***Правила написания кода на ЯП Python***

При написании кода на языке программирования python вы должны руководствоваться стандартом написания кода PEP8. Этот документ описывает соглашение о том, как писать код для языка python, включая стандартную библиотеку, входящую в состав python.

Ключевая идея такова: код читается намного больше раз, чем пишется. Собственно, рекомендации о стиле написания кода направлены на то, чтобы улучшить читаемость кода и сделать его согласованным между большим числом проектов.

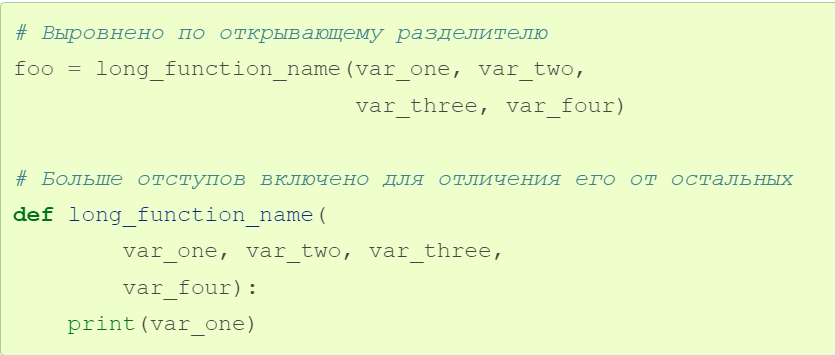
Ниже перечислены некоторые пункты, для полного ознакомления вам необходимо прочитать полное руководство по написанию кода на python.

***Внешний вид кода***

**Отступы**

Используйте 4 пробела на каждый уровень отступа. Продолжительные строки должны выравнивать обернутые элементы либо вертикально, используя неявную линию в скобках (круглых, квадратных или фигурных), либо с использованием висячего отступа. При использовании висячего отступа следует применять следующие соображения: на первой линии не должно быть аргументов, а остальные строки должны четко восприниматься как продолжение линии.

Правильно



Неправильно

Изображение выглядит как текст

Автоматически созданное описание

**Табуляция или пробелы?**

Разрешается использовать либо пробелы, либо табуляцию, но Python 3 запрещает смешивание табуляции и пробелов в отступах. Поэтому договоримся использовать только табуляцию.

**Максимальная длина строки**

Ограничьте длину строки максимум 79 символами. Для более длинных блоков текста с меньшими структурными ограничениями (строки документации или комментарии), длину строки следует ограничить 72 символами. Ограничение необходимой ширины окна редактора позволяет иметь несколько открытых файлов бок о бок, и хорошо работает при использовании инструментов анализа кода, которые предоставляют две версии в соседних столбцах.

**Пустые строки**

Отделяйте функции верхнего уровня и определения классов двумя пустыми строками. Определения методов внутри класса разделяются одной пустой строкой. Дополнительные пустые строки возможно использовать для разделения различных групп похожих функций. Пустые строки могут быть опущены между несколькими связанными однострочниками (например, набор фиктивных реализаций). Используйте пустые строки в функциях, чтобы указать логические разделы.

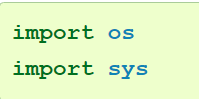
**Кодировка исходного файла**

Кодировка Python должна быть UTF-8.

**Импорты**

Каждый импорт, как правило, должен быть на одной строке.

Правильно



Неправильно

Изображение выглядит как текст

Автоматически созданное описание

В то же время можно писать так:

Изображение выглядит как текст

Автоматически созданное описание

Импорты всегда помещаются в начале файла, сразу после комментариев к модулю и строк документации, и перед объявлением констант. Импорты должны быть сгруппированы в следующем порядке:

* импорты из стандартной библиотеки
* импорты сторонних библиотек
* импорты модулей текущего проекта

Вставляйте пустую строку между каждой группой импортов.

Рекомендуется абсолютное импортирование, так как оно обычно более читаемо и ведет себя лучше (или, по крайней мере, даёт понятные сообщения об ошибках).

**Именование переменных и функций**

При именовании переменных и функций следует руководствоваться исключительно назначением соответствующей переменной или функцией. При именовании использовать прописные латинские буквы в формате snake\_case. Если функция используется для получения какого-то значения использовать префикс get, если устанавливает какое-то значение – set. Для объявления констант использовать UPPER\_SNAKE\_CASE.

**Именование классов**

Для именования классов использовать CapWords, классы именовать в единственном числе.