

[◀ VOLTAR](#)

Histórico e características da linguagem Python

Apresentar a linguagem Python, um pouco da sua história e suas principais características.

NESTE TÓPICO

[> O INTERPRETADOR](#)[> O COMPILADOR](#)[> A LINGUAGEM PYTHON](#)[> CARACTERÍSTICAS DA LINGUAGEM](#) Ver tópico[> COMO TRABALHA O PYTHON](#)

Olá alunos,

Bem, depois que vimos a necessidade de antes criarmos um algoritmo para resolver um problema do mundo real e depois utilizarmos uma linguagem de programação para implementarmos o algoritmo, vamos ver algumas das principais linguagens utilizadas na história da programação de computadores:

Evolução das Linguagens de Programação:

- Assembly – 1947: linguagem de montagem de baixo nível
- Fortran – 1954: linguagem sequencial (não estruturada)
- COBOL – 1959: linguagem sequencial (não estruturada)
- BASIC – 1964: linguagem sequencial (não estruturada)
- PASCAL – 1968: estruturada: orientada a funções (procedimentos)
- C – 1972: estruturada: orientada a funções (procedimentos)
- DELPHI e Visual Basic – 1994: orientada a eventos
- Simula – 1978: orientada a objetos (OO)
- C++ - 1980: orientada a objetos (OO)

- **ADA – 1983: orientada a objetos (OO)**
- **JAVA – 1995: orientada a objetos (OO)**
- **Object Pascal, Eiffel, C# - 2001: orientada a objetos (OO)**

No princípio, a programação era feita diretamente com a linguagem de máquina, a linguagem binária composta pelos bits: 0 e 1. Para facilitar a vida dos programadores, foi criada a linguagem Assembly com comandos em linguagem humana: **mov**, **printf**, por exemplo, mas como o computador só entende linguagem binária, foi criado o **Assembler** (montador) que traduzia os comandos em linguagem humana para linguagem de máquina. A próxima geração de linguagens: FORTRAN, COBOL e BASIC começam a utilizar o INTERPRETADOR para fazer a tradução da linguagem de alto nível (linguagem humana) para a linguagem binária. A partir da linguagem PASCAL, começa-se a utilizar COMPILADORES e o pessoal da Sun Microsystems® (adquirida pela Oracle Corp®), na década de 90, utilizaram os dois: o INTERPRETADOR e o COMPILADOR na criação da linguagem JAVA.

A diferença entre o interpretador e o compilador:

O INTERPRETADOR

Pega a primeira linha do código, verifica a sintaxe e se estiver correta, transforma em linguagem binária e a instrução é executada pelo processador. Depois pega a segunda linha de código e realiza o mesmo processo, e assim por diante até a última instrução.

O COMPILADOR

Pega o conjunto de instruções e verifica a sintaxe e se estiver tudo correto, gera um arquivo em linguagem binária chamado de módulo de carga (porque será carregada na memória principal) com a extensão **“.EXE”**, este tipo de arquivo é mais conhecido como aplicativo. Para a execução, basta dar dois cliques no arquivo e o conjunto de instruções em binário é carregada na Memória Principal para ser executada pelo Processador.

Tanto o INTERPRETADOR e o COMPILADOR transformam o código escrito em linguagem humana em linguagem de máquina, mas a diferença é que o interpretador faz isto em tempo de execução enquanto o compilador faz isto em tempo de compilação.

Na linguagem JAVA, o programador escreve o código e este passa pelo compilador e é transformado em um arquivo de bytecodes com uma extensão **“.class”**. Uma outra característica do interpretador é que ele era construído de acordo com o hardware, e assim para executar o código em JAVA, basta ter o interpretador instalado naquele hardware, no caso, o interpretador do JAVA é conhecido como **JVM** (Java Virtual Machine). Esta grande sacada do pessoal, que criou a linguagem Java, fez com que o programa se tornasse portátil, uma vez escrito e compilado, poderia ser executado em qualquer hardware, desde que tivesse a JVM. Por isso, os programas em Java, não só estão em computadores como também em celulares, TVs, micro-ondas, carros e também nos robôs exploradores de Marte. O slogan entre os programadores em Java era: **“WRITE ONCE, RUN ANYWHERE”** (uma vez escrito, roda em qualquer lugar).

A LINGUAGEM PYTHON

Criada a partir do final da década de 80, e lançada em 1991, pelo holandês: Guido Van Rossum. E em 2008 foi lançado o **Python 3**. Ela nasce com o principal objetivo de “aumentar a produtividade do programador”.

Apesar de utilizar uma serpente como símbolo, da região da Índia: a piton, que no Brasil tem a sua equivalente que é a anaconda, também conhecida como sucuri, a origem do nome está no seriado britânico de tv e cinema das décadas de 60, 70 e 80: MONTY PYTHON'S FLYING CIRCUS® (por curiosidade, um dos seus seriados chama-se SPAM).

A linguagem é de código aberto (open source), com licença tipo GPL (General Public License) e como todo sistema de licença livre tem um animal como símbolo: o pinguim do Linux, o golfinho do MySQL, a foca da MariaDB, o pássaro do FireBird, etc (porque todo animal é livre), a serpente foi adotada como símbolo da linguagem Python.

CARACTERÍSTICAS DA LINGUAGEM

- A linguagem possui um INTERPRETADOR, a exemplo da linguagem Java e tem uma programação sequencial, estruturada, semelhante as linguagens C, Pascal, ADA e orientada a objetos semelhante a Java, C++ e C#.
- Linguagem de alto nível.
- Linguagem interpretada e interativa.
- Tem portabilidade e permite automatização de tarefas.
- Possui muitas aplicações nos setores: comercial, científico, educacional e em muitas plataformas: servidor, desktop, web e móvel.
- Utilizadas pelos motores de busca do Google®, projetos da NASA, CIA, YouTube®, o site do governo federal, etc.

COMO TRABALHA O PYTHON

CÓDIGO EM PYTHON —> SCRIPT (nome.py) —> BYTECODE —> Execução

CÓDIGO EM PYTHON:

Para desenvolver em qualquer linguagem devemos utilizar um **IDE (Integrated Development Environment)**, um ambiente de desenvolvimento.

Em Python podemos utilizar um IDE em **CLI (Command Line Interface)** em linha de comando, conhecido como prompt ou uma IDE em **GUI (Graphical User Interface)** conhecida como **IDLE** (o nome da interface é em homenagem a Eric Idle, um dos atores de Monty Python).

SCRIPT

Através da IDLE, podemos criar scripts, documentos tipo texto, só que em vez de ter um texto qualquer, terá linha de códigos. Basta salvar o código com um nome, na extensão **.py** e o arquivo poderá ser executado com um duplo clique, e também poderá ser executado em outro dispositivo como o celular.

BYTECODE

O script ou código no interpretador interativo é transformado em bytecode pelo PVM (Python Virtual Machine) e executado.

COMO PROGRAMAR EM PYTHON

- Podemos também utilizar o interpretador interativo na página do próprio site da Python: <https://www.python.org/> (<https://www.python.org/>)

Para **LINUX**:

- O Python já é nativo na maioria das distribuições LINUX.

Para **Windows**:

- Podemos fazer download do interpretador através do site: <https://www.python.org/downloads/> (<https://www.python.org/downloads/>)

Atualmente existem duas versões para Windows:

Python 3.x.y (versão 3) e *Python 2.x.y* (versão 2)

Utilizaremos o **Python 3** e muitas aplicações feitas em Python 2 ou simplesmente Python, não rodam em Python 3.

Veja no vídeo abaixo (sem áudio) como acessar o site do Python e fazer o download para o Windows:

./videos/370207913.mp4



COMO UTILIZAR

No **LINUX**:

Abrir o terminal (prompt Shell) e digitar: *python3* para a versão 3 ou digitar *python* para a versão 2.

No **WINDOWS®**:

Pode-se abrir o **Python 3** (prompt em CLI) ou **IDLE Python** (em GUI).

Trabalhando com a IDLE:

Diferente da CLI do DOS, do SHELL ou do PYTHON, não recupera os comandos digitados através das teclas de seta para cima ou para baixo. Para retornarmos os comandos, deve-se utilizar as teclas:

Alt + P - para recuperar os comandos anteriores (Previous)

Alt + N - para recuperar os comandos seguintes (Next)

Para encerrar:

Ctrl + D ou digitar: **exit()** e pressionar Enter

Trabalhando com um script Python:

Abra o **IDLE** e em **FILE – NEW FILE**

Escreva o código em Python

1. Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Intel)] on win32
2. Type "copyright", "credits" or "license()" for more information.
3. >>> 2 + 5
4. 7
5. >>>

Depois: **File - Save as - teste.py**

Para executar o script:

Localize o script e dê duplo clique.

OU

Pelo **prompt do Windows**:

cd c:\nome_pasta\teste.py

Pelo **terminal do Linux**:

python3 teste.py

Pelo **IDLE**:

File - Open - teste.py e pressione **F5**

SAIBA MAIS...

Dê uma olhada nos links abaixo para saber mais sobre a linguagem Python:

<https://www.python.org/doc/> (<https://www.python.org/doc/>)

<https://wiki.python.org/moin/PythonBooks>
(<https://wiki.python.org/moin/PythonBooks>)

Neste tópico apresentamos a linguagem Python, conhecemos um pouco da sua origem e suas principais características. Vimos também, as várias formas de utilizarmos a linguagem, que pode ser diretamente no seu interpretador interativo ou através da criação de scripts.

Quiz

Exercício Final

Histórico e características da linguagem Python

INICIAR ➤

Referências

SUMMERFIELD, M. *Programação em Python 3: Uma introdução completa à linguagem Python*. Rio de Janeiro Alta Books, 2012. 495 p.

MENEZES, N. N. C. *Introdução à programação com Python: algoritmos e lógica de programação para iniciantes*. 2. ed. São Paulo: Novatec, 2014. 328 p.

SWEIGART, AL. *Automatize tarefas maçantes com Python: programação prática para verdadeiros iniciantes*. São Paulo: Novatec, 2015. 568 p.

PYTHON, doc. Disponível em: <<https://www.python.org/doc/>>. Acesso em: Junho/2018.

PYTHON, books. Disponível em: <<https://wiki.python.org/moin/PythonBooks>>. Acesso em: Junho/2018.



Avalie este tópico



ANTERIOR

Conceitos e princípios da lógica de programação



Índice

Biblioteca
(<https://www.uninove.br/conhec-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/>)
Portal Uninove
(<http://www.uninove.br>)
Mapa do Site

Estrutura do código Python e a indentação

Ajuda?
PRÓXIMO
(<https://ava.uninove.br/ava/estrutura-do-codigo-python-e-a-indentacao/>)

© Todos os direitos reservados

