

Требования к оформлению кода в параллели С'

Форматирование

- Используйте 4 пробела для отступов. Не делайте отступов в два пробела.
- Пишите `import` каждого модуля в отдельной строке
- Располагайте все `import`'ы в верхней части файла перед любыми глобальными объявлениями.
- Не используйте конструкцию `from ... import *`
- Скобки не отделяются пробелами с внутренней стороны. Между функцией и ее аргументами пробел не ставится

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

- Перед запятой, двоеточием пробел не ставится, после — ставится.

```
if x == 4:
    print x, y
    x, y = y, x      # Правильно
if x == 4 :
    print x , y
    x , y = y , x   # 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', 'O', и '1' 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_with_several_words_in_it
```

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

```
NAME, NAME_WITH_SEVERAL_WORDS_IN_IT
```

— Давайте переменным говорящие английские имена, не используйте транслит.

```
num_letters = int(input())      # Правильно  
kolvo_bukv = int(input())      # Неверно
```

Программирование

— Для последовательностей (строк, списков, кортежей) используйте факт, что пустая последовательность эквивалентна `False`.

— Не проверяйте переменные логического типа на `True` и `False` используя `==`.

Функции

— Разделяйте определения функций двумя пустыми строками.

— Разделяйте тело функции на логические части одной пустой строкой.