# EXPLORE - Create HTML report of Jupyter Notebook run

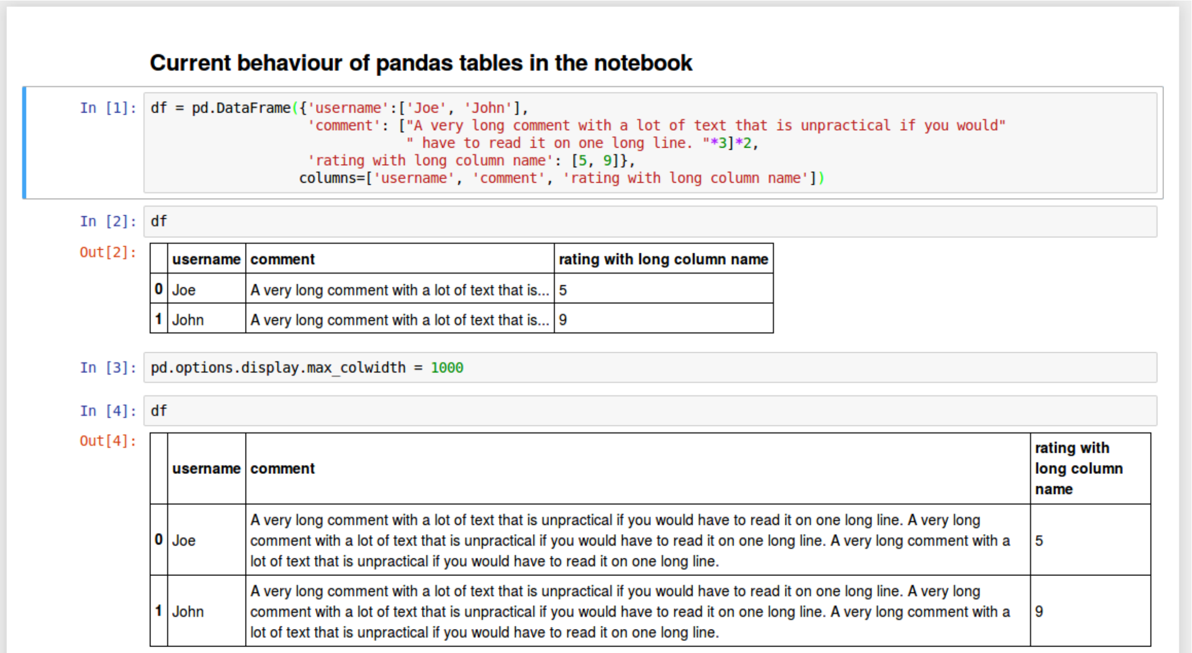
- https://stackoverflow.com/questions/40371187/execute-jupyter-notebook-and-keep-attached-to-existing-kernel

```
    jupyter nbconvert --execute --allow-errors --stdout test.ipynb >> result.html 2>&1

    jupyter nbconvert --debug --allow-errors --stdout --execute test.ipynb
```

# New Version of Jupyter 5.0 brings issues

Table Style changed



### Current behaviour of pandas tables in the notebook

```
In [1]: df = pd.DataFrame({'username':['Joe', 'John'],
                           'comment': ["A very long comment with a lot of text that is unpractical if you would"
                                       " have to read it on one long line. "*3]*2,
                           'rating with long column name': [5, 9]},
                          columns=['username', 'comment', 'rating with long column name'])

In [2]: df
```

Out[2]:

| | username | comment | rating with long column name |
|---|---|---|---|
| 0 | Joe | A very long comment with a lot of text that is... | 5 |
| 1 | John | A very long comment with a lot of text that is... | 9 |

```
In [3]: pd.options.display.max_colwidth = 1000

In [4]: df
```

Out[4]:

| | username | comment | rating with long column name |
|---|---|---|---|
| 0 | Joe | A very long comment with a lot of text that is unpractical if you would have to read it on one long line. A very long comment with a lot of text that is unpractical if you would have to read it on one long line. A very long comment with a lot of text that is unpractical if you would have to read it on one long line. | 5 |
| 1 | John | A very long comment with a lot of text that is unpractical if you would have to read it on one long line. A very long comment with a lot of text that is unpractical if you would have to read it on one long line. A very long comment with a lot of text that is unpractical if you would have to read it on one long line. | 9 |

# Markdown Syntax

### Writing Documentation in Jupyter

- Writing Documentation in Jupyter

### Adding Picture to Markdown

- Insert Image
- nest the image and link syntax

### Mastering Markdown

- Mastering Markdown

### iPython Cookbook

- Minibook for beginners
- Advanced Cookbook
- Featured Recipes advanced
- Example Recipe for Machine Learning
- Cookbook Code

---

# Indent or Comment block of code in Jupyter

Jupyter CodeMirror

- Indent less with CTRL+[
- Indent more with CTRL+]
- Comment block with CTRL+/

---

# Advanced Topic - Import Module from Jupyter Notebook

Importing Jupyter Notebooks as Modules

- Notebook Loader
- The Module Finder
- Register the hook
- Aside: displaying notebooks
- Notebooks in packages

# Advanced Topic - Batch Execution of Jupyter Notebooks

http://tritemio.github.io/smbits/2016/01/02/execute-notebooks/

---

# Cell Magic Commands

Magic Commands, Cell Magics in IPython, Great Cell Magic list

### Load file.py into notebook

- It will load file into existing cell and command original command line

```
%load C:\Users\bilsto1\python\unittest\unittest_1.py
```

### To write/save file

- write/save cell contents into myfile.py (use -a to append). Another alias: %%file myfile.py

```
%%writefile myfile.py
```

### To run Python file

- run myfile.py and output results in the current cell

```
%run myfile.py
```

### To load/import file

- load "import" myfile.py into the current cell

```
%load myfile.py
```

# For more magic and help

- list all the other cool cell magic commands.

```
%lsmagic
```

- for help on how to use a certain command. i.e. %run?

```
%COMMAND-NAME?
```

---

# Unix command

- Beside the cell magic commands, IPython notebook (now Jupyter notebook) is so cool that it allows you to use any unix command right from the cell (this is also equivalent to using the %%bash cell magic command).

To run a unix command from the cell, just precede your command with **!** mark.

- to see your python version

```
!python --version
```

- run myfile.py and output results in the current cell, just like %run (see the difference between !python and %run in the comments below).

```
!python myfile.py
```

# Execute cmd commands from IPython

Using Shell Commands

```
cmd "/c activate py3k && ipython --pylab"
```

In [1]:

```
%pwd
%cd C:\Users\bilsto1\Anaconda2\envs\py3Env
!python --version
!cd
```

```
C:\Users\bilsto1\Anaconda2\envs\py3Env
```

```
Python 3.6.1 :: Anaconda 4.4.0 (64-bit)
```

```
C:\Users\bilsto1\Anaconda2\envs\py3Env
```

In [33]:

```
# List Environment Variables
%env
```

In [7]:

```
# Show command help
%load?
```

In [30]:

```
# Show list of all magic cell commands
%lsmagic
```

In [31]:

```
# List command history
%history
```

In [32]:

```
# show directory history
%dhist
```

In [37]:

```
# Exports history of notebook
#%notebook Jupyter_tips.ipynb
```

In [37]:

```
#This function provides very basic timing functionality.
#Use the timeit magic for more control over the measurement.
%time 2*3
```

In [39]:

```
# Lists all namespaces
#Print all interactive variables, with some minimal formatting.
%who
%who_ls
%whos

a = 2
b = 'biljana'
```

In [24]:

```
# Delete interactive variable
%xdel a
```

In [34]:

```
# Inline
%%html
%%javascript
%%js
```

# Start Jupyter Notebook Server

Run Anaconda Prompt, or open cmd prompt then go to working folder, and type command:

```
        Jupyter Notebook
```

It will start jupyter notebook at http://localhost:8888/tree

- Jupyter notebook server starts at default port 8888.

***KERNEL - provides programming language support in Jupyter, such as Python, Julia, R, rtc.***

- When Notebook in open, Kernel (computational engine) is started.
- Closing Browser will now close the server running notebook

***KERNEL(s) Running***

```
> jupyter notebook list

It will show all kernels and their tokens
```

- Go to localhost:8888/tree, and then enter token for this address/server

# Jupyter Kernels

To run Notebook in different languages, you need to install particular kernel:

- IPyKernel
- IJavaScript
- IRKernel
- IOctave
- IMatLab

# View Kernel Version from Jupyter, List all available Kernels

```
import sys
sys.version
sys.version_info
```

In [1]:

```
import sys
sys.version_info
```

Out[1]:

```
sys.version_info(major=3, minor=6, micro=1, releaselevel='final', serial=0)
```

In [3]:

```
!jupyter-kernelspec list
```

```
Available kernels:
  python2    C:\Users\bilsto1\Anaconda2\share\jupyter\kernels\python2
  py3env     C:\ProgramData\jupyter\kernels\py3env
  testenv    C:\ProgramData\jupyter\kernels\testenv
```

In [7]:

```
!jupyter --paths
```

```
config:
    C:\Users\bilsto1\.jupyter
    C:\Users\bilsto1\Anaconda2\etc\jupyter
    C:\ProgramData\jupyter
data:
    C:\Users\bilsto1\AppData\Roaming\jupyter
    C:\Users\bilsto1\Anaconda2\share\jupyter
    C:\ProgramData\jupyter
runtime:
    C:\Users\bilsto1\AppData\Roaming\jupyter\runtime
```

# Autocompletion in Jupyter

enter TAB

- If any issues, install **pyreadline**
- Open jupyter notebook & run the following in one the cell:

```
    %config IPCompleter.greedy=True
```

# IJavaScript

https://github.com/n-riesco/ijavascript#installation

# Run unittest in Jupyter

Modify call to main() to add arguments explicitely, since no cmd line used

```
    argv=['first-arg-is-ignored'], exit=False
```

- The reason is that unittest.main looks at sys.argv and first parameter is what started IPython or Jupyter, therefore the error about kernel connection file not being a valid attribute.
- Passing explicit list to unittest.main will prevent IPython and Jupyter look at sys.argv.
- Passing exit=False will prevent unittest.main to shutdown the kernell process

In [2]:

```
import unittest

class TestStringMethods(unittest.TestCase):

    def test_upper(self):
        self.assertEqual('foo'.upper(), 'FOO')

    def test_isupper(self):
        self.assertTrue('FOO'.isupper())
        self.assertFalse('Foo'.isupper())

    def test_split(self):
        s = 'hello world'
        self.assertEqual(s.split(), ['hello', 'world'])
        # check that s.split fails when the separator is not a string
        with self.assertRaises(TypeError):
            s.split(2)

if __name__ == '__main__':
    unittest.main(argv=['first-arg-is-ignored'], exit=False)
```

```
...
----------------------------------------------------------------------
Ran 3 tests in 0.047s

OK
```

In [9]:

```
# this command doesn't work in windows
#!WHERE firefox.exe

# Run executable from Python
import os
os.system( '"C:\\Program Files (x86)\\Mozilla Firefox\\firefox.exe"' )
```

Out[9]:

```
1
```

# Jupyter Configuration

***See all options***

```
jupyter notebook --help
```

***Generate config file***

```
jupyter notebook --generate-config
```

***Set logging level***

```
jupyter notebook --log-level=0

0 = DEBUG
10 = INFO
20 = WARN
```

```
30 = ERROR
40 =
50 = CRITICAL
```

***Change default browser when opening jupyter from command line***

```
jupyter notebook --browser=u'"C:\\Program Files (x86)\\Mozilla Firefo
x\\firefox.exe"' | clip
```

***Change default browser in config file***

- https://stackoverflow.com/questions/35229604/how-to-change-the-default-browser-used-by-the-ipython-jupyter-notebook-in-linux

# Jupyter Security

- Why are tokens or passwords necessary

# Show line number in Jupyter Notebook

StackOverflow Show Jupyter Line Number for all cells

- Command will show jupyter configuration directory

    ```
    jupyter --config-dir
    ```

configuration file in json:

```
{
  "Cell": {
    "cm_config": {
      "lineNumbers": true
    }
  },
  "Notebook": {
    "Header": true,
    "Toolbar": true
  }
}
```

# Output hyperlink in the Notebook using Python code

In [19]:

```
from IPython.core.display import display, HTML
display(HTML("""<a href="https://google.com">Go to Google.com</a>"""))
```

Go to Google.com

## Jupyter NodeJS

https://github.com/notablemind/jupyter-nodejs

## IJavaScript

https://www.npmjs.com/package/ijavascript

## nbmanager - to view all kernels/servers running and shutdown

https://github.com/takluyver/nbmanager

## Advanced Jupyter Notebook Tricks - 1 & 2 & Trump

- https://blog.dominodatalab.com/lesser-known-ways-of-using-notebooks/
- https://blog.dominodatalab.com/interactive-dashboards-in-jupyter/
- https://blog.dominodatalab.com/data-science-just-one-way-donald-trump-different/