# Python code policy

## Общие комментарии

1. Код должен быть простым, ясным и читаемым, в той мере, в которой это возможно для реализации требуемой функциональности.
2. Не делайте что-то на будущее, если нет полной уверенности, что это понадобится. Не должно быть неиспользуемых классов, интерфейсов и методов, логики и пр.
3. Имена должны быть читабельными и понятными, желательно без сокращений, аббревиатур и грамматических ошибок. Исключение – общепринятые сокращения, напр. i, j, k в циклах. Имя переменной number предпочтительнее, чем n или num.
4. Код должен быть **самодокументируемым**. Комментируйте только неочевидные вещи – костыли, воркэраунды, почему именно так сделано. Комментарии должны быть на английском.

## Импорт модулей

Порядок размещения импорта модулей в файлах

1. нативные модули питона (datetime)

пустая строка

2. сторонние модули (pandas)

пустая строка

3. локальные модули

## Именование функций

Используйте правила именования функций: слова в нижнем регистре, разделенные символами подчеркивания.

## Параметры методов

Все параметры методов должны быть типизированы

## Константы

Константы обычно определяются на уровне модуля и пишутся заглавными буквами с символами подчеркивания, разделяющими слова. Например: MAX\_OVERFLOW, TOTAL.

## Документация

Все методы должны сопровождаться краткой документацией в формате:

"""

Method description

:param PARAM\_NAME: parameter description

:return: This line is required only if the method has a return and it describes what the return is

"""

## Пример

import datetime

import pandas as pd

def get\_test\_data(file\_name: str, date: datetime.date) -> pd.DataFrame:

"""

This method when called will return a pandas dataframe from .csv file provided

:param file\_name:

:param date:

:return:

"""

df = pd.read\_csv(file\_name)

return df

## Полезные ссылки

https://peps.python.org/pep-0008/#:~:text=Use%20the%20function%20naming%20rules,invoke%20Python's%20name%20mangling%20rules.