

## Форматирование

- Используйте 4 пробела для отступов.
- Никогда не смешивайте пробелы и символы табуляции.
- Не делайте строки длиннее 79 символов. Разделяйте строки на несколько с помощью заключения выражения в скобки. Этот способ лучше, чем использование обратного слэша. Делайте соответствующий отступ при переносе. При разбиении строки около бинарного оператора делайте разрыв **после** него.

```
var = (a * 10 +  
      b / 15)      # Правильно  
var = a * 10 \  
      + b / 15     # Неверно
```

- Пишите `import` каждого модуля в отдельной строке
- Располагайте все `import`'ы в верхней части файла перед любыми глобальными объявлениями.
- Не используйте лишних пробелов в следующих ситуациях:

- Сразу после скобок.

```
spam(ham[1], {eggs: 2})      # Правильно  
spam( ham[ 1 ], { eggs: 2 } ) # Неверно
```

- Перед запятой, точкой с запятой, двоеточием.

```
if x == 4: print x, y; x, y = y, x      # Правильно  
if x == 4 : print x , y ; x , y = y , x # Неверно
```

- Перед открывающей круглой скобкой при вызове функции и перед открывающей квадратной скобкой, обозначающей индексирование или срез.

```
spam(kind['key'], lst[1:3])      # Правильно  
spam (kind ['key'], lst [1:3])  # Неверно
```

- Всегда окружайте следующие бинарные операторы ровно одним символом пробела с каждой стороны:
  - присваивания (`=`, `+=`, `-=` и т.д.),
  - сравнения (`==`, `<`, `>`, `!=`, `<>`, `<=`, `>=`, `in`, `not in`, `is`, `is not`),
  - логические (`and`, `or`, `not`),
  - арифметические (`+`, `-`, `*`, `/`, `//`).

- Не располагайте несколько инструкций в одной строке. Разнесите их по разным строкам.

```
x = 3      # Правильно  
func(10)  
x = 3; func(10) # Неверно
```

- Не располагайте блок из нескольких инструкций на той же строке сразу после двоеточия (после `if`, `while` и т.д.).

## Комментарии

- Комментарии, противоречащие коду, хуже, чем их отсутствие.
- Располагайте однострочные комментарии после кода в той же строке и отделяйте их от кода не менее, чем двумя пробелами. Комментарии должны начинаться с `#` и одного пробела.

## Имена

- Не используйте символы `'l'`, `'0'`, и `'I'` как имена переменных. В некоторых шрифтах они могут быть очень похожи на цифры.
- Имена переменных и функций должны содержать только маленькие буквы. Слова разделяются символами подчёркивания. Примеры:

```
name , name_from_several_words
```

- Имена констант должны содержать только заглавные буквы. Слова разделяются символами подчёркивания. Примеры:

```
NAME , NAME_FROM_SEVERAL_WORDS
```

## Программирование

- Сравнение с такими значениями как `None` должно осуществляться с помощью операторов `is` и `is not`, а не операторами сравнения.
- Используйте методы `''.startswith()` и `''.endswith()` вместо срезов для проверки начала и конца строк.
- Для последовательностей (строк, списков, кортежей) используйте факт, что пустая последовательность эквивалентна `False`.
- Не проверяйте переменные логического типа на `True` и `False` используя `==`.

## Функции

- Не ставьте пробелы вокруг знака `=` при передаче функции именованных аргументов.

```
f (5 , x=7)      # Правильно
f (5 , x = 7)   # Неверно
```

- Разделяйте определения функций двумя пустыми строками.
- Разделяйте тело функции на логические части одной пустой строкой.

## Исключения из правил

Перечисленные выше правила имеет смысл нарушать в тех случаях, когда следование им понижает читаемость кода даже для людей, привыкших к этим правилам. Трижды подумайте, прежде чем делать это.