

Curso de especialização em Geoprocessamento - UFABC/UAB
Disciplina de Introdução a Programação para Geoprocessamento

MATERIAL ADICIONAL1 : INDENTAÇÃO E SÍMBOLOS EM PYTHON

Prof. Dr. Alexandro G. Schafer

1. Indentação no Python

Em Python, a indentação é usada para definir blocos de código que são executados como uma unidade, como o corpo de uma função, loops, condicionais (if, elif, else), entre outros. Isso significa que o nível de espaçamento à esquerda das linhas de código determina a organização lógica do programa. A obrigatoriedade da indentação força um estilo visual limpo e consistente, que facilita a leitura e a compreensão do código por outros desenvolvedores.

O Python 3 (que utilizaremos ao longo de nosso curso) não permite a mistura de TABs e espaços para a indentação dentro de um mesmo bloco de código. A recomendação é usar 4 espaços por nível de indentação, conforme a PEP 8 (guia de estilo para Python). Códigos que são parte do mesmo bloco devem ter a mesma indentação.

```
def saudacao(nome):  
    mensagem = "Olá, " + nome # Esta linha está indentada com 4 espaços.  
    print(mensagem) # Esta linha também está no mesmo nível de indentação.
```

Para blocos de código aninhados (por exemplo, um if dentro de outro if), cada nível adicional de bloco deve ser indentado um nível mais à direita.

```
for i in range(3):  
    print("Número:", i)  
    if i == 2:  
        print("i é igual a 2") # Esta linha está mais indentada, com 8 espaços,  
                               # para indicar que é parte da condição if.
```

Erros Comuns

- **IndentationError**: Se você não seguir as regras de indentação corretamente, o Python irá levantar um **IndentationError**, indicando que há um problema com a forma como seu código está organizado espacialmente.

- Blocos vazios: Para definir um bloco de código que ainda não tem conteúdo, use a palavra-chave `pass` para evitar erros de indentação.

```
if condição_verdadeira:  
    pass # Permite a definição de um bloco vazio sem causar erro.
```

2. Entendendo a Linguagem de Símbolos e Aspas em Python: Fundamentos para Iniciantes

No Python, assim como em muitas outras linguagens de programação, símbolos específicos e aspas têm significados particulares e são usados para realizar diversas ações. Vamos detalhar o uso de aspas, além dos símbolos `:`, `=`, e `==`, para ajudar a entender melhor como eles funcionam em códigos Python.

Aspas (" ", ' ')

Aspas são usadas para definir strings, que são sequências de caracteres. Você pode usar tanto aspas simples (' ') quanto aspas duplas (" "), desde que o uso seja consistente dentro da mesma string.

```
nome = "João" # Aspas duplas  
saudacao = 'Olá, mundo!' # Aspas simples
```

Para strings que contêm aspas internamente, você pode usar aspas duplas para encapsular aspas simples e vice-versa, ou escapar as aspas usando uma barra invertida (`\`).

```
frase = "Ele disse: 'Olá!'"  
frase2 = 'Ele disse: \'Olá!\''
```

Dois Pontos (:)

O símbolo de dois pontos é usado em Python principalmente para definir o início de um bloco de código associado a estruturas de controle (como if, for, while) e definições de funções.

```
if idade > 18:
    print("Maior de idade") # O código indentado após o ~: é executado
                           #se a condição for verdadeira.
```

Sinal de Igual (=)

O sinal de igual é usado para atribuir um valor a uma variável. Ele diz ao Python para armazenar o valor à direita do sinal na variável à esquerda.

```
numero = 10 # Atribui o valor 10 à variável numero
```

Duplo Igual (==)

O duplo igual é usado para testar igualdade entre dois valores. Ele verifica se o valor à esquerda é igual ao valor à direita e retorna True (verdadeiro) ou False (falso).

```
resultado = (10 == 10) # Retorna True, pois os valores são iguais.
```

Uso Adicional de Símbolos

- Parênteses (()): Usados para definir a ordem das operações em expressões matemáticas e para passar argumentos para funções.

- Colchetes ([]): Usados para definir listas ou acessar elementos de listas e strings por índice.
- Chaves ({}): Usados para definir dicionários ou conjuntos.