КУРС “Программирование на PYTHON”
27.10.2018 ЛЕКЦИЯ 05

BASE LINE

- PEP8 - правила красивого кода
- MVC - проектирование приложений
- Рекурсия
- GIT - храним и делимся

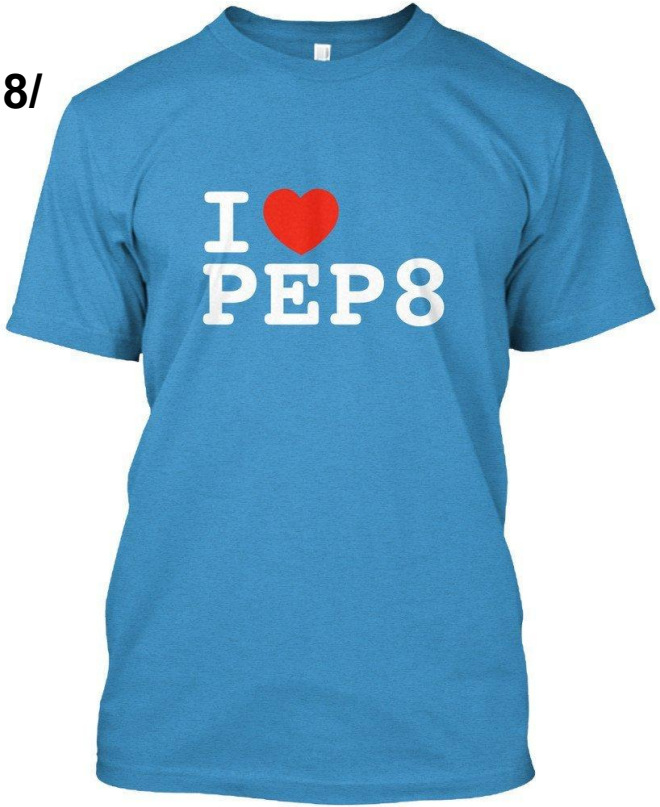
НЕ НАДО ЭТО ЗАПОМИНАТЬ!!!

PEP8

https://www.python.org/dev/peps/pep-0008/
https://geekbrains.ru/posts/pep8

Python Enhancement Proposals

Предложения по развитию Python



PEP8

Используйте 4 пробела на каждый уровень отступа.

PEP8

Пробелы - самый предпочтительный метод отступов.

Python 3 запрещает смешивание табуляции и пробелов в отступах.

PEP8

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.

Предпочтительный способ переноса длинных строк является использование подразумеваемых продолжений строк Python внутри круглых, квадратных и фигурных скобок.

Это предпочтительнее использования обратной косой черты для продолжения строки.

PEP8

Отделяйте функции верхнего уровня и определения классов двумя пустыми строками.

PEP8

Используйте пустые строки в функциях, чтобы указать логические разделы.

PEP8

Кодировка Python должна быть UTF-8.

PEP8

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

Импорты всегда помещаются в начале файла, сразу после комментариев к модулю и строк документации, и перед объявлением констант.

1. **импорты из стандартной библиотеки**
2. **импорты сторонних библиотек**
3. **импорты модулей текущего проекта**

PEP8

Избегайте использования пробелов в следующих ситуациях:

- **Непосредственно внутри круглых, квадратных или фигурных скобок.**
- **Непосредственно перед запятой, точкой с запятой или двоеточием.**
- **Сразу перед открывающей скобкой, после которой начинается список аргументов при вызове функции.**
- **Использование более одного пробела вокруг оператора присваивания (или любого другого) для того, чтобы выровнять его с другим.**

PEP8

Всегда окружайте эти бинарные операторы одним пробелом с каждой стороны:

присваивания (=, +=, -= и другие),

сравнения (==, <, >, !=, <>, <=, >=, in, not in, is, is not),

логические (and, or, not).

PEP8

Если используются операторы с разными приоритетами, попробуйте добавить пробелы вокруг операторов с самым низким приоритетом.

PEP8

Не используйте пробелы вокруг знака =, если он используется для обозначения именованного аргумента или значения параметров по умолчанию.

PEP8

Иногда можно писать тело циклов while, for или ветку if в той же строке, если команда короткая, но если команд несколько, никогда так не пишите.

PEP8

Программисты, которые не говорят на английском языке, пожалуйста, пишите комментарии на английском, если только вы не уверены на 120%, что ваш код никогда не будут читать люди, не знающие вашего родного языка.

PEP8

Комментарии в строке с кодом не нужны и только отвлекают от чтения, если они объясняют очевидное.

PEP8

Модули должны иметь короткие имена, состоящие из маленьких букв. Можно использовать символы подчеркивания, если это улучшает читабельность.

Имена функций должны состоять из маленьких букв, а слова разделяться символами подчеркивания — это необходимо, чтобы увеличить читабельность.

PEP8

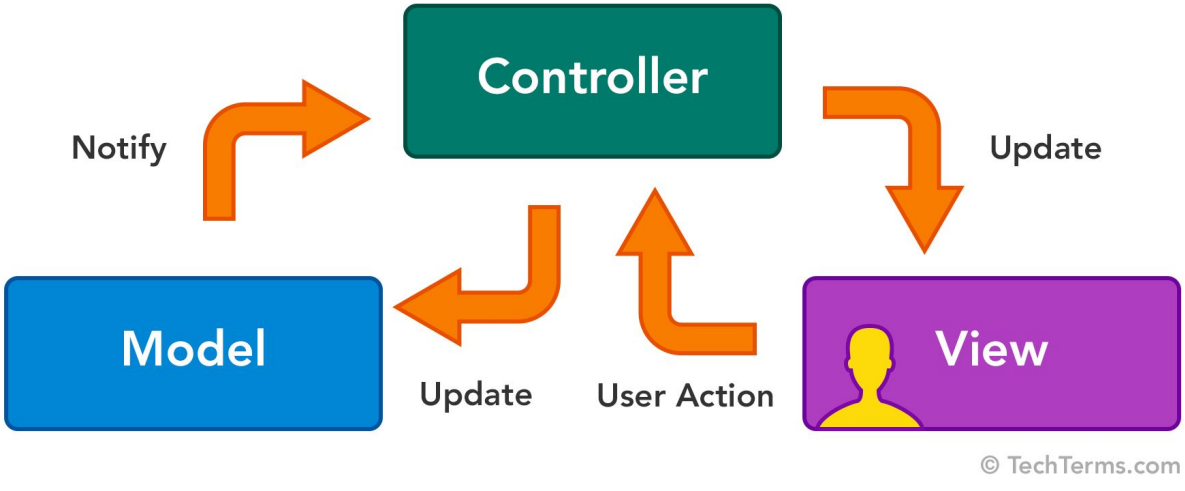
Константы обычно объявляются на уровне модуля и записываются только заглавными буквами, а слова разделяются символами подчеркивания.

PEP8

Когда ресурс является локальным на участке кода, используйте выражение with для того, чтобы после выполнения он был очищен оперативно и надёжно.

MVC

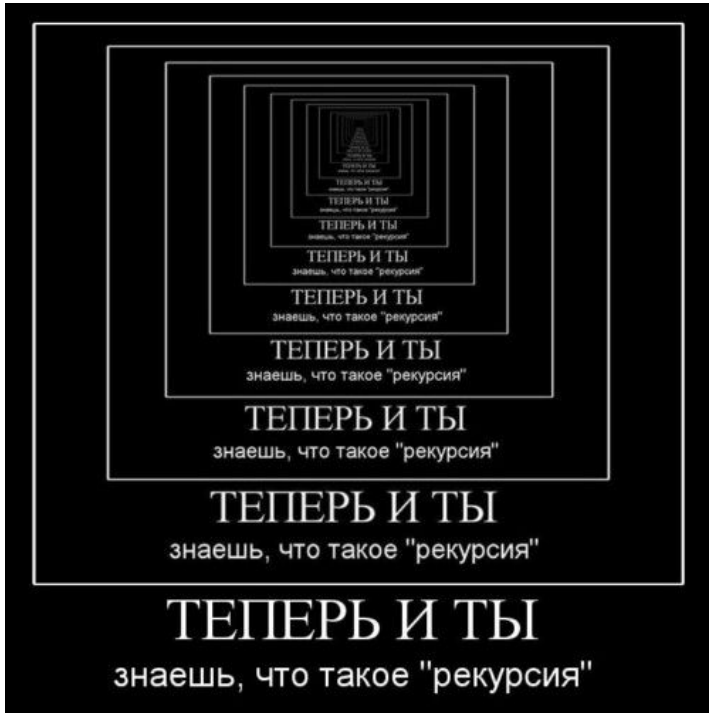
- MODEL - данные
- VIEW - внешний вид
- CONTROLLER - расчеты



РЕКУРСИЯ

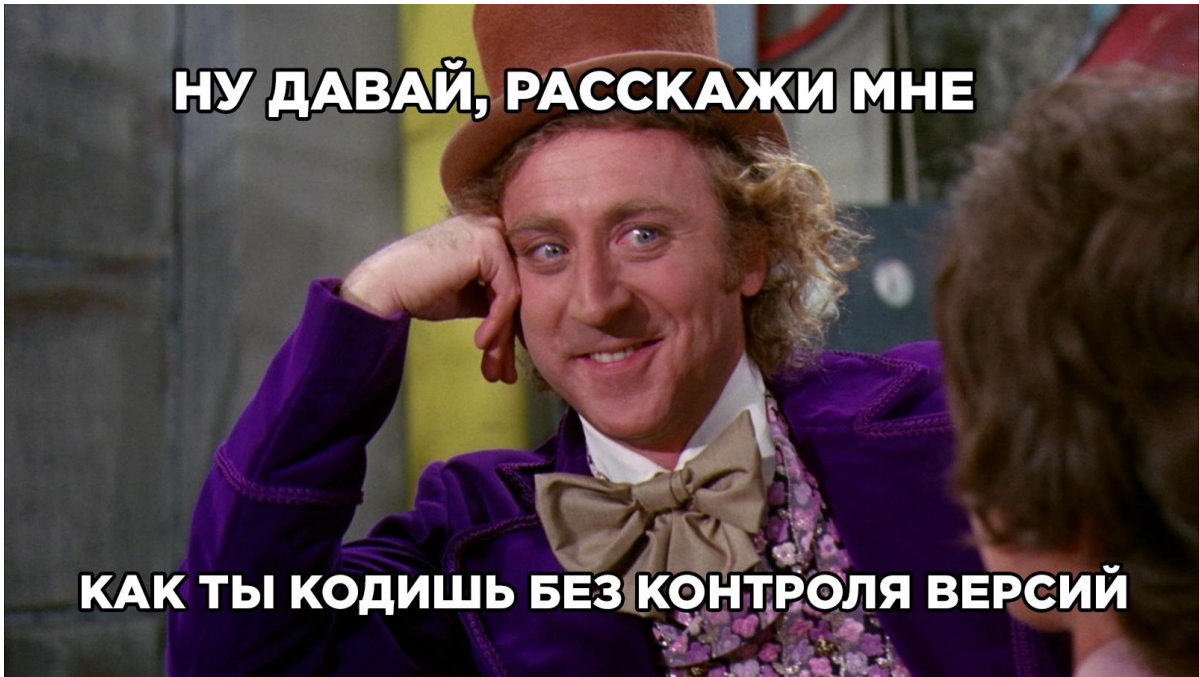

```
def factorial(n):
    res = 1
    for i in range(1, n + 1):
        res *= i
    return res
```

```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)
```



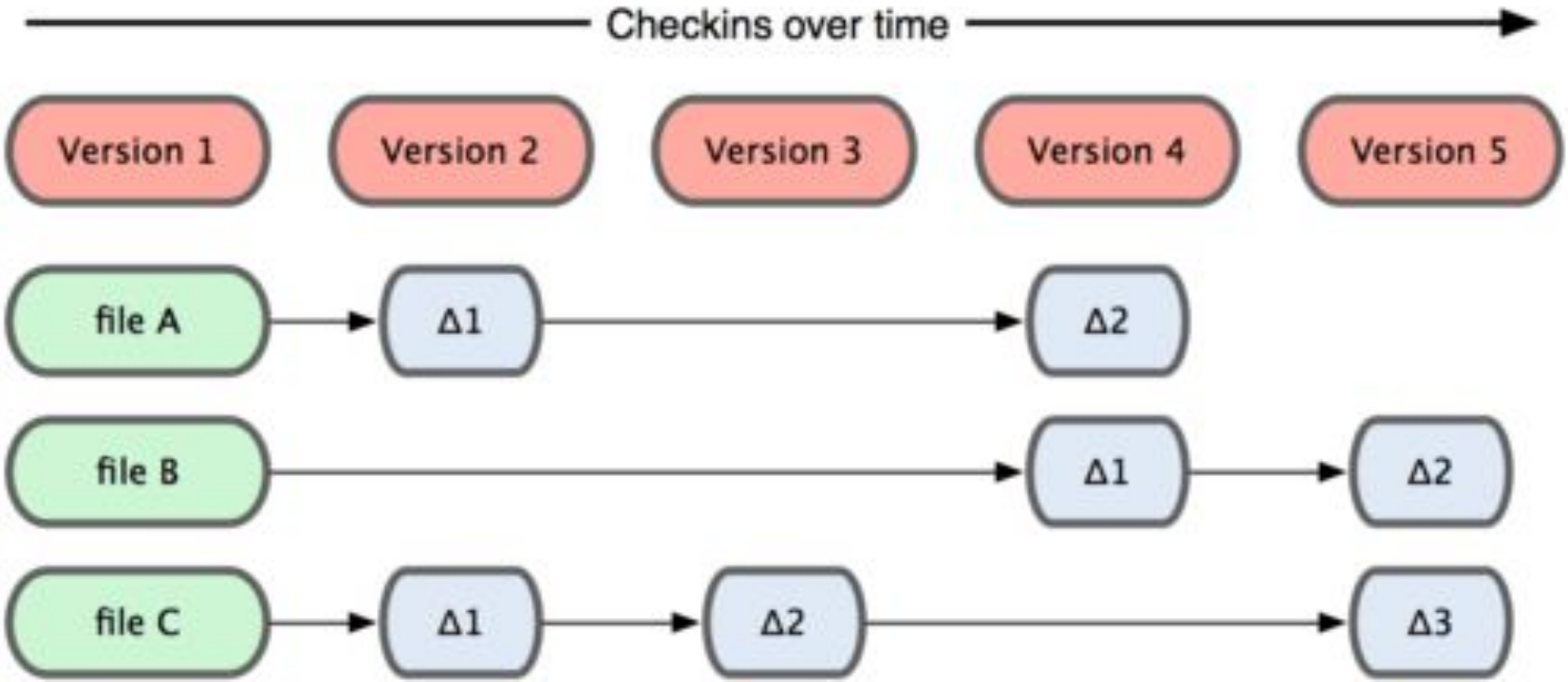
GIT

https://github.com/gorodetskiykp/py_2k18
Распределённая система управления версиями



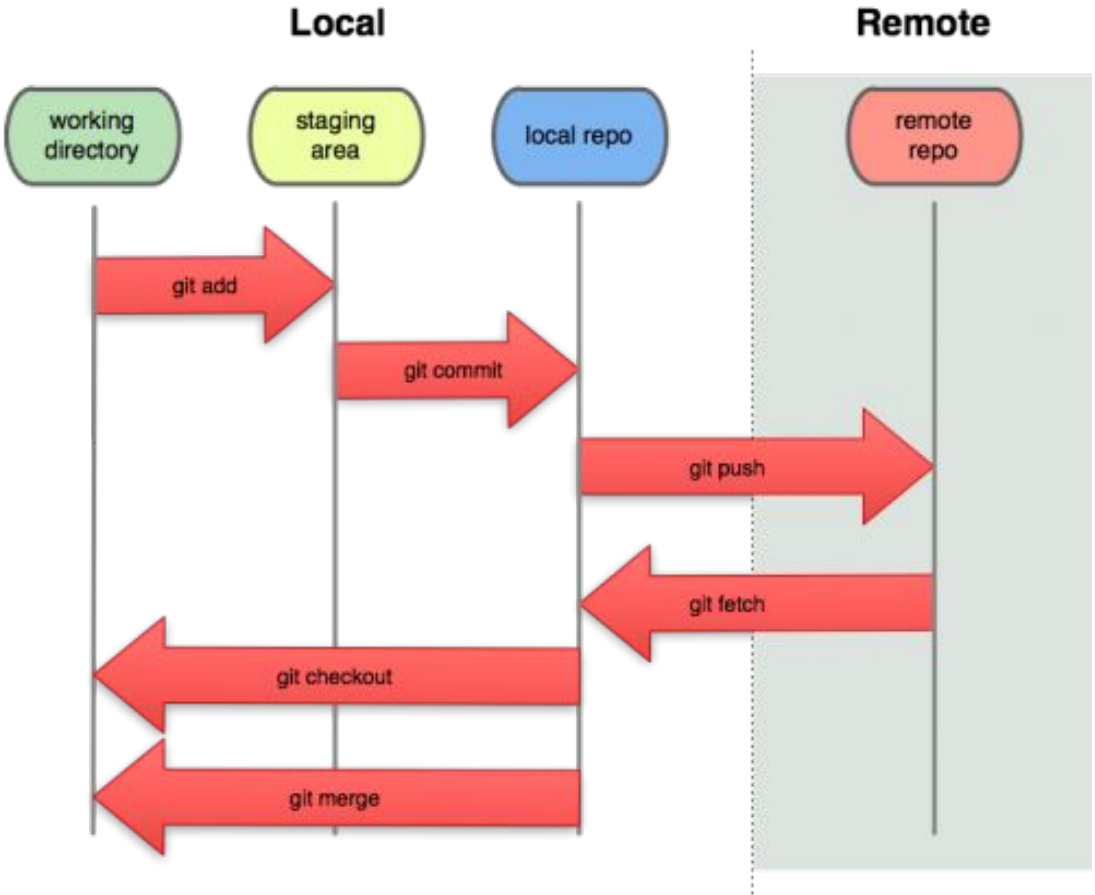
GIT


КП

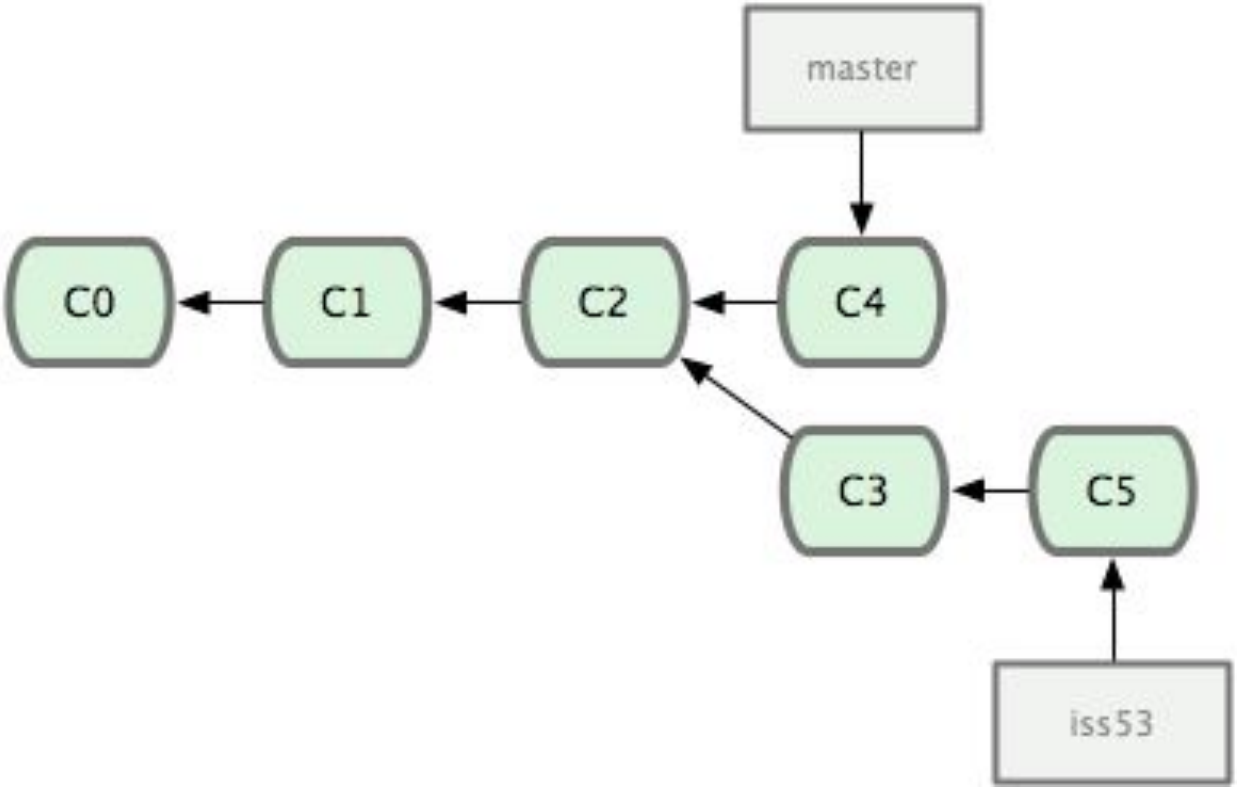

Checkins over time
Version 1
Version 2
Version 3
Version 4
Version 5
file A
Δ1
Δ2
file B
Δ1
Δ2
file C
Δ1
Δ2
Δ3

GIT


КП


Local
Remote
working directory
staging area
local repo
remote repo
git add
git commit
git push
git fetch
git checkout
git merge

GIT



GIT

- git init
- git add
- git clone
- git commit
- git status
- git push
- git pull
- git diff
- git log