VOLTAR

Estrutura do código Python e a indentação

Apresentar a estrutura do código em Python e o principal da estrutura, a indentação.

NESTE TÓPICO

Marcar tópico

Olá alunos,

Bem pessoal, vamos começar a colocar a "mão na massa", como é uma disciplina de programação, então não há outra forma de aprender bem, senão praticando. Espero que já tenham instalado o interpretador Python para plataforma Windows ou se você estiver utilizando Linux, não é necessário instalar, pois o Python já vem nativo e basta então abrir o terminal (prompt) do Linux e digitar: **python3**.

No Windows, você pode abrir o **Python 3.6,** por exemplo, que é o Shell ou prompt, onde os comandos são executados em linha e digitar o seguinte exemplo de código:

o programa alô pessoal

print("Alô pessoal","estou estudando 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 "help", "copyright", "credits" or "license" for more information.
3.  >>> # o  programa alô pessoal
4.  ... print("Alô pessoal","estou estudando Python!!")
5.  Alô pessoal estou estudando Python!!
6.  >>>
```

Agora, abra o **IDLE (Python 3.6)** e digite o mesmo código:

```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>> # o  programa alô pessoal
>>> print("Alô pessoal","estou estudando Python!!")
Alô pessoal estou estudando Python!!
>>>
```

Trabalhando com um script Python

O código Python pode ser escrito com qualquer editor de texto puro, o Bloco de Notas do Windows, por exemplo, **não** utilize o Word do Office.

Python utiliza a codificação de caracteres UTF-8 que é um subconjunto de ASCII, já as strings no Python 3 são Unicode por padrão, daí você não vai ter problemas em utilizar acentuação gráfica ou o "ç".

Abra o **IDLE** novamente e vá em **File** e **New File** (no menu acima, à esquerda). Ele abrirá um documento em branco, é aí que você vai criar o **script.** Digite novamente o mesmo código:

```
# o  programa alô pessoal
print("Alô pessoal","estou estudando Python!!")
```

Agora, para executarmos o **script** é um pouco diferente (e existem várias formas!):

- Primeiro salve o **script** com um nome, por exemplo, **teste.py** (é necessário salvar com a extensão **.py**):

Vá em **File** -- **Save as – teste.py** (escolha onde você vai gravar o arquivo, pode ser na "Área de Trabalho", por exemplo)

Daí, pressione **F5** ou vá no menu acima, em **Run – Run Module** para executar.

OU

- Se você fechar o arquivo, você pode abri-lo pelo **IDLE**:

Em **Open** -- **teste.py**, para editar, ou pressione **F5** ou vá no menu em **Run – Run Module** para executar.

******* Se você tiver que mexer (editar) o código, é melhor abri-lo pelo **IDLE.**

OU

- **Para executar o script:**

Localize o arquivo do script e dê duplo clique.

OU

- Pelo prompt do **Windows**:

cd c:\nome_pasta\teste.py

- Pelo terminal do **Linux**:

python3 teste.py

> **EXECUTANDO O MESMO SCRIPT EM OUTROS DISPOSITIVOS:**
>
> **** Atenção! Uma vez que você criou o script, você pode rodá-lo no seu celular ou tablet, basta instalar o interpretador do Python no seu celular ou tablet (se o seu celular é Android, pode instalar o QPython, por exemplo) mas também existem outros interpretadores para Python e para iOS na App Store. Daí, basta copiar o arquivo do script para o seu celular ou tablete.**

Vamos analisar o que aconteceu na execução dos diversos ambientes:

- No primeiro exemplo, criamos e executamos o código no **Python 3.6**, é uma interface em linha de comando, tipo Shell do Unix/Linux. Este é um **interpretador interativo**, você vai digitando o código e ele já vai interpretando e executando o seu código.
- No segundo exemplo, criamos e executamos o mesmo código no **IDLE**, é uma interface GUI em linha de comando. Este também é um **interpretador interativo**, você vai digitando o código e ele já vai interpretando e executando o seu código.
- No terceiro exemplo, abrimos uma janela para a criação do **script** através da interface do **IDLE**. Note que não aparecem os sinais "**>>>**" que representam o prompt do interpretador interativo. Para executar tem que salvar o script com um nome e depois executar o arquivo.

**** Daqui para frente, vamos trabalhar com o script!!!**

ESTRUTURA DO CÓDIGO PYTHON

Numa primeira observação, percebe-se que **não** existe o famoso "**;**" para finalizar a linha de comando como em outras linguagens, para finalizar a linha em Python, basta pressionar "**ENTER**".

 # o programa alô pessoal

Nesta linha de código, temos uma linha de comentário: ela não será executada e será ignorada pelo interpretador. Toda linha de comentário em Python inicia com o sinal "**#**", semelhante ao que existe em C, Java, C# com "//".

print("Alô pessoal","estou estudando Python!!")

Nesta segunda linha de código, temos um comando de saída de tela: "**print**" (semelhante ao comando *printf* da linguagem C), acompanhado de um par de ( ) e dentro deles, a mensagem de texto deverá ser colocada entre **aspas** e a **vírgula** representa a concatenação (semelhante ao sinal de + em Java ou aos dois sinais de pipes (||) em PL/SQL).

O Python, além de não utilizar o ";" para finalizar a linha de comando, também não tem marcações para iniciar e finalizar bloco de comandos, como "BEGIN", "END;", "END IF", "END LOOP" em PASCAL, ADA ou "{" e "}" como em C, C#,JAVA.

INDENTATION (INDENTAÇÃO)

Em Python, o início de um bloco geralmente é com "**:**" depois do comando e o interpretador sabe que bloco foi finalizado pela **INDENTAÇÃO.**

Exemplo:

comandos

|——>while condição:

comando1

if condição:

|————————>comando2

else:

|————————>comando3

comando4

No último comando, no **comando4**, foi retirado o recuo ou a indentação, isto representa que o **comando4** não faz parte do comando **else,** mas o **comando3**, sim..

* Não misturar espaços e tabulação na indentação, utilizar um ou outro.

* Várias IDEs já vêm com a indentação intuitiva como a **IDLE**.

Os comandos em Python e os nomes de variáveis e funções determinados pelos programadores são **case sensitive**.

Exemplos:

If é diferente de **IF**

nomeDaVariavel é diferente de **NomeDaVariavel**

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 vimos a estrutura da linguagem Python, com uma atenção especial para a indentação na hora de escrevermos o código. Vimos também, como criar e executar scripts em Python.

Quiz

Exercício Final

Estrutura do código Python e a indentação

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
Histórico e características da linguagem Python

Índice

PRÓXIMO
Declaração de variáveis e funções de entrada e saída

Ajuda?
(https://ava.uninove.br/seu/AVA/topico/topico.php?idCurso=)