

Aprendendo Python

Um Guia Prático para Iniciantes na Programação



Título: Sintaxe Básica de Python

Autor: Patrick Yuri

Sintaxe Básica de Python

1. Introdução

A sintaxe do Python define um conjunto de regras usadas para criar programas nesta linguagem. Embora apresente semelhanças com linguagens como Perl, C e Java, o Python possui características distintivas que o tornam único e especialmente acessível para iniciantes.

2. Primeiro Programa Python

2.1 Modo Interativo (Terminal)

Abra o terminal (cmd) ou o que desejar, certifique que o python esteja instalado, e insira o comando python. O **terminal** trará a versão instalada:

```
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on  
win32 Type "help", "copyright", "credits" or "license" for more information.  
>>>
```

O modo interativo permite executar comandos Python diretamente no **terminal**:

```
>>> print("Hello, World!")  
Hello, World
```

Prompt Python: O símbolo >>> indica onde você pode digitar comandos.

2.2 Modo Script

Programas Python são salvos em arquivos com extensão .py:

```
# arquivo: test.py
print("Hello, World!")
```

Execução: python3 test.py

3. Identificadores

Identificadores são nomes usados para variáveis, funções, classes, módulos ou outros objetos.

Regras para Identificadores:

- Iniciam com letra (A-Z, a-z) ou underscore (_)
- Seguidos por letras, underscores ou dígitos (0-9)
- Não podem conter caracteres especiais (@, \$, %)
- **Python é case-sensitive:** Manpower ≠ manpower

Convenções de Nomenclatura:

- **Classes:** Iniciam com letra maiúscula (ex: MinhaClasse)
 - **Outros identificadores:** Iniciam com letra minúscula (ex: minha_variavel)
 - **Identificadores privados:** Iniciam com underscore (ex: _privado)
 - **Fortemente privados:** Iniciam com dois underscores (ex: __muito_privado)
-

4. Palavras Reservadas

Palavras-chave que **NÃO** podem ser usadas como identificadores:

and	as	assert
break	class	continue
def	del	elif
else	except	False
finally	for	from
global	if	import
in	is	lambda
None	nonlocal	not
or	pass	raise
return	True	try
while	with	yield

5. Indentação (Característica Essencial)

Python usa **indentação obrigatória** para definir blocos de código (não usa chaves como C/Java).

Exemplo Correto:

```
if True:
    print("True")
else:
    print("False")
```

Exemplo Incorreto:

```
if True:
    print("Answer") # Erro: indentação inconsistente
    print("True")
else:
    print("False")
```

Regra: Todos os comandos dentro do mesmo bloco devem ter a mesma indentação.

6. Declarações Multi-linha

6.1 Continuação Explícita (barra invertida)

```
total = item_one + \
        item_two + \
        item_three
```

6.2 Continuação Implícita (colchetes, chaves, parênteses)

```
days = ['Monday', 'Tuesday', 'Wednesday',
        'Thursday', 'Friday']
```

7. Aspas e Strings

Python aceita três tipos de aspas para strings:

```
# Aspas simples
word = 'word'
```

```
# Aspas duplas  
sentence = "This is a sentence."
```

```
# Aspas triplas (múltiplas linhas)  
paragraph = """This is a paragraph.  
It is made up of multiple lines  
and sentences."""
```

Regra: O tipo de aspa que inicia deve ser o mesmo que termina a string.

8. Comentários

8.1 Comentários de Linha Única

```
# Este é um comentário  
print("Hello, World!") # Comentário no final da linha
```

8.2 Comentários Multi-linha

```
# Este é um comentário  
# que ocupa múltiplas  
# linhas  
  
# OU usando aspas triplas:  
"""  
Este é um comentário  
de múltiplas linhas  
"""
```

Símbolo: O hash (#) inicia um comentário (exceto dentro de strings).

9. Linhas em Branco

Linhas contendo apenas espaços em branco ou comentários são ignoradas pelo interpretador Python. Use-as para melhorar a legibilidade do código.

10. Múltiplas Declarações em Uma Linha

Use ponto e vírgula (;) para separar múltiplas declarações:

```
import sys; x = 'foo'; sys.stdout.write(x + '\n')
```

Nota: Evite esta prática quando possível, pois reduz a **legibilidade**.

11. Suítes (Blocos de Código)

Suítes são grupos de declarações que formam um bloco de código. Usadas em estruturas compostas como **if**, **while**, **def**, **class**.

Estrutura:

```
if expression:
    suite
elif expression:
    suite
else:
    suite
```

Características:

- Linha de cabeçalho termina com dois pontos (:)
 - Seguida por uma ou mais linhas indentadas (a suíte)
-

12. Resumo dos Conceitos-chave

1. **Indentação obrigatória** define blocos de código
 2. **Case-sensitive:** diferencia maiúsculas de minúsculas
 3. **Comentários** começam com #
 4. **Três tipos de aspas** para strings
 5. **Palavras reservadas** não podem ser identificadores
 6. **Identificadores** seguem regras específicas de nomenclatura
 7. **Suítes** formam blocos de código em estruturas compostas
-

13. Exercícios Práticos

Exercício 1: Hello World

Crie um programa que imprima "Bem-vindo ao Python!" na tela.

Exercício 2: Identificadores

Identifique quais dos seguintes são identificadores válidos:

- minha_variavel
- 2variavel
- _privado
- Classe@Nome
- valorTotal

Exercício 3: Indentação

Corrija o seguinte código:

```
if 5 > 3:  
print("Cinco é maior")  
print("Que três")
```

Exercício 4: Comentários

Adicione comentários explicativos ao seguinte código:

```
nome = "Python"  
versao = 3.10  
print(f"{nome} versão {versao}")
```

14. Boas Práticas

1. **Use indentação consistente** (4 espaços é o padrão)
2. **Nomeie identificadores de forma descritiva**
3. **Comente código complexo**, não o óbvio
4. **Use linhas em branco** para separar seções lógicas
5. **Evite múltiplas declarações** em uma linha
6. **Siga as convenções PEP 8** (guia de estilo Python)

Referências

- Documentação Oficial Python: <https://docs.python.org>
- PEP 8 - Style Guide for Python Code: <https://peps.python.org/pep-0008/>
- Tutorial Tutorialspoint - Python Basic Syntax: <https://www.tutorialspoint.com/python>