

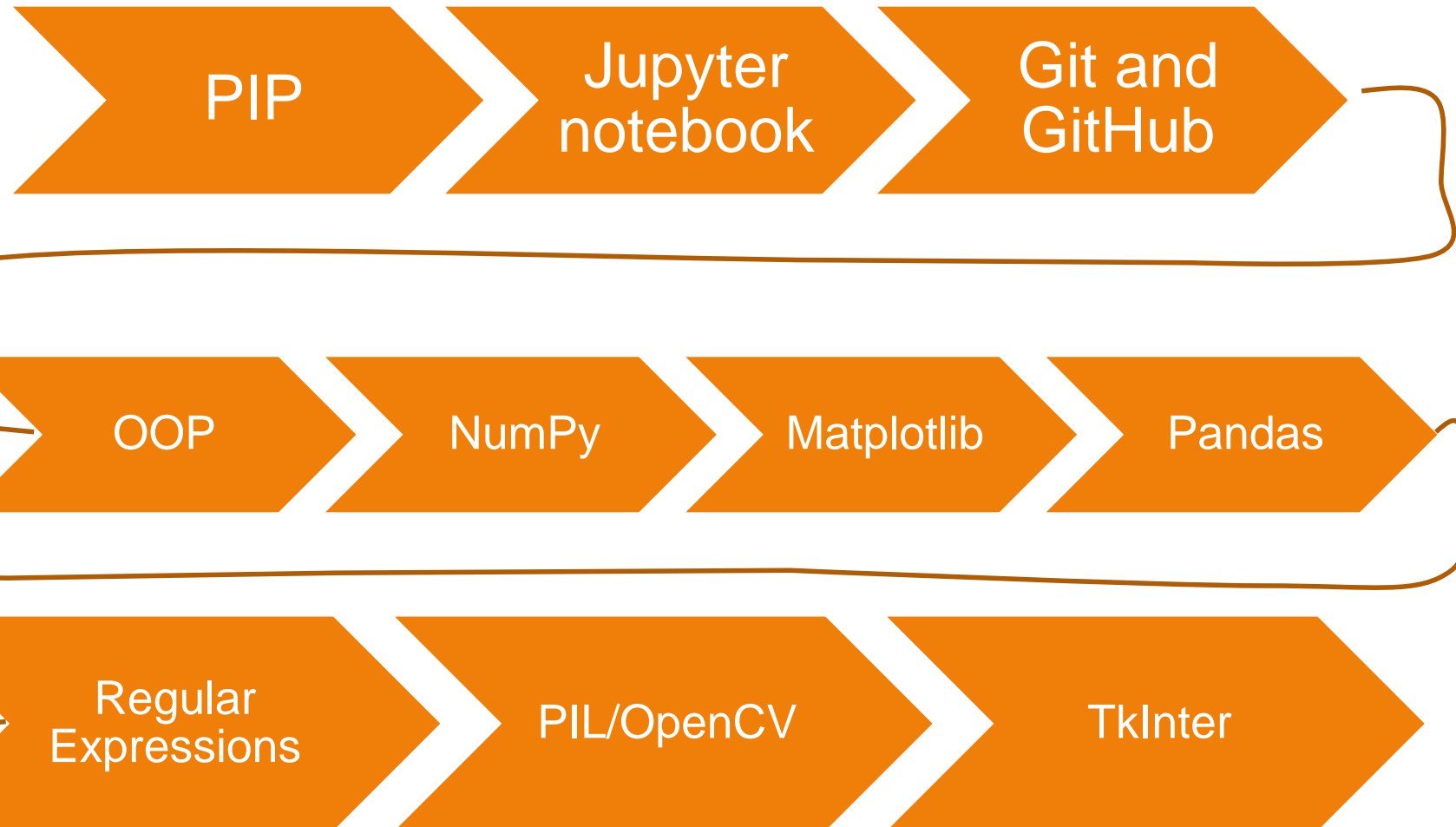
Advanced Python Programming Course

Lecture 1.

**Pip. Installation and using Jupyter
Notebook.**

Assoc. Prof. Kovalenko S.M.
Department of software engineering and
intelligent management technologies,
NTU “KhPI”

Roadmap



PIP

- PIP stands for a recursive acronym "Pip Installs Packages" .
- PIP (<https://pip.pypa.io/en/stable/>) is a package **manager** for Python.
- You can use it to install packages from the Python Package Index (<https://pypi.org/>) and other indexes
- The latest versions of Python (3.4 and higher) come with PIP pre-installed

PIP commonly used commands

- pip version

```
pip --version
```

- help

```
pip help <command>
```

- list of packages along with the installed version

```
pip list
```

- output the installed packages in the requirements format

```
pip freeze
```

- create a requirements.txt file with the list of installed packages and their corresponding versions

```
pip freeze > requirements.txt
```

PIP commonly used commands

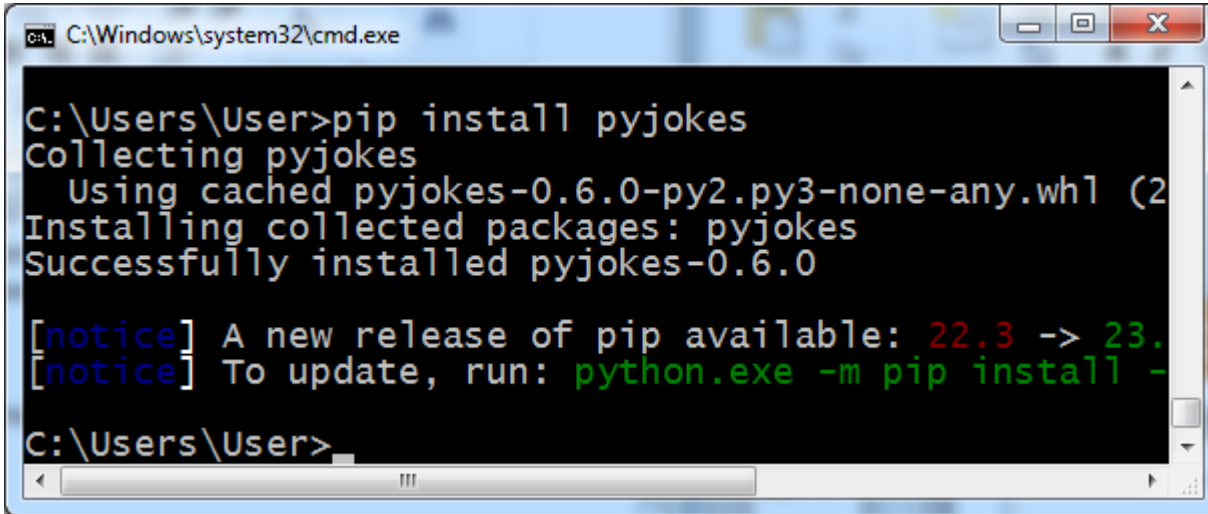
- install the dependencies using the requirements file
`pip install -r <name of the requirement file>`
- install a package
`pip install <package>`
- install a package with a specific version
`pip install <package-name==version-number>`
- uninstall package
`pip uninstall <package>`

NB

- If you are working with multiple versions of Python, using the `pip install` command will install the package to the version of Python from which you run the command.
- To avoid this issue, you can use the command
`python -m pip install <module>.`
- This command will install the package for the version of Python that you are currently using, regardless of which version of Python is the default

Example

- `pip install pyjokes`



```
C:\Windows\system32\cmd.exe

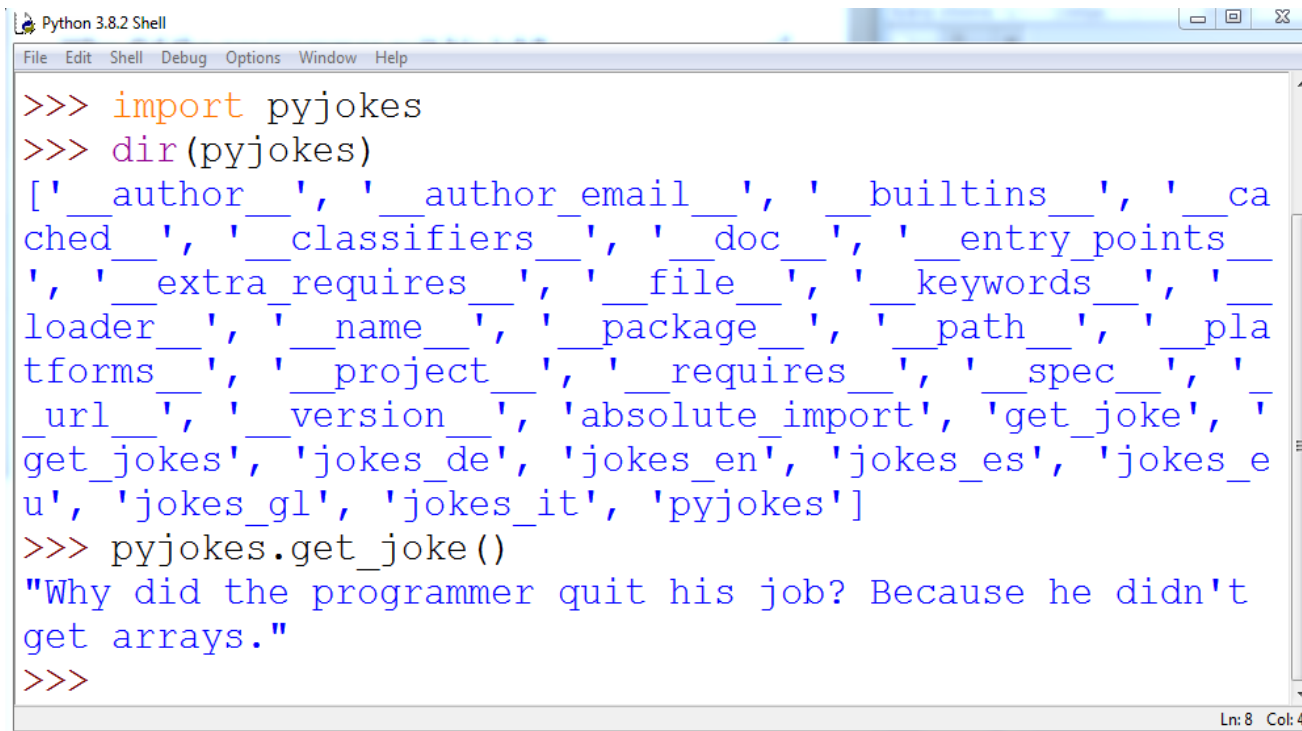
C:\Users\User>pip install pyjokes
Collecting pyjokes
  Using cached pyjokes-0.6.0-py2.py3-none-any.whl (2
Installing collected packages: pyjokes
Successfully installed pyjokes-0.6.0

[notice] A new release of pip available: 22.3 -> 23.
[notice] To update, run: python.exe -m pip install -
C:\Users\User>
```

Example

```
>>>import pyjokes  
>>>dir(pyjokes)  
>>> pyjokes.get_joke()
```

<http://www.jokeexplainer.com/2017/04/why-did-programmer-quit-his-job.html>



```
Python 3.8.2 Shell  
File Edit Shell Debug Options Window Help  
>>> import pyjokes  
>>> dir(pyjokes)  
['_author_', '_author_email_', '_builtins_', '_cached_', '_classifiers_', '_doc_', '_entry_points_', '_extra_requires_', '_file_', '_keywords_', '_loader_', '_name_', '_package_', '_path_', '_platforms_', '_project_', '_requires_', '_spec_', '_url_', '_version_', 'absolute_import', 'get_joke', 'get_jokes', 'jokes_de', 'jokes_en', 'jokes_es', 'jokes_eu', 'jokes_gl', 'jokes_it', 'pyjokes']  
>>> pyjokes.get_joke()  
"Why did the programmer quit his job? Because he didn't get arrays."  
>>>
```

Ln: 8 Col: 4

About Jupyter Notebook

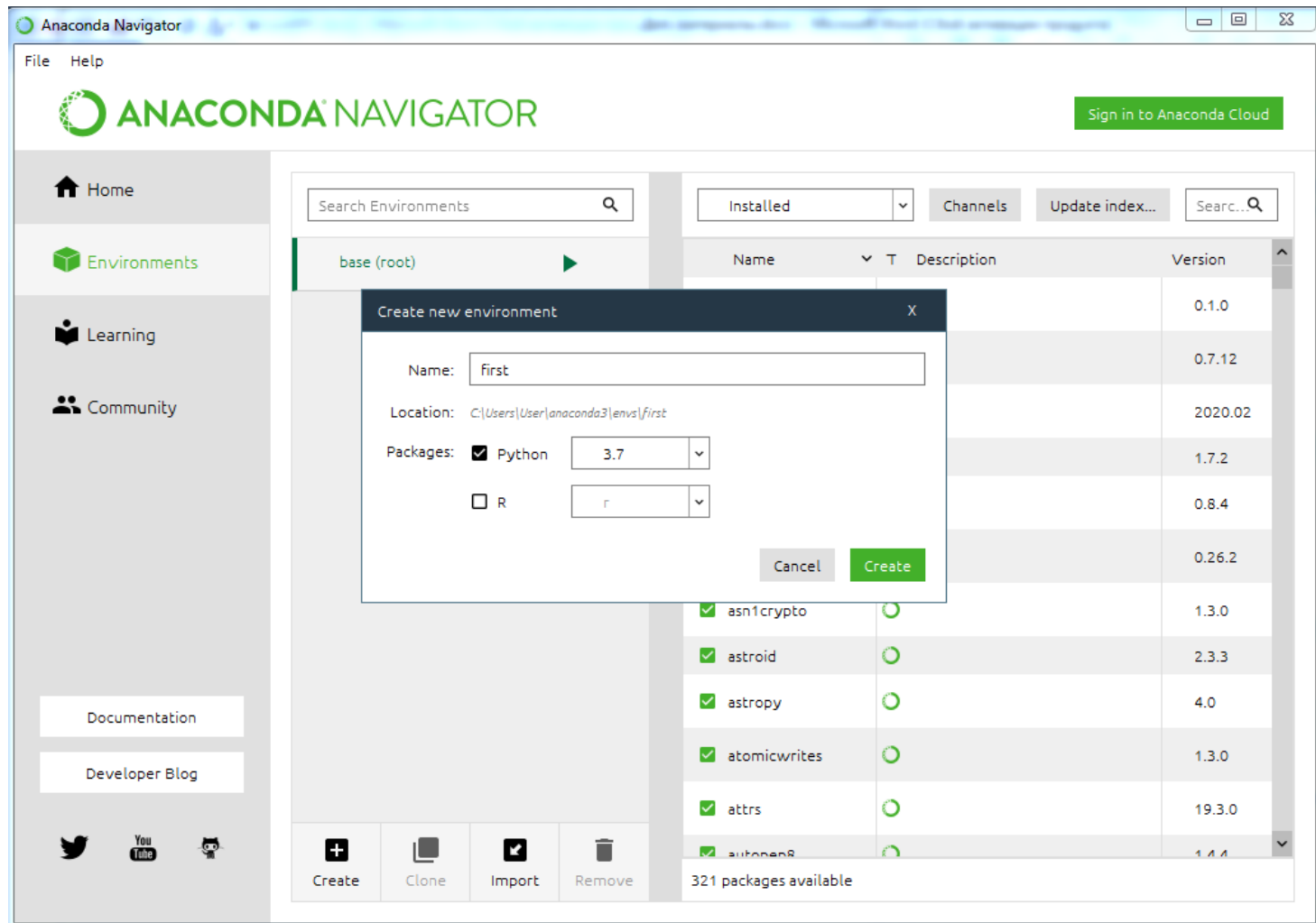
- The **Jupyter Notebook** is an incredibly powerful tool for interactively developing and presenting data science projects. This lecture will walk you through how to use Jupyter Notebooks for data science projects and how to set it up on your local machine.
- "Jupyter" is a loose acronym meaning Julia, Python and R, but today, the notebook technology supports many programming languages.

Installing Jupyter Notebook

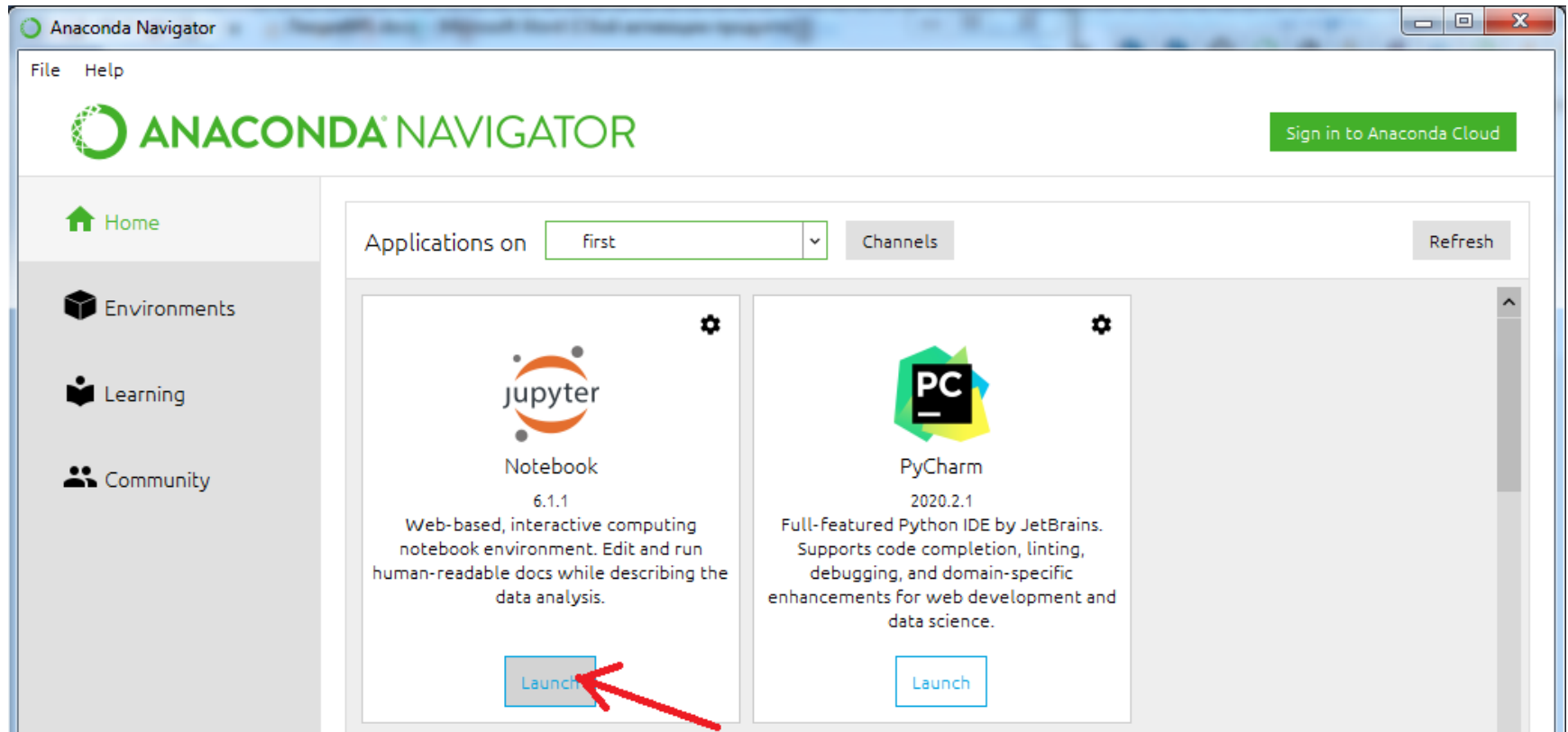
- The easiest way for a beginner to get started with Jupyter Notebooks is by installing Anaconda
(<https://www.anaconda.com/products/individual>)



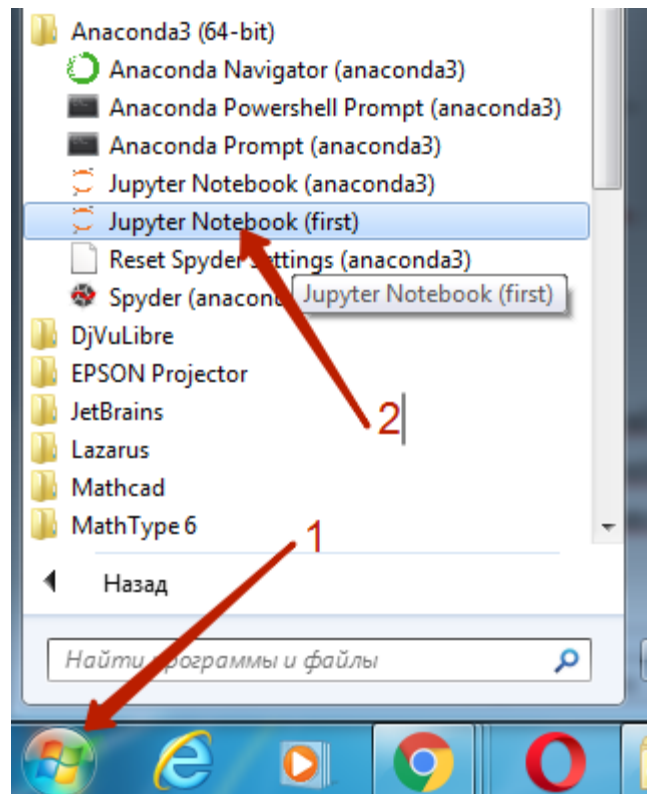
Anaconda Navigator



Install and run the Jupyter Notebook



Launch Notepad from the Start menu



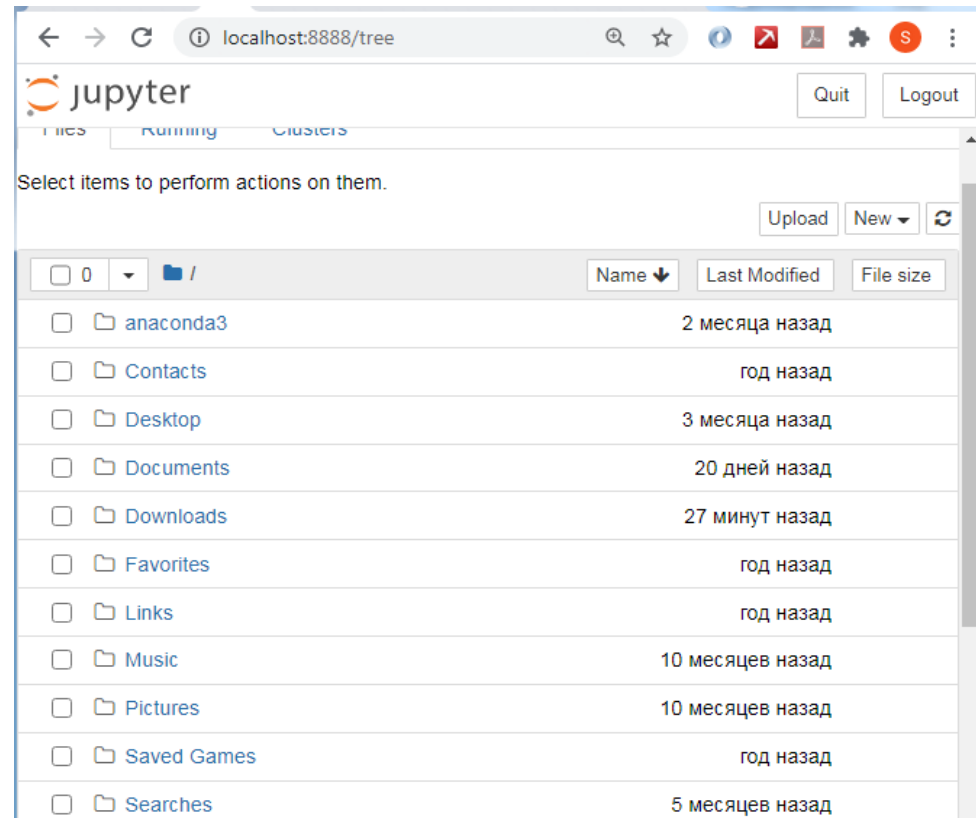
Installing using pip

- Install Jupyter Notebook

```
pip install jupyter
```

- Launch notebook

```
jupyter notebook
```



Change start-up directory

```
jupyter notebook --notebook-dir  
"full_path_to_the_folder"
```

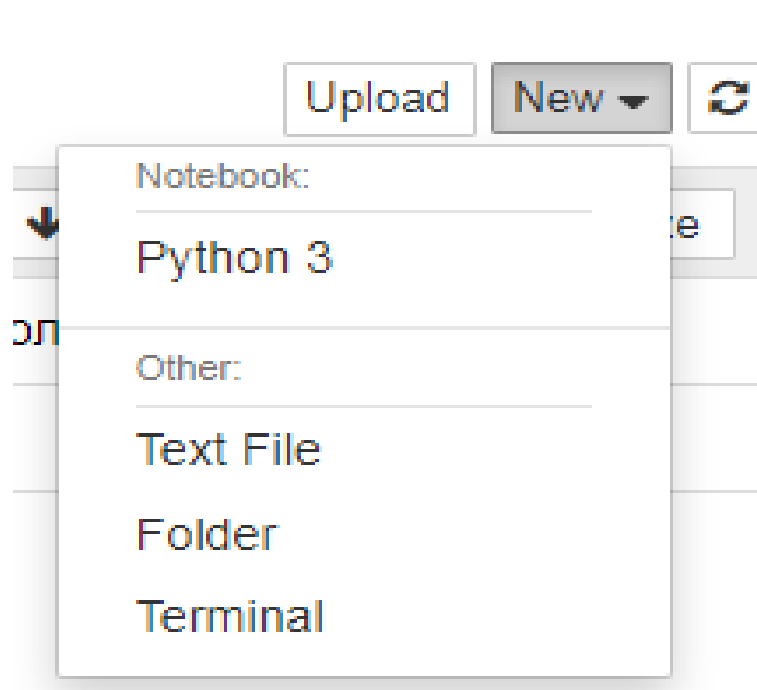
for example

```
jupyter notebook --notebook-dir "D:\py"
```

Other ways you can view at the link

<https://stackoverflow.com/questions/35254852/how-to-change-the-jupyter-start-up-folder>

Create the first notebook



Rename notebook

- You can change the file name by double-clicking it



The image shows a 'Rename Notebook' dialog box. It has a title bar with the text 'Rename Notebook' and a close button (X) in the top right corner. Below the title bar, there is a label 'Enter a new notebook name:'. Underneath the label is a text input field containing the text 'My_first_notebook'. At the bottom right of the dialog box, there are two buttons: 'Cancel' and 'Rename'.

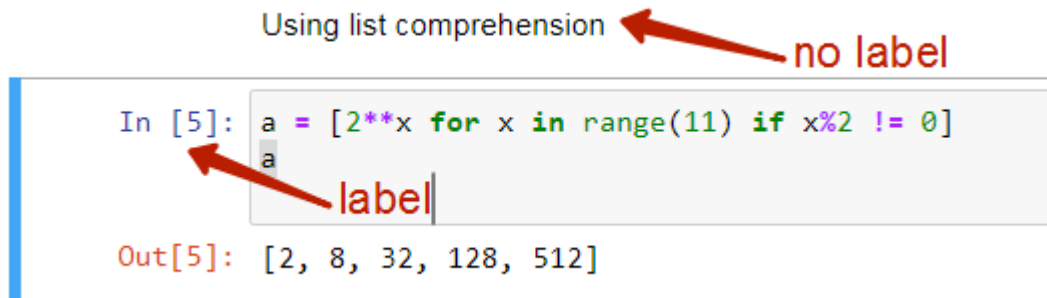
Kernels and cells

- A kernel is a "computational engine" that executes the code contained in a notebook document.
- A cell is a container for text to be displayed in the notebook or code to be executed by the notebook's kernel.

Cells

Cells form the body of a notebook. In the screenshot of a new notebook in the section above, that box with the green outline is an empty cell. There are two main cell types that we will cover:

- A **code cell** contains code to be executed in the kernel and displays its output below.
- A **Markdown cell** contains text formatted using Markdown and displays its output in-place when it is run.



The screenshot shows a Jupyter Notebook code cell. The code is: `In [5]: a = [2**x for x in range(11) if x%2 != 0]`. Below the code is the output: `Out[5]: [2, 8, 32, 128, 512]`. There are two red arrows pointing to the code. One arrow points to the text "Using list comprehension" above the code, and the other points to the variable "a" in the code. The text "no label" is written next to the first arrow, and the text "label" is written next to the second arrow.

```
Using list comprehension ← no label
In [5]: a = [2**x for x in range(11) if x%2 != 0]
        a
        label
Out[5]: [2, 8, 32, 128, 512]
```

Notebook modes

Jupyter Notebooks have two modes:

- edit
- command.

Edit Mode

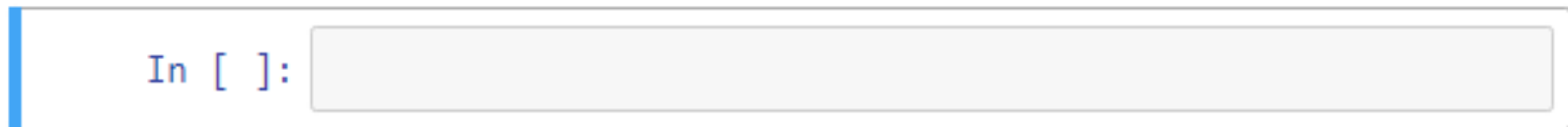
- To enter edit mode, press Enter on your keyboard or click a cell.
- The edit mode can be defined by the **green frame** around the cell with the green left field.
- When you are in edit mode, you can enter text in the cells.



In []:

Command Mode

- To enter command mode, press Esc or click anywhere outside the cell.
- You will see a **gray border** around the cell with a **blue left edge**.
- When you are in command mode, you can edit your notebook, but you cannot enter cells.



Most useful shortcuts

- Run the selected cell: `Ctrl + Enter`
- Run the cell and insert a new cell below: `Alt + Enter`
- Run the cell and go to the cell below: `Shift + Enter`
- Insert a cell above: `Esc + A`
- Insert a cell below: `Esc + B`
- Cut the selected cells: `Esc + X`
- Copy selected cells: `Esc + C`

Google Colaboratory

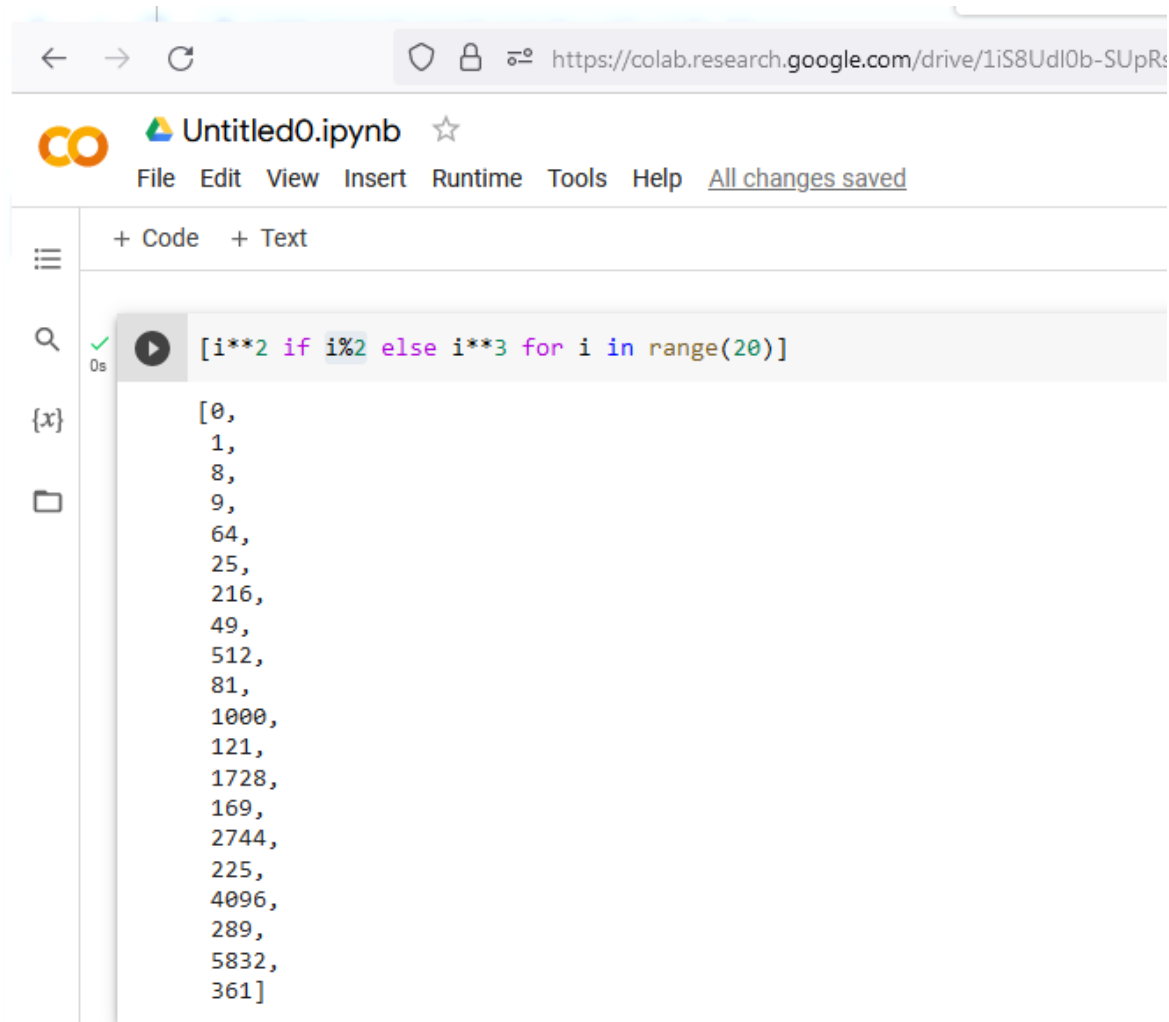
<https://colab.research.google.com/?hl=en>

Colab, or "Colaboratory", allows you to write and execute Python in your browser, with

- Zero configuration required
- Access to GPUs free of charge
- Easy sharing

But you need a Google account

Google Colaboratory



The screenshot displays the Google Colaboratory web interface. The browser's address bar shows the URL `https://colab.research.google.com/drive/1iS8Udl0b-SUpRs`. The notebook is titled "Untitled0.ipynb" and has a star icon. The menu bar includes "File", "Edit", "View", "Insert", "Runtime", "Tools", "Help", and a link "All changes saved". The left sidebar contains icons for a menu, search, variables (showing an empty set `{x}`), and a file explorer. The main code area shows a list comprehension being executed, with a play button icon and a "0s" timer. The code is `[i**2 if i%2 else i**3 for i in range(20)]`. The output is a list of 20 numbers: `[0, 1, 8, 9, 64, 25, 216, 49, 512, 81, 1000, 121, 1728, 169, 2744, 225, 4096, 289, 5832, 361]`.

```
[i**2 if i%2 else i**3 for i in range(20)]
```

```
[0,  
 1,  
 8,  
 9,  
 64,  
 25,  
 216,  
 49,  
 512,  
 81,  
 1000,  
 121,  
 1728,  
 169,  
 2744,  
 225,  
 4096,  
 289,  
 5832,  
 361]
```