op. Lógicos) (>, <, ==, !=, >=, <=)

**Obs.:** 1 sinal de igual (=) atribue; dois sinais de igual (==) compara; (!=) diferente

## Tipos de operadores

Operadores básicos: (+, -, \*, /)

Potenciação (\*\*) Ex.: 2 \*\* 3

Div (//)

Mod( %) Ex.: 5 % 2 - resto de uma divisão

Ex.: 5/2 = 2.5 (resultado real)

5//2 = 2 (só pega a parte inteira) – não arredonda os valores

**Sintaxe** Um programa feito em Python é constituído de linhas, que podem continuar nas linhas seguintes, pelo uso do caractere de barra invertida (\) ao final da linha ou parênteses, colchetes ou chaves, em expressões que utilizam tais caracteres. O caractere **#** marca o início de comentário. Qualquer texto depois do **#** será ignorado até o fim da linha.

```
>>> print (7 * 3)      # Escrever na tela o resultado de 7 vezes 3
21
```

```
>>> a = 7 * 3 + \      # Uma linha quebrada por contra barra
5 / 2
```

```
>>> b = ['a', 'b', 'c',  # Uma lista (quebrada por vírgula)
'd', 'e']
```

```
>>> c = list (range(1,   # Uma chamada de função (quebrada por vírgula) - formato de lista
11))
```

```
>>> print (a, b, c)      # imprime todos na tela
```

Testar no IDLE  
Na versão 3 do Python

## Mais exemplos...

O sinal usado por Python para potências algébricas é \*\*:

```
>>> 2**3    # É o mesmo que 2³ (dois ao cubo)  
8
```

```
>>> 2**(3+6) # Dois elevado a 9  
512
```

Há um operador que retorna o **resto** da divisão entre 2 números:

```
>>> 7 % 2    # O resto da divisão entre 7 e 2  
1
```

Quanto deve ser o resultado de **2\*\*(3+6)%7?**

---



# Variáveis: nada a declarar...

**Variável** - é o nome que se dá para uma região da memória onde é armazenado algum dado que está ou estará sendo manipulado pelo programa.

Como dissemos anteriormente, Python é uma linguagem dinamicamente tipada. Logo, **não é necessário tipar as variáveis** para usá-las. Por meio de um processo chamado “**binding**”, atribui-se um objeto a um nome e este nome incorpora o tipo de objeto.

**Ex.:**

```
>>> a=10      # O Python subentende que trata-se de um número inteiro
>>> b=32      # O Python subentende que trata-se de um número inteiro
>>> c=a==b    # O Python subentende que trata-se de uma variável booleana
>>> texto = "O céu é azul"  # O Python subentende que trata-se de uma string
>>> salario = 1750.80      # O Python subentende que trata-se de um float (real)
>>> l= list(range(1,10))   # O Python reconhece essa variável como uma lista
>>> l2=list(range(1,10,2)) # O Python reconhece essa variável como uma lista
```

Para saber qual o tipo de determinado objeto, usamos a função **type(x)**, que retorna o tipo do objeto x:

**Ex:**

```
print(type(a))    # mostrará qual é o tipo de dados da variável "a".
print(type(l2))   # mostrará qual é o tipo de dados da variável "l2".
print (type(c))   # mostrará qual é o tipo de dados da variável "c".
```

---



Contato:  
**jacfel@gmail.com**

**Bons estudos!**

