Python Basic Style Guide - Penggantian Baris, Komentar, dan Dokumentasi

Mengganti baris : Sebelum atau Sesudah Operator Binary

Bagian ini hanya memberikan gambaran mengenai standar penulisan, pembahasan mengenai kondisional dibahas di modul Operator, Operands, dan Expressions.

Penggantian baris setelah operator binary memang pernah menjadi rekomendasi. Namun ternyata penggunaan metode ini membuat mata cepat lelah dan Anda perlu melakukan pengecekan ulang pada baris berbeda. Contohnya:

```
1. income = (gross_wages +
2.           taxable_interest +
3.           (dividends - qualified_dividends) -
4.           ira_deduction -
5.           student_loan_interest)
```

Untuk menyelesaikan masalah ini, dipilih pendekatan baris baru sebelum operator binary. Hal ini untuk mempermudah pembaca kode mengerti operasi yang dilakukan terhadap variabel berikutnya.

```
1. income = (gross_wages
2.           + taxable_interest
3.           + (dividends - qualified_dividends)
4.           - ira_deduction
5.           - student_loan_interest)
```

Kedua pendekatan ini dimungkinkan di Python. Anda direkomendasikan untuk menggunakan pendekatan kedua (baris baru sebelum operator) untuk menulis kode baru.

Baris Kosong

Anda disarankan untuk menambahkan dua baris kosong pada top level function dan class definitions. Kemudian untuk setiap deklarasi method, dipisahkan dengan satu baris kosong.

kosong tidak diperlukan jika deklarasi fungsi/method Anda bersifat satu baris (one-liner), umumnya untuk fungsi/method yang belum diimplementasikan secara penuh.

File Encoding

Kode dalam inti Python, selalu menggunakan encoding UTF-8 (Python 3) atau ASCII (Python 2). Dalam hal ini, apabila dalam sebuah berkas tidak ditulis deklarasi encoding, maka berkas tersebut menggunakan encoding ASCII (Python 2) atau UTF-8 (Python 3). Dalam standard library, non-default encoding hanya digunakan untuk pengujian atau memberikan komentar/dokumentasi, misalnya nama penulis yang tidak menggunakan karakter ASCII.

Untuk Python 3 dan seterusnya, pada standard library hanya menggunakan karakter ASCII dan sebisa mungkin menggunakan kata-kata dalam Bahasa Inggris. Proyek yang menggunakan python 3 didorong untuk menggunakan standar yang sama. Lihat PEP 3131.

Import

Saat melakukan import library, lakukan import setiap library pada baris berbeda.

```
1. Yes: import os
2.      import sys
3.
4. No:  import sys, os
```

Kecuali, jika anda memerlukan lebih dari satu sub-library dari library yang sama.

```
1. from subprocess import Popen, PIPE
```

Import umumnya diletakkan pada bagian awal berkas. Setelah komentar dan dokumentasi tentang berkas tersebut (misalnya definisi kelas, dll), sebelum variabel global dan konstanta. Jika memungkinkan, kelompokkan import dalam urutan berikut:

1. Standard Library
2. Library Pihak Ketiga
3. Local/Library spesifik

Setiap grup baiknya dipisahkan oleh sebuah baris kosong.

```
import mypkg.sibling
from mypkg import sibling
from mypkg.sibling import example
```

Kode pada Standard library umumnya dapat menggunakan absolute import. Anda juga dapat mengimpor kelas/sub-library, Anda tentu saja dapat menggunakan pemanggilan berikut:

```
from myclass import MyClass
from foo.bar.yourclass import YourClass
```

Jika ada penamaan kelas yang sama, gunakan pemanggilan secara eksplisit:

```
import myclass
import foo.bar.yourclass
```

saat memanggil, gunakan "myclass.MyClass" dan "foo.bar.yourclass.YourClass".

```
from <module> import *
```

Wildcard imports seperti tertulis, sedapat mungkin dihindari untuk mengatasi ambiguitas dan ketidaktahuan tentang modul apa yang di-import.

Tanda Petik

Petik tunggal (') dan petik ganda (") dianggap sama oleh Python, dan tidak memiliki preferensi khusus untuk penggunaannya. Hal ini dikarenakan ada kemungkinan string yang memuat salah satunya. Anda disarankan untuk menggunakan salah satunya secara konsisten.

Docstring (dokumentasi kode/fungsi/method) pada Python didefinisikan dengan tiga tanda petik, disarankan tanda petik ganda (""") pada awal dan akhir statement docstring.

Whitespace pada Expressions dan Statements

> Wajib dihindari penambahan whitespace yang tidak perlu.

Antara kurung, kurawal, kurung siku.

```
2. No:  spam( ham[ 1 ], { eggs: 2 } )
```

Setelah koma, tanpa argumen lain setelahnya.

```
1. Yes: foo = (0,)
2. No:  bar = (0, )
```

Sebelum koma, titik dua, atau titik koma.

```
1. Yes: if x == 4: print x, y; x, y = y, x
2. No:  if x == 4 : print x , y ; x , y = y , x
```

Namun, jika Anda menggunakan titik dua/colon sebagai slice (sub-list), pastikan ia memiliki spasi/whitespace yang sama pada kedua sisinya.

```
1. Yes:
2. ham[1:9], ham[1:9:3], ham[:9:3], ham[1::3], ham[1:9:]
3. ham[lower:upper], ham[lower:upper:], ham[lower::step]
4. ham[lower+offset : upper+offset]
5. ham[: upper_fn(x) : step_fn(x)], ham[:: step_fn(x)]
6. ham[lower + offset : upper + offset]
7.
8. No:
9. ham[lower + offset:upper + offset]
10. ham[1: 9], ham[1 :9], ham[1:9 :3]
11. ham[lower : : upper]
12. ham[ : upper]
```

Saat memberikan parameter pada fungsi, sebelum kurung tidak boleh ada spasi.

```
1. Yes: spam(1)
2. No:  spam (1)
```

Saat memberikan parameter/index pada list, sebelum kurung siku tidak boleh ada spasi.

```
1. Yes: dct['key'] = lst[index]
2. No:  dct ['key'] = lst [index]
```

Saat membuat assignment pada variabel, sebaiknya tidak menambahkan whitespace yang tidak pe

```
y = 2
long_variable = 3

No:
x             = 1
y             = 2
long_variable = 3
```

Rekomendasi Lainnya

Hindari menambahkan whitespace di belakang statement apapun, utamanya di statement akhir dalam sebuah baris, karena whitespace tersebut tidak mudah dilihat.

Biasakan untuk menambahkan satu spasi baik di kiri maupun kanan untuk operasi berikut:

1. Assignment (=),
2. Augmented assignment (+=, -=etc.),
3. Comparisons (==, <, >, !=, <>, <=, >=, in, not in, is, is not),
4. Booleans (and, or, not).

Jika operator dengan berbagai tingkatan prioritas digunakan, letakkan whitespace pada operator-operator dengan prioritas terendah. Namun Anda juga dapat menyesuaikannya sendiri.

> Catatan: jangan pernah menggunakan >1 spasi dan gunakan spasi yang sama baik di sebelah kiri maupun kanan dari operator-operator binary Anda.

```
Yes:
i = i + 1
submitted += 1
x = x*2 - 1
hypot2 = x*x + y*y
c = (a+b) * (a-b)

No:
i=i+1
submitted +=1
x = x * 2 - 1
hypot2 = x * x + y * y
c = (a + b) * (a - b)
```

Dalam sebuah kode Python, Anda diajak untuk memastikan kode Anda terbaca oleh programmer lain. Salah satu caranya adalah dengan menggunakan fitur komentar untuk memberitahu fungsi atau informasi lain terkait kode Anda. Pastikan komentar Anda ter-update dan tidak mengalami kontradiksi dengan kode yang ada.

Umumnya, komentar dituliskan dalam kalimat utuh dengan memperhatikan penulisan (huruf besar di awal kalimat, huruf kecil saat diawali dengan identifier atau variabel, dan diakhiri titik di akhir kalimat). Anda juga bisa menggabungkan beberapa kalimat menjadi blok komentar dengan menambah dua spasi saat berganti kalimat dalam satu paragraf, kecuali pada kalimat terakhir.

Jika memungkinkan, tuliskan komentar dalam bahasa Inggris, kecuali Anda yakin bahwa pembaca komentar ini dipastikan mengerti bahasa Anda.

Blok Komentar

Blok komentar umumnya digunakan untuk menjelaskan fungsi utuh atau sub-fungsi yang mengikuti/berada di bawahnya. Blok komentar diindentasi setara dengan kode yang dijelaskan. Setiap barisnya diawali dengan # dan sebuah spasi serta setiap paragrafnya dimulai pada baris baru.

Komentar Inline

Komentar Inline pada Python umumnya diletakkan pada baris yang sama dengan kode. Umumnya dipisahkan dan dirapikan dengan jarak dua spasi dari kode yang dimaksud, diawali # dan sebuah spasi. Komentar inline dapat juga digunakan di atas baris yang ingin diberikan komentar, agar tidak mengurangi jumlah karakter yang dapat dituliskan dalam sebuah baris. Untuk semua jenis komentar, jangan menuliskan komentar untuk hal yang sudah langsung dapat dibaca dari kodenya, seperti contoh berikut:

Tidak disarankan:

```
1.  x = x + 1                 # Tambahkan x
```

Disarankan (kontekstual):

```
1.  x = x + 1                 # Mengakomodasi layar ukuran Z
```

Dokumentasi

Guideline untuk menuliskan dokumentasi (*docstring*) yang baik tersedia di PEP 257. Kuncinya:

- Docstring tidak diwajibkan pada method yang tidak bersifat public, namun Anda disarankan menambahkan komentar tentang Apa saja yang dilakukan fungsi/modul ini beserta informasi lainnya yang mungkin diperlukan. Komentar ini diletakkan setelah baris def.

PEP 257 memberikan panduan detil yang dapat digunakan. Seperti yang sudah-sudah, Anda disarankan untuk menutup sebuah docstring yang lebih dari satu baris, pada baris baru berikutnya:

```
1. """Return a foobang
2. Optional plotz says to frobnicate the bizbaz first.
3. """
```

Untuk docstring satu baris, Anda disarankan untuk meletakkan penutup """ - nya pada baris yang sama.

Meskipun secara sintaksis Anda dapat menggantikan 3-tanda-kutip-dua """ dengan 3-tanda-kutip-satu ''', untuk penulisan komentar multi-baris, tetapi PEP 257 memberikan panduan gunakan 3-tanda-kutip-dua untuk dokumentasi (*docstring*).

Sebelumnya

Selanjutnya