

# Полезные расширения Jupyter Notebook

## Практикум на ЭВМ 2017/2018

Шамшиев Мамат Мамбетович

МГУ имени М. В. Ломоносова, факультет ВМК, кафедра ММП

14 ноября 2017 г.

- Расширение функционала Jupyter Notebook;
- Экономия времени;
- Работа в Jupyter Notebook становится немного легче и приятнее.

Самый простой способ работы с расширениями — установить *jupyter\_contrib\_nbextensions*.

- Включает в себя множество полезных расширений;
- Удобный интерфейс для включения/отключения расширений;
- Каждое расширение сопровождается короткой документацией.

- На панели инструментов Jupyter notebook появляется кнопка, при нажатии которой код в текущей ячейке автоматически приводится к стилю PER8;
- Использует модуль *autoper8.py*;
- Возможна настройка списка исправляемых ошибок (впрочем, не самая удобная).

# Пример использования Autoper8

До применения Autoper8:

```
In [1]: import numpy

def example ( argument):
    ####This is a Long comment. This should be wrapped to fit within 72 characters.
    some_tuple=( 1,2, 3,'a' );
    some_variable={'long':'Long code lines should be wrapped within 79 characters',
'other':[math.pi, 100,200,300,9876543210,'This is a long string that goes on']}
    return (some_tuple, some_variable)
```

И после:

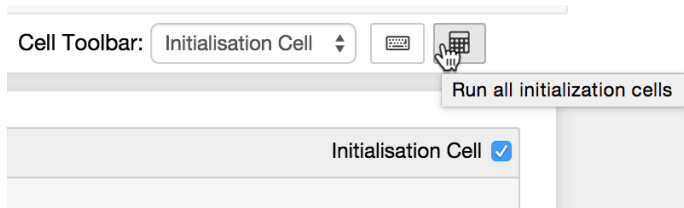
```
In [1]: import numpy

def example(argument):
    # This is a long comment. This should be wrapped to fit within 72
    # characters.
    some_tuple = (1, 2, 3, 'a')
    some_variable = {
        'long': 'Long code lines should be wrapped within 79 characters',
        'other': [
            math.pi,
            100,
            200,
            300,
            9876543210,
            'This is a long string that goes on']]
    return (some_tuple, some_variable)
```

# Расширение Initialization cells

Каждая ячейка может быть отмечена как инициализационная. При стандартных настройках код таких ячеек будет выполнен:

- При нажатии соответствующей кнопки на панели инструментов;
- При готовности ядра, если ноутбук отмечен как «trusted».



Строки кода с одинаковым отступом могут быть «свёрнуты».

```
In [1]: class KNNClassifier:
>     def __init__(self, k, strategy='my_own', metric='euclidean', weights=False, test_block_size=0):↵
>
>     def fit(self, X, y):↵
>
>     def find_kneighbors(self, X, return_distance):↵
>
>     def predict(self, X):↵
```

Отображает когда был выполнен код в ячейке и сколько времени это заняло. Может быть полезным при проведении экспериментов.

```
In [8]: cv_accuracy_k = knn_cross_val_score(X_train, y_train, range(1, 11))  
executed in 2m 13s, finished 19:15:05 2017-11-11
```



Отображает когда был выполнен код в ячейке и сколько времени это заняло. Может быть полезным при проведении экспериментов.

```
In [8]: cv_accuracy_k = knn_cross_val_score(X_train, y_train, range(1, 11))  
executed in 2m 13s, finished 19:15:05 2017-11-11
```