

ВСП 1.1. Постановка задачи: Создание ЭОР на тему «Использование платформы IPython для решения научных и исследовательских задач»

IPython

IPython¹ представляет собой мощный инструмент для работы с языком Python. Базовые компоненты IPython – это интерактивная оболочка для с широким набором возможностей и ядро для Jupyter. Jupyter notebook является графической веб-оболочкой для IPython, которая расширяет идею консольного подхода к интерактивным вычислениям.

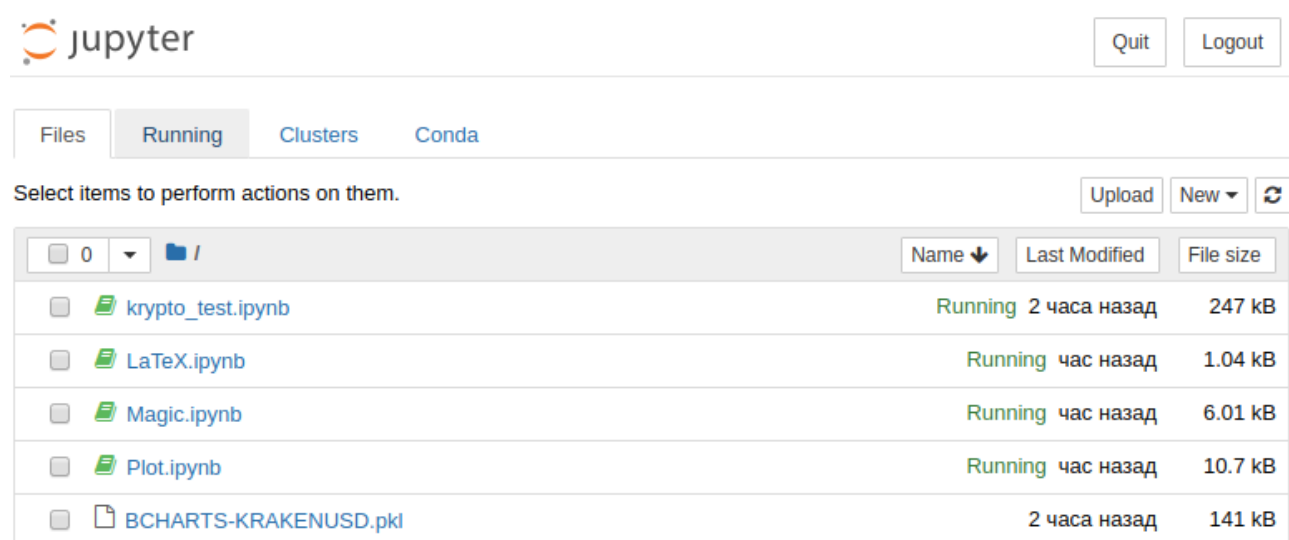
IPython находится в составе Anaconda, и поэтому для его использования необходимо иметь установленную Anaconda.

Запуск данного инструмента можно осуществлять с помощью команды

```
$ jupyter notebook
```

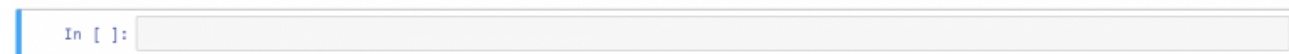
Таким образом запускается локальный сервер, зайти на который можно по адресу localhost:8888.

Перед нами откроется интерактивная страничка, в которой мы и можем начать работать.



Нажав на кнопку New -> Python 3 мы создадим новую запись. В ней мы можем запускать код на Python.

Вводить его нужно в ячейки.



Запускать код можно, нажав на кнопку с изображением стрелки или нажав сочетание клавиш Ctrl+Enter.

Пример запущенного кода

¹ <https://devpractice.ru/python-lesson-6-work-in-jupyter-notebook/>

```
In [3]: import os
import numpy as np
import pandas as pd
import pickle
import quandl
from datetime import datetime

import plotly.offline as py
import plotly.graph_objs as go
import plotly.figure_factory as ff
py.init_notebook_mode(connected=True)

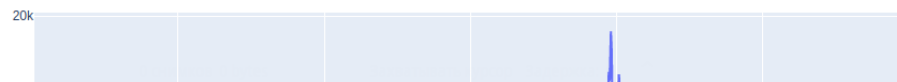
def get_quandl_data(quandl_id):
    '''Download and cache Quandl dataseries'''
    cache_path = '{}.pkl'.format(quandl_id.replace('/', '-'))
    try:
        f = open(cache_path, 'rb')
        df = pickle.load(f)
        print('Loaded {} from cache'.format(quandl_id))
    except (OSError, IOError) as e:
        print('Downloading {} from Quandl'.format(quandl_id))
        df = quandl.get(quandl_id, returns="pandas")
        df.to_pickle(cache_path)
        print('Cached {} at {}'.format(quandl_id, cache_path))
    return df

btc_usd_price_kraken = get_quandl_data('BCHARTS/KRAKENUSD')

btc_usd_price_kraken.head()

btc_trace = go.Scatter(x=btc_usd_price_kraken.index, y=btc_usd_price_kraken['Weighted Price'])
py.iplot([btc_trace])
```

Loaded BCHARTS/KRAKENUSD from cache



*График с помощью
plotly*

Loaded BCHARTS/KRAKENUSD from cache

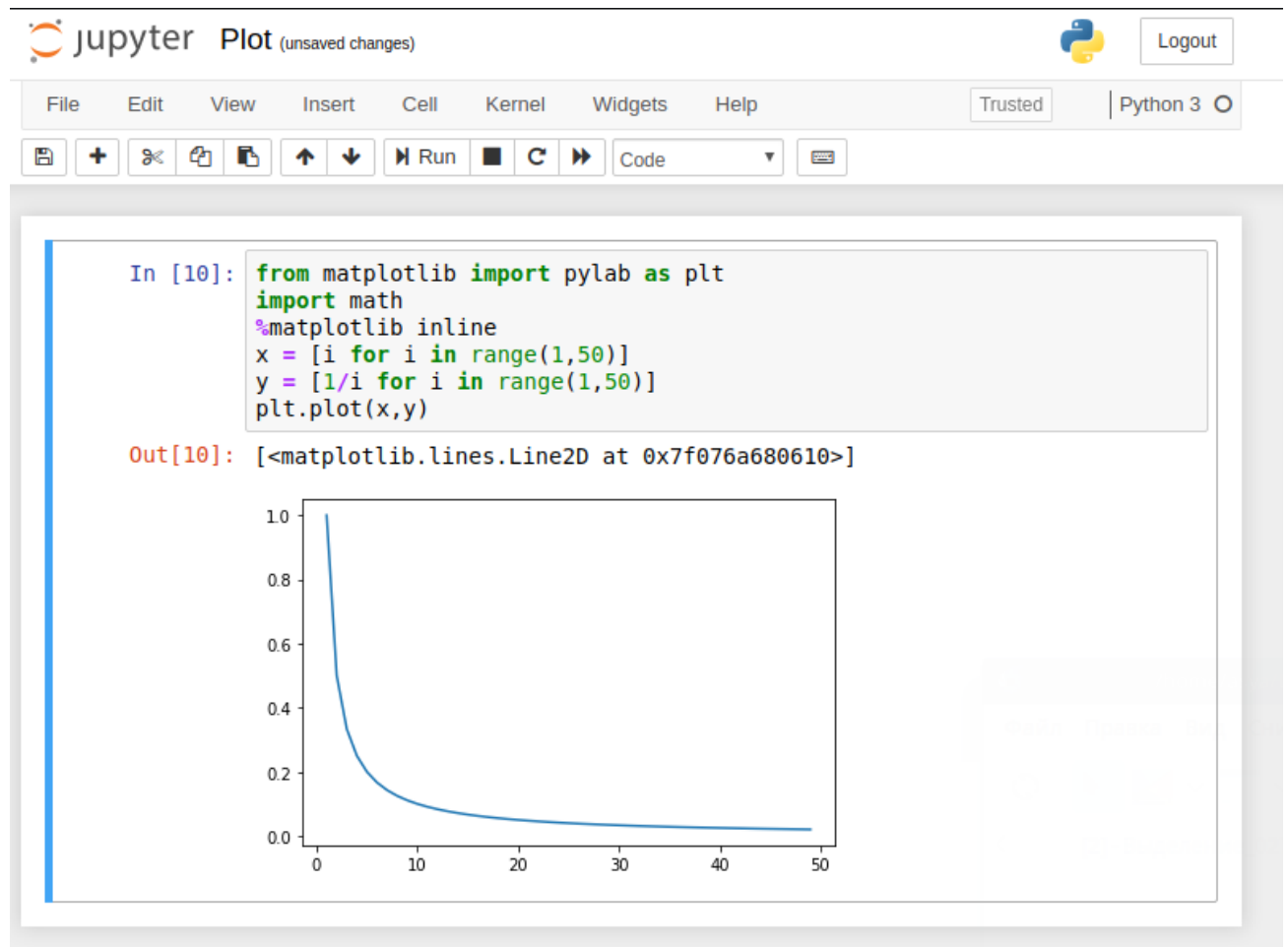


In []:

Или matplotlib.

Примечание: чтобы график вывелся в запись, необходимо написать строку:

%matplotlib inline



Также Jupyter позволяет верстать странички на языке разметки Markdown, используя при этом

LaTeX:

The image shows a Jupyter Notebook interface with the title "LaTeX Last Checkpoint: 6 минут назад (unsaved changes)". The top menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The right side shows a "Trusted" status and "Python 3" as the selected kernel. The toolbar contains icons for saving, adding cells, undo, redo, and running code. The main area displays a Markdown document with the following content:

Пробуем использовать Jupyter

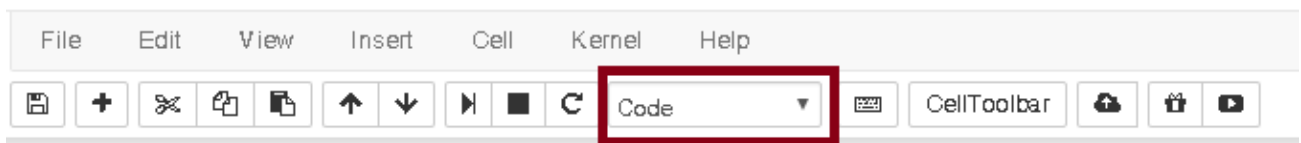
Здесь можно использовать LaTeX

$$y = x^2$$

где $x = 2$

```
In [2]: # и параллельно вставляя код
x = 2
y = x**2
print(y)
4
```

Для того чтобы писать на Md нужно в панели инструментов выставить свойство "Markdown".



Последним хочется отметить "Магию" в JN. Магия — это дополнительные команды, выполняемые в рамках оболочки, которые облегчают процесс разработки. Список команд можно вывести командой

`%lsmagic`

```
In [1]: %lsmagic
```

```
Out[1]: Available line magics:
```

```
%alias %alias_magic %autoawait %autocall %automagic %autosave %bookmark %cat %cd %clear %colors %conda %config %connect_info %cp %debug %dhist %dirs %doctest_mode %ed %edit %env %gui %hist %history %killbgscripts %ldir %less %lf %lk %ll %load %load_ext %loadpy %logoff %logon %logstart %logstate %logstop %ls %lsmagic %lx %macro %magic %man %matplotlib %mkdir %more %mv %notebook %page %pastebin %pdb %pdef %pdoc %pfile %pinfo %pinf2 %pip %popd %pprint %precision %prun %psearch %psource %pushd %pwd %pycat %pylab %qtconsole %quickref %recall %rehashx %reload_ext %rep %rerun %reset %reset_selective %rm %rmdir %run %save %sc %set_env %store %sx %system %tb %time %timeit %unalias %unload_ext %who %who_ls %whos %xdel %xmode
```

```
Available cell magics:
```

```
%%! %%HTML %%SVG %%bash %%capture %%debug %%file %%html %%javascript %%js %%latex %%markdown %%perl %%prun %%pypy %%python %%python2 %%python3 %%ruby %%script %%sh %%svg %%sx %%system %%time %%timeit %%writefile
```

```
Automagic is ON, % prefix IS NOT needed for line magics.
```