

# Спецификация PEP 8

**№ урока:** 8 **Курс:** Python Advanced

**Средства обучения:** Python3.6, PyCharm

## Обзор, цель и назначение урока

Получение знаний спецификации PEP 8 и приобретения навыков писать простой и читаемый код.

## Изучив материал данного занятия, учащийся сможет:

- Понимать основные цели и причины создания спецификаций PEP.
- Писать более простой и понятный код, основанный на лучших практиках данной спецификации.

## Содержание урока

1. Понятия и определения.
2. Основные цели и причины создания спецификаций PEP.
3. Примеры кода из реальных проектов.
4. Применение и изучение оформления различных конструкций из PEP 8 на практике.

## Резюме

- PEP 8 является стандартом оформления кода на языке Python. Данная спецификация описывает рекомендации по оформлению кода. Считается, что каждый разработчик на языке Python должен придерживаться этих правил. Однако, в реальных проектах, многие создают свои собственные так называемые style-guide-ы в рамках компаний и следуют им, что является распространенной практикой.
- Помимо PEP 8 существует ряд других спецификаций, которые описывают другие аспекты программирования на Python. Например, спецификация PEP 7 описывает правила оформления кода CPython, в случае если вы пишете свои собственные C-расширения для языка Python. В качестве еще одного примера, существует специальный модуль, который связан с нашим предыдущим уроком «Типизация в Python: модуль typing». Для данного модуля была разработана собственная спецификация 484, в которой можно найти советы по использованию и все возможности данного модуля.
- Язык Python не имеет какой-то жесткой системы правил, которые утвердились во всех компаниях и во всем сообществе. Но за долгие годы написания кода на Python возникает необходимость создавать некоторые стандарты, которым бы следовали программисты. Если взять в пример Java, то у данного языка существуют только лишь один свод общих правил оформления кода, к которому придерживаются все Java-разработчики. Однако, взяв два или три проекта, написанными разными компаниями или разработчиками из разных команд, мы увидим, что кодовая база и её стилизация разительно отличаются. Это и становится причиной того, чтобы придерживаться некоторых рекомендаций по стилизации Python кода, чтобы наш код был более понятен и прост для чтения, ведь, как правило, код больше читают, нежели пишут.
- Необходимо понимать, что в написании данных стандартов участвовал и сам создатель языка- Python- Гвидо ван Россум, что, как минимум, является показателем целесообразности данных рекомендаций. Но также, надо учитывать, что PEP 8 не является исчерпывающим руководством по оформлению кода. Зачастую, в практике, вы будете

встречать множество неоднозначных ситуаций, в котором сами будете принимать решения. Однако, основываться вы будете на основных идеях PEP 8.

### Закрепление материала

- Что такое PEP 8 и существуют ли еще спецификации?
- Зачем нужны правила оформления кода?
- Является ли PEP 8 исчерпывающим руководством?

### Дополнительное задание

#### Задание

Создайте класс пользователя (ФИО, дата рождения), к каждому методу составьте docstrings и постарайтесь максимально следовать правилам, описанным в PEP 8. Вы можете создавать любое количество методов, с целью потренироваться, но как минимум класс должен содержать:

1. `get_short_name`
2. `get_full_name`
3. `get_age`
4. `get_initials`
5. `set_full_name` (принимает строку вида «Петров Иван Иванович», которую необходимо разбить по соответствующим полям. Условимся, что порядок будет следующим: Фамилия Имя Отчество)

Создайте код, который будет использовать все разработанные вами методы и поля, соблюдая PEP 8.

### Рекомендуемые ресурсы

Официальный сайт Python – список спецификаций

<https://www.python.org/dev/peps/>

Официальный сайт Python – спецификация PEP 8

<https://www.python.org/dev/peps/pep-0008/>