

Visão Geral

Links importantes

PYTHON

Configurando o Ambiente

Python no Linux

Python no OS X

Python no Windows

Usando o interpretador Python

Modo interativo

Modo Script

Blocos

Links importantes

- Fórum GUJ: <http://www.guj.com.br>
- Python Software Foundation (PSF): <https://python.org.br/>
- Fóruns Oficiais da Comunidade: <https://python-forum.io/> e <https://python.org.br/lista-de-discussoes/>


PYTHON

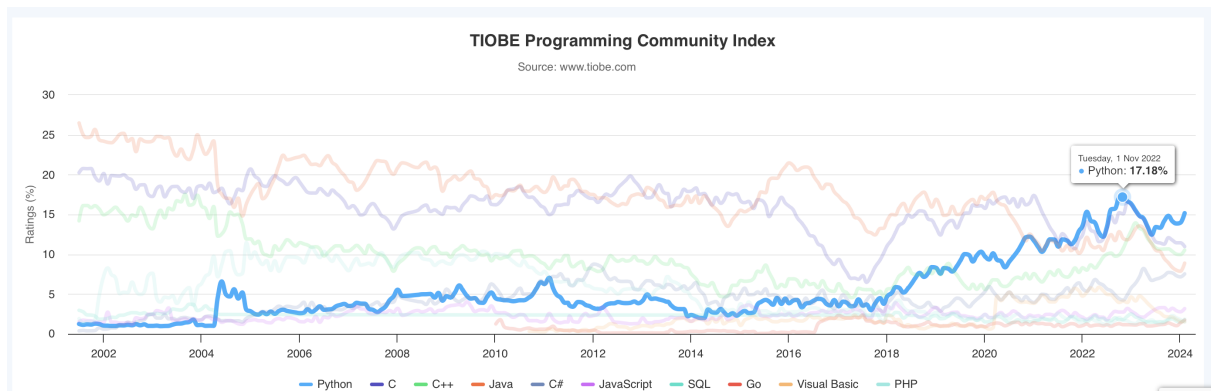
Python é uma linguagem de programação interpretada, orientada a objetos, de altíssimo nível e com semântica dinâmica e forte. A simplicidade do Python reduz a manutenção de um programa. Python suporta módulos e pacotes, que encoraja a programação modularizada e reuso de códigos. É um software de código aberto.

Ano	Evento
1989	Guido van Rossum inicia o desenvolvimento do Python
1991	Lançamento da primeira versão pública do Python (Python 0.9.0)
2000	Lançamento das versões 2.x do Python
2008	Lançamento das versões 3.x do Python
Presente	Python torna-se uma das linguagens de programação mais populares do mundo

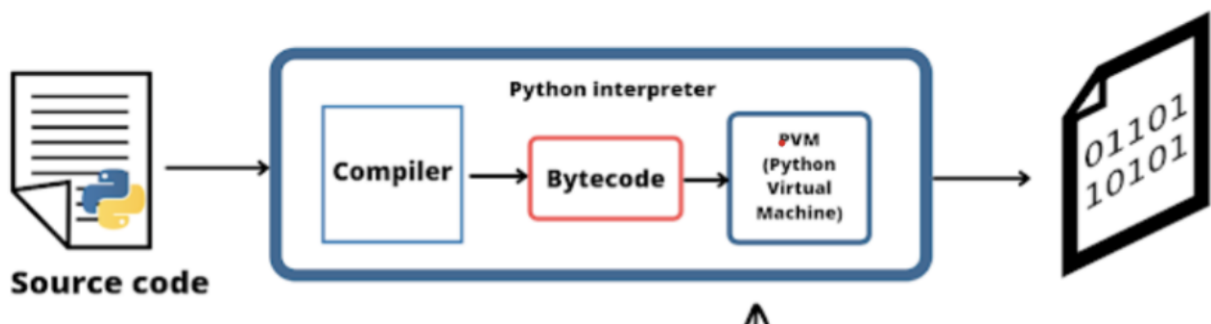


Todos os lançamentos de Python são de código aberto (veja <http://www.opensource.org>).

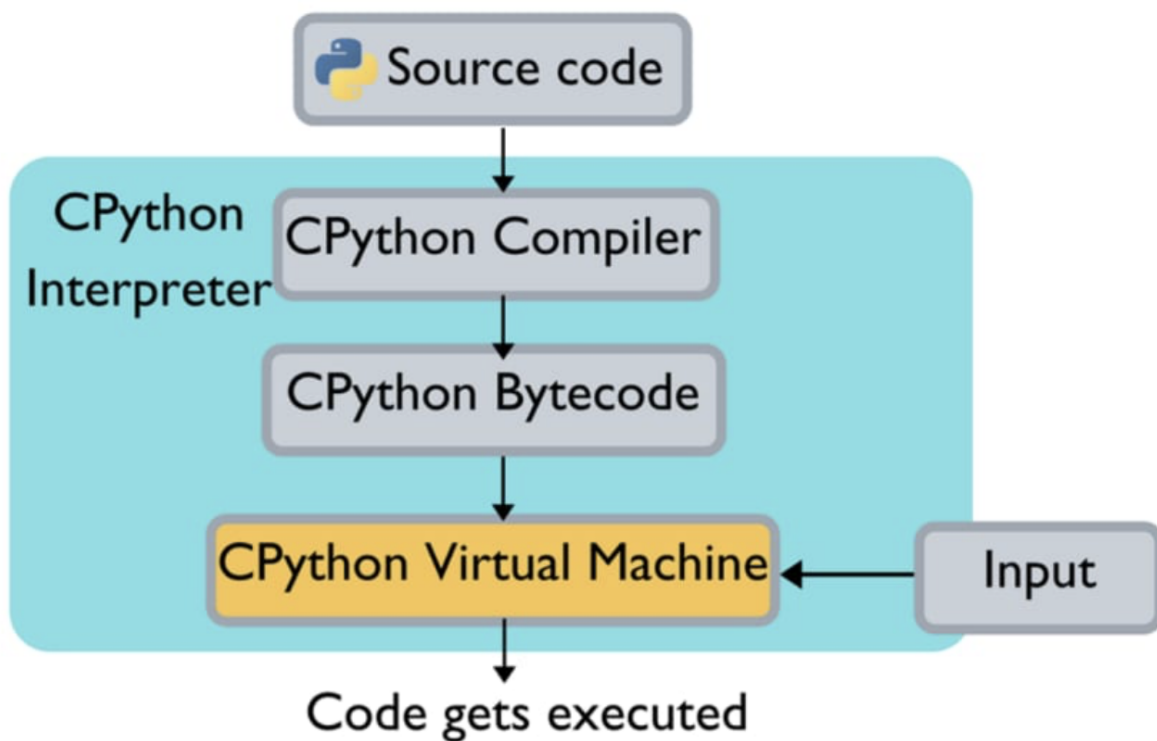
Feb 2024	Feb 2023	Change	Programming Language		Ratings	Change
1	1			Python	15.16%	-0.32%
2	2			C	10.97%	-4.41%
3	3			C++	10.53%	-3.40%
4	4			Java	8.88%	-4.33%
5	5			C#	7.53%	+1.15%
6	7	^		JavaScript	3.17%	+0.64%
7	8	^		SQL	1.82%	-0.30%
8	11	^		Go	1.73%	+0.61%
9	6	v		Visual Basic	1.52%	-2.62%



Um **compilador** traduz linguagem **Python** em linguagem de máquina - código Python é traduzido em um código intermediário que deve ser executado por uma máquina virtual conhecida como **PVM** (Python Virtual Machine).



É muito similar ao Java - há ainda um jeito de traduzir programas Python em **bytecode** Java para JVM (Java Virtual Machine) usando a implementação **Jython**.



CPython é uma implementação da linguagem Python. Para facilitar o entendimento, imagine que é um pacote que vem com um compilador e um interpretador Python (no caso, uma Máquina Virtual Python) além de outras ferramentas para usar e manter o Python.



Vale ressaltar para quem deseja utilizar uma implementação alternativa do Python, como o **IronPython** ou **Jython**, que o suporte ao Python3 ainda é bastante limitado. A diferença é que estas implementações são apenas os compiladores. O *bytecode* gerado pelo **Jython** é interpretado por uma **JVM** (Java Virtual Machine) e o *bytecode* do **IronPython** por uma **Virtual Machine .NET**.

Configurando o Ambiente

Antes de começarmos, precisamos instalar o interpretador da linguagem Python. O interpretador Python **não** vem instalado com o Microsoft Windows. Se você utiliza Mac OS X ou Linux, provavelmente **isso já foi feito**.

Python no Linux

Sistemas Linux são projetados para programação, portanto, Python já está instalado na maioria dos computadores Linux.

Abra uma janela de terminal executando a aplicação Terminal em seu sistema (CTRL-ALT-T, no Ubuntu). Digite `python -v`, a saída informará qual versão de Python está instalada:

```
$ python Python 2.7.6 (default, Mar 22 2014, 22:59:38) [GCC 4.8.2] on linux2
```

Essa saída informa que, no momento, Python 2.7.6 é a versão-padrão de Python instalada nesse computador. Saia pressionando CTRL-D ou `exit()`.

Para verificar se Python 3 está instalada, talvez você precise especificar essa versão, como:

```
$ python3 Python 3.5.0 (default, Sep 17 2015, 13:05:18) [GCC 4.8.4] on linux Type
```

No **Ubuntu** (versão 18.04), digite, para instalar a versão 3.7, por exemplo:

```
sudo apt update
```

```
sudo apt install -y python3.7 idle-python3.7
```

Como recomendação, utilize Geany (<http://geany.org/>) ou Sublime (<http://www.sublimetext.com/>), editores de texto simples. É fácil de instalar, permitirá executar quase todos os seus programas diretamente do editor em vez do terminal. Você pode instalar o Geany como uma linha na maioria dos sistemas Linux:

```
$ sudo apt-get install geany
```

Se várias versões de Python estiverem instaladas em seu sistema, será preciso garantir que o Geany esteja configurado para utilizar a versão correta. Acesse **Build > Set Build Commands**. Você deverá ver as palavras *Compile* e *Execute* como um comando de lado de cada um. O Geany pressupõe que o comando correto para cada um é **python**, mas seu sistema utiliza o comando **python3**, será necessário alterar essa informação.

Mude os comando *Compile* e *Execute* para que o Geany utilize o interpretador do Python 3. Seu comando *Compile* deve ser:

```
python3 -m py_compile "%f"
```

Seu comando *Execute* deve ser:

```
python3 "%f"
```

Pronto, agora está pronto para executar um comando `print("Hello Python World!")` no Geany ou no terminal.

Python no OS X

Python **já vem instalado** na maioria dos sistemas OS X. Depois que tiver certeza de que Python está presente, você precisará instalar um editor de texto.

Abra uma janela de terminal acessando **Applications > Utilities > Terminal**. Você também pode pressionar **COMMAND**-barra de espaço, digitar terminal e, então, teclar **ENTER**. Digite `python` ou `python3`, você deverá ver uma saída que informa qual versão de Python está instalada em seu sistema:

```
$ python Python 2.7.5 (default, Mar 9 2014, 22:15:05) [GCC 4.2.1 Compatible Apple LLVM 5.0 (clang-500.0.68)] on darwin Type "help", "copyright", "credits", or "license" for more information.
```

Essa saída informa que, no momento, Python 2.7.5 é a versão padrão de Python instalada nesse computador. Para conferir se você tem Python 3, experimente executar o comando **python3**. Quando vir a saída, pressione CTRL-D ou digite `exit()` para sair do Prompt de Python.

O Sublime Text é um editor de texto simples: é fácil de instalar no OS X, permitirá executar quase todos os seus programas diretamente do editor em vez do terminal. Você pode fazer download do instalador do Sublime Text a partir de <http://sublimetext.com/3>.

Para configurar o Sublime Text, dê o seguinte comando para descobrir o caminho completo de seu interpretador Python:

```
$ type -a python3
```

```
python3 is usrlocal/bin/python3
```

Agora abra o Sublime Text e acesse **Tools > Build System > New Build System**. Apague o que vir e insira o seguinte:

```
Python3.sublime-build {"cmd": ["usrlocal/bin/python3", "-u", "$file"], }
```

Esse código diz ao Sublime Text para usar o comando **python3** de seu sistema quando executar o arquivo aberto no momento.

Para executar seu primeiro programa, abra o Sublime Text, digite `print("Hello Python world!")`. Pressione CTRL-B ou selecione **Tools>Build**.

Python no Windows

O Windows nem sempre vem com Python, portanto, é provável que você vá precisar fazer download e instalá-lo.

Primeiro, verifique se o Python está instalado em seu sistema. Abra uma janela de comando, digite **python**. Se você obtiver um prompt Python (`>>>`), é sinal de que Python está instalado em seu sistema. No caso, de uma mensagem de erro, faça download do instalador de Python para Windows acessando <http://python.org/downloads/>. Neste exemplo, os arquivos foram instalados na pasta `C:\Python3.12`.

Depois de instalado, abra o terminal e digite **python**, vai aparecer:

```
C:\> python Python 3.12.0 (v3.5.0:374f501f4567, Sep 13 2015, 22:15:05) [MSC v.1900 32 bit (Intel)] on win32
```

Caso não apareça essa mensagem, localize o arquivo **python**, copie o caminho do sistema, abra uma janela de comando e digite o caminho completo:

```
C:\> C:\Python3.12\python Python 3.12.0 (v3.5.0:374f501f4567, Sep 13 2015, 22:15:05) [MSC v.1900 32 bit (Intel)] on win32
```

O editor de texto indicado será o Geany (<http://geany.org/>). Se for necessário configurar o Geany, acesse **Build>Set Build Commands**. Nos comandos Compile e Execute, acrescente respectivamente:

```
C:\Python3.12\python -m py_compile "%f" e
```

```
C:\Python3.12\python "%f"
```

Usando o interpretador Python

Modo interativo

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. Basta executar:

```
python ou python3
```

Ele estará pronto para receber comandos após o surgimento do sinal de espera `>>>`.

Modo Script

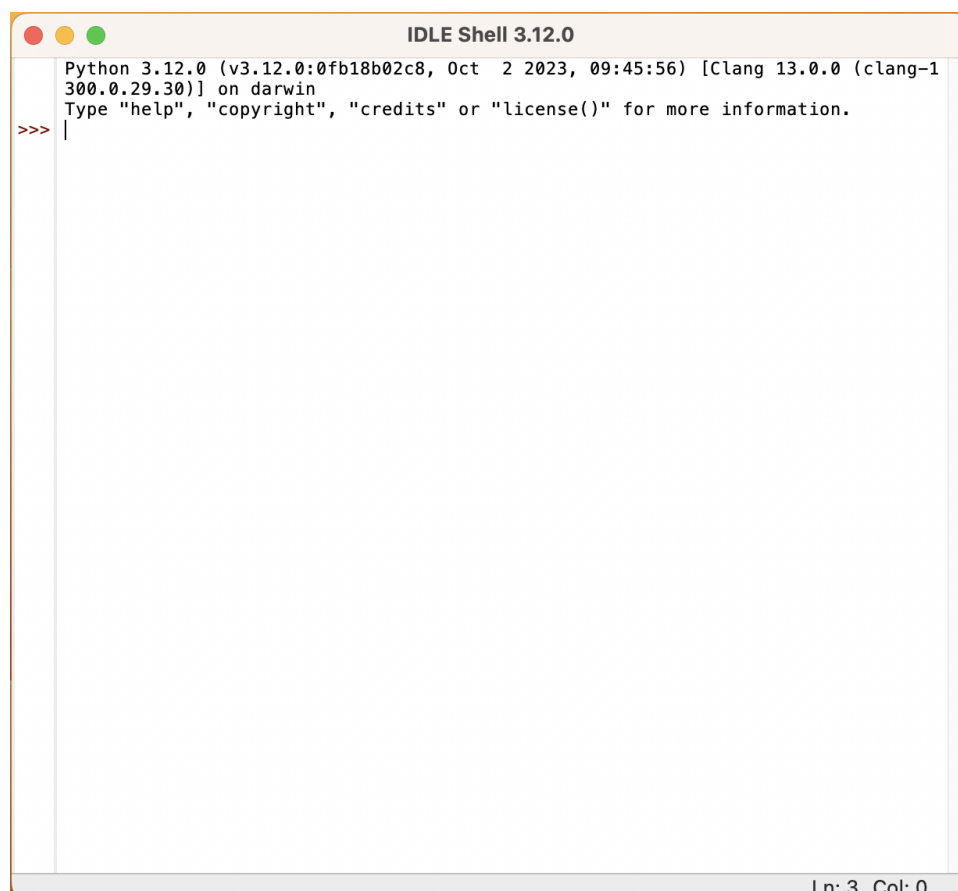
O **IDLE** é uma interface gráfica para o interpretador Python, permitindo também a edição e execução de nossos programas.

No Windows, você deve ter uma pasta no menu Iniciar>Programas>Python3.12. Escolha IDLE.

No Linux, abra o terminal e digite: `idle-python3.12 &`

No MacOS X, abra o terminal e digite: `IDLE3.12 &`

A janela inicial do **IDLE** é mostrada na figura abaixo. Se você utiliza Windows ou Linux, essa janela não será exatamente igual, mas muito parecida.



Observe que o cursor está posicionado dentro da janela **Python Shell**, e que a linha de comandos é iniciada pela sequência `>>>`.

Digite: `print("Hello World!")` Ao pressionar a tecla Enter, você deve obter o texto abaixo **Hello World**.



O `print()` é uma função do Python utilizada para imprimir alguma mensagem na tela. Uma mensagem deve estar delimitada entre aspas simples (') ou duplas (").

No caso da linguagem Python, podemos usar qualquer **editor de textos** disponíveis: Notepad++, Sublime, VS Code, TextMate (Mac OS X), Emacs (Linux).

Continuaremos utilizando o editor de textos incluído na instalação do interpretador **Python (IDLE)**. Abra um novo arquivo, digite o código anterior e salve. Escreva **teste.py** no nome do arquivo e clique no botão para salvar. Atenção: a extensão **.py** não é adicionada automaticamente pelo IDLE. Lembre-se de sempre gravar seus programas escritos em Python com a extensão **.py**. Agora clique no menu **Run** para executar seu programa. Ou pressionar a tecla **F5**. O programa será executado na outra janela, no interpretador.



Normalmente não precisamos compilar explicitamente, pois o Python está fazendo nos bastidores. Mas se quiser compilar explicitamente, use o `import py_compile` OU `python3 -m py_compile teste.py`

```
import py_compile
py_compile.compile('teste.py')
```

Dentro da pasta é criado um arquivo **teste.cpython-34.pyc** - esta é a versão compilada do programa, o **bytecode** gerado pelo **CPython**.

Outro exemplo de programa em Python:

```
#Uma lista de instrumentos musicais
instrumentos = ['Baixo', 'Bateria', 'Guitarra']

#Para cada nome na lista de instrumentos
for instrumento in instrumentos:

    #mostre o nome do instrumento musical
    print(instrumento)
```

Em **Python**, você deve tomar cuidado com os seguintes itens:

1. Letras maiúsculas e minúsculas são diferentes.
2. Aspas são muito importantes e não devem ser esquecidas.
3. Parênteses não são opcionais em Python.
4. Espaços são muito importantes. A linguagem Python se baseia na quantidade de espaço em branco antes do início de cada linha para realizar diversas operações.

Python utiliza **tipagem dinâmica**, o que significa que o tipo de uma variável é inferido pelo interpretador em tempo de execução. A tipagem do Python é **forte**, ou seja, o interpretador verifica se as operações são válidas e não faz coerções automáticas entre tipos incompatíveis.

O interpretador Python também pode ser utilizado como calculadora.

Experimente digitar `2+3` ou `5-3` ou `20/4`.

Blocos

Em Python, os blocos de código são delimitados pelo uso de endentação, que deve ser constante no bloco de código, porém é considerada uma boa prática

manter a consistência no projeto todo e evitar a mistura tabulações e espaços.

A linha anterior ao bloco sempre termina com dois pontos (:) e representa uma estrutura de controle da linguagem ou uma declaração de uma nova estrutura (uma função, por exemplo).

