

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií



reCommers

Metodika štandardov pre písanie kódu

Tím: 03 - reCommers

Vedúci tímu: Ing. Ivan Srba, PhD.

Vyhotovil: Jozef Pazúrik

Dátum poslednej zmeny: 11.05. 2018

Verzia: 1.8

Obsah

1. Úvod	3
2. Pravidlá	3
2.1. Odsadenie textu:	3
2.2. Používanie bodkočiarky:	3
2.3. Maximálna dĺžka riadkov a funkcií:	3
2.4. Prázdne riadky:	4
2.5. Kódovanie súborov:	4
2.6. Používanie importov:	4
2.7. Medzery a biele znaky	4
2.8. Názvy	5
2.9. Globálne premenné	6

1. Úvod

Táto metodika sa zaoberá pravidlami písania zdrojového kódu za účelom prehľadnosti a jednotného formátovania kódov. Dodržiavanie jednotného štýlu v kóde je kľúčové pre zjednodušenie orientácie v zdrojových súboroch ako aj ľahšie pochopenie kódu budúcim čitateľom. Táto metodika vychádza zo známeho štandardu pre python PEP-8 <https://www.python.org/dev/peps/pep-0008>, najmä však z jeho skrátenej verzie od googlu: <https://google.github.io/styleguide/pyguide.html>

2. Pravidlá

V každom prípade sa riadte týmto štandardom:

<https://google.github.io/styleguide/pyguide.html>. Táto kapitola zahŕňa len stručné zhrnutie vybraných pravidiel.

Všetky názvy funkcií aj premenných píšete v angličtine.

2.1. Odsadenie textu:

Na odsadenie používajte 4 medzery nie tabulátor.

2.2. Používanie bodkočiarky:

Nepíšete bodkočiarku na konci príkazu a nepoužívajte bodkočiarku na oddelenie viacerých príkazov v jednom riadku.

2.3. Maximálna dĺžka riadkov a funkcií:

Riadok nesmie byť dlhší ako 80 znakov.

Maximálna dovolená dĺžka jednej funkcie je 30 riadkov. Pokiaľ je nejaká dlhšia, rozdeľte ju.

3.3. Delenie dlhých výrazov:

Pre dlhé aritmetické výrazy, ktoré je potrebné rozdeliť na viac riadkov, používajte operátor pred výrazom na začiatku riadku, nie opačne. Príklad:

```
income = (gross_wages
          + taxable_interest
          + (dividends - qualified_dividends)
          - ira_deduction
          - student_loan_interest)
```

2.4. Prázdné riadky:

Globálne funkcie a triedy oddeľujte dvoma prázdnyimi riadkami, funkcie v rámci triedy jedným riadkom. V rámci funkcie používajte prázdné riadky iba ak chcete oddeliť logické celky.

2.5. Kódovanie súborov:

Na kódovanie súborov vždy používajte UTF-8.

2.6. Používanie importov:

Imports musia byť vždy na začiatku súboru. Pri importovaní viacerých modulov, každý import musí byť na osobitnom riadku:

```
Yes: import os
      import sys
No:  import sys, os
```

Pri importovaní dodržujte nasledovné poradie:

- štandardné knižnice,
- knižnice tretích strán,
- lokálne súbory aplikácie.

Medzi jednotlivými skupinami importov vložte prázdny riadok. Vždy používajte úplnú cestu k modulu, nie relatívnu. Napr.

```
Yes: from sound.effects import echo
No:  from .effects import echo
```

2.7. Medzery a biele znaky

Nepoužívajte zbytočne medzery tam kde nie sú potrebné. Napr.

```
Yes: spam(ham[1], {eggs: 2})
No:  spam( ham[ 1 ], { eggs: 2 } )
Yes: foo = (0,)
No:  bar = (0, )
Yes: if x == 4: print x, y; x, y = y, x
No:  if x == 4 : print x , y ; x , y = y , x
```

Nepokúšajte sa o vertikálne zarovnanie viacerých príkazov pod sebou. Horšie sa to potom neskôr upravuje ak sa má zarovnanie zachovať:

```
Yes:
    foo = 1000 # comment
    long_name = 2 # comment that should not be aligned

    dictionary = {
        'foo': 1,
```

```
'long_name': 2,
}
```

No:

```
foo      = 1000  # comment
long_name = 2    # comment that should not be aligned
```

```
dictionary = {
    'foo'      : 1,
    'long_name': 2,
}
```

Aritmetické operátory (+, -, *, /, +, -, ' = ...) a porovnávacie operátory (==, <, >, !=, <>, <=, >=, in, not in, is, is not) oddeľujte medzerami, jednou pred a druhou za operátorom. V prípade zložených výrazov dávajte medzery okolo operátorov s najnižšou prioritou:

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)
```

2.8. Názvy

V skratke (tl;dr):

module_name, package_name, ClassName, method_name, ExceptionName, function_name, GLOBAL_CONSTANT_NAME, global_var_name, instance_var_name, function_parameter_name, local_var_name.

Podrobnejšie:

Názvy modulov, balíčkov, funkcií, parametrov funkcií a premenných píšete v tvare: lower_case_with_underscores – malými písmenami oddelenými podčiarkovníkom.

Názvy tried, typov a výnimiek píšete v *Camel/Case* tvare. Teda spojené slová, kde každé začína veľkým písmenom.

Privátne/interné (v rámci jedného modulu) premenné a funkcie začínajte podčiarkovníkom: _private_function_name, _internal_variable_name, _InternalClassName, _INTERNAL_GLOBAL_VARIABLE ...

Nikdy nepoužívajte :

- Ako názov premennej znak 'l' (malé L), 'I' (veľké i) alebo O (veľké o).
- pomlčku na oddelenie slov v názve.
- Dva podčiarkovníky na začiatku alebo konci názvu - rezerované názvy jazyka Python

2.9. Globálne premenné

Nepoužívajte globálne premenné, pokiaľ je to možné. Takmer vždy je lepšie použiť premennú triedy.