RESUMO: PEP8

Israel Cassiano de Oliveira

O PEP8 (*Python Enhancement Proposal 8*) consiste num conjunto de regras para que a escrita de código em linguagem Python seja otimizada no sentido de facilitar a leitura e revisão pelo autor e por terceiros. Assim, por meio de uma padronização de estilo, a aplicação das diretrizes do PEP8 facilita tanto a identificação de erros no código como a colaboração entre programadores. As principais instruções são exploradas a seguir.

1. **Indentação**

Cada nível de indentação é definido com 4 espaços. Não misturar tab e espaços no mesmo código. Se for usar tab, use em todo o código.

Elementos contidos em parênteses, colchetes ou chaves devem estar alinhados à abertura do conjunto:

```
a = function_name(var_one, var_two,
                  var_three, var_four)
```

Alternativamente, abre-se o conjunto e a lista de elementos começa na linha abaixo, após 1 nível de indentação (4 espaços):

```
a = function_name(
    var_one, var_two,
    var_three, var_four
)
```

Note que essa indentação de 1 nível é opcional para o fechamento de parênteses/colchetes/chaves.

2. **Comprimento e quebra de linhas**

Limite as linhas a 79 caracteres. Ao usar operadores binários, realize a quebra de linha antes:

```
total = (val1
            + val2
            + val3
           )
```

3. **Linhas e espaços em branco**

Use duas linhas em branco abaixo e acima de funções de alto nível e definição de classe. Use uma linha em branco abaixo e acima de grupos de importações, de definições de métodos dentro de uma classe e de declarações.

Use um espaço antes e um depois de operadores binários. Não use espaços:

- Antes de vírgulas (somente após)
- Antes de dois pontos (para alguns casos, somente após)
- Entre chaves/colchetes/parênteses e os elementos contidos por eles
- Antes de chaves/colchetes/parênteses

```
spam(ham[1], {eggs: 2})
foo = (0,)
```

4. **Importação**

Cada importação deve estar em linhas separadas, sempre no início do arquivo. Organize as importações em três blocos: importações padrão da biblioteca, importações de bibliotecas de terceiros e importações de módulos locais.

5. **Codificação, idioma, nomes, comentários e docstrings**

Sempre utilize a codificação UTF-8 e caracteres ASCII. Escreva seu código em inglês, com exceções para alguns termos técnicos e nomes próprios.

Não use as letras L minúscula, i maiúscula e O maiúscula isoladamente para criar nomes, já que as duas primeiras podem ser confundidas entre si e a letra O pode ser confundida com o número zero.

Nomeie funções e variáveis com letras minúsculas e palavras separadas por underscores (ex.:genero_sp). Para classes, use o padrão CamelCase (ex.:GeneroSp). Já nomes de constantes serão sempre maiúsculas com underscores separando palavras (ex.: GENERO_SP).

Comentários são precedidos de #, começam com letra maiúscula e terminam com ponto final. Utilize-os para justificar seu código (contar o porquê), não para explicar os processos em andamento. Os comentários devem ser sempre claros e mantidos atualizados.

Docstrings de uma linha são delimitados por aspas triplas (“”””) no início e no final, na mesma linha. Docstrings com mais de uma linha são fechadas por aspas triplas uma linha abaixo. Sempre escreva docstrings para módulos, funções, classes e métodos públicos, no topo do código. Comece com letra maiúscula e termine com ponto final.