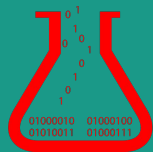
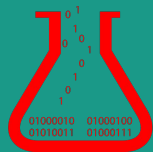

Intro and Setup

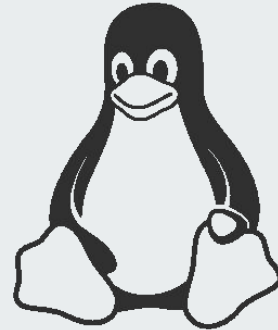
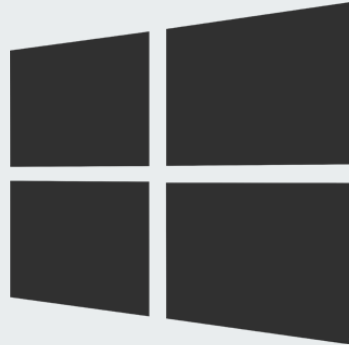


Intro and Setup

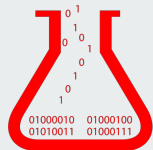


Required Materials

- Laptop running Windows, OSX, or Linux
 - Internet Connection



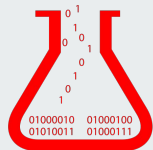
Required Software



What is Anaconda?



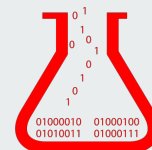
Why do we want to use it?



What is Anaconda?

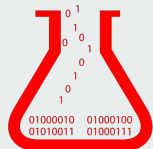
Anaconda is a package manager for Python

- Provides: easily installed, curated packages that do not conflict.
Installs most important python software with a push of a button



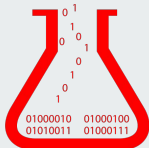
What is Spyder?

Spyder - included in the full install, is an open source (free) IDE for Python



Jupyter Notebooks

Jupyter notebooks provide an easy way to run and see the output of a Python program



Getting Setup



Download from
<https://www.anaconda.com/downloads/>
(make sure to download the 3.7 version)

The graphical installer is easiest and can be used for mac and PC

Python 3.7 version *

 Download

[64-Bit Graphical Installer \(614.3 MB\)](#) ?

[32-Bit Graphical Installer \(509.7 MB\)](#)

Python 2.7 version *

 Download

[64-Bit Graphical Installer \(560.6 MB\)](#) ?

[32-Bit Graphical Installer \(458.6 MB\)](#)



Getting Setup

Find where you downloaded the Anaconda installer and start the installation process

- The graphical installer is likely on the downloads folder, just double click it.
- Allow Anaconda to prepend paths (select YES on setting the PATH during setup)
 - This is likely an **unselected** checkbox (that is you have to check it!)

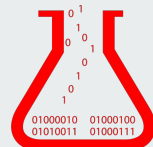


Running Python from the Command Line

- Find and open your terminal, this will be powershell on PC and shell/terminal on mac
- Launch python from the shell, by typing 'python'
- Should look like this:

```
(py3) D:\>python
Python 3.6.7 |Anaconda, Inc.| (default, Dec 10 2018, 20:35:02) [MSC v.1915 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- Make sure the version is Python 3.6 or above



Python - simple test commands

- Type in `100 + 100` and hit enter - should get a response of 200

```
(py3) D:\>python
Python 3.6.7 |Anaconda, Inc.| (default, Dec 10 2018, 20:35:02) [MSC v.1915 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 100 + 100
200
>>>
```

- Next type in a simple print command in Python:
 - `print("Hello World")`

```
(py3) D:\>python
Python 3.6.7 |Anaconda, Inc.| (default, Dec 10 2018, 20:35:02) [MSC v.1915 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 100 + 100
200
>>> print("Hello World")
Hello World
>>>
```

Python - simple test commands

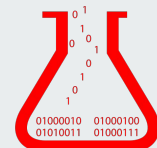
- This interpreter is useful for short commands or testing of some code but probably isn't practical for a large program
 - To exit: type `exit()` or CTRL-D

```
(py3) D:\>python
Python 3.6.7 |Anaconda, Inc.| (default, Dec 10 2018, 20:35:02) [MSC v.1915 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 100 + 100
200
>>> print("Hello World")
Hello World
>>> exit()

(py3) D:\>
```

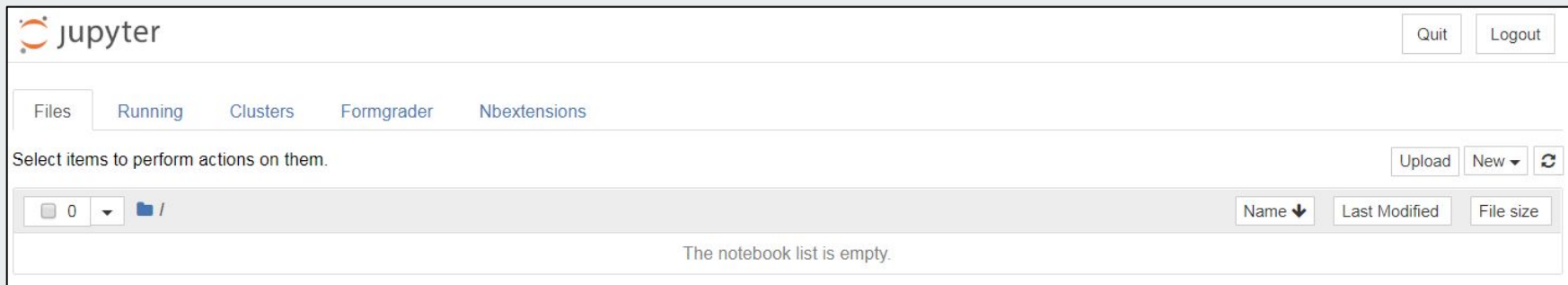


Python - Jupyter Notebook



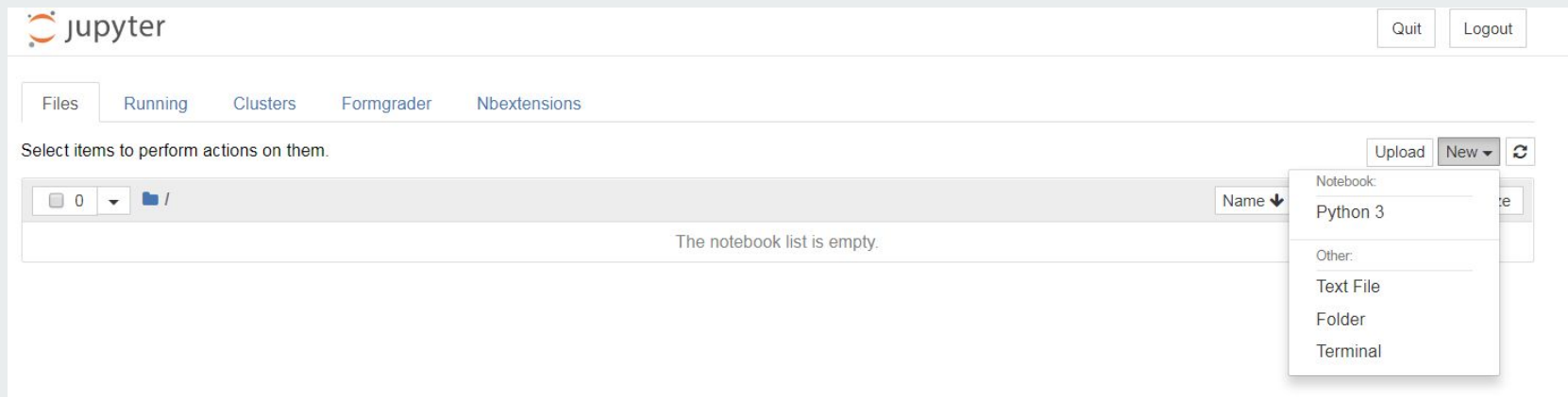
Opening Jupyter Notebook

- Type in 'jupyter notebook' on your command line
- The notebook server should open in your browser like this:



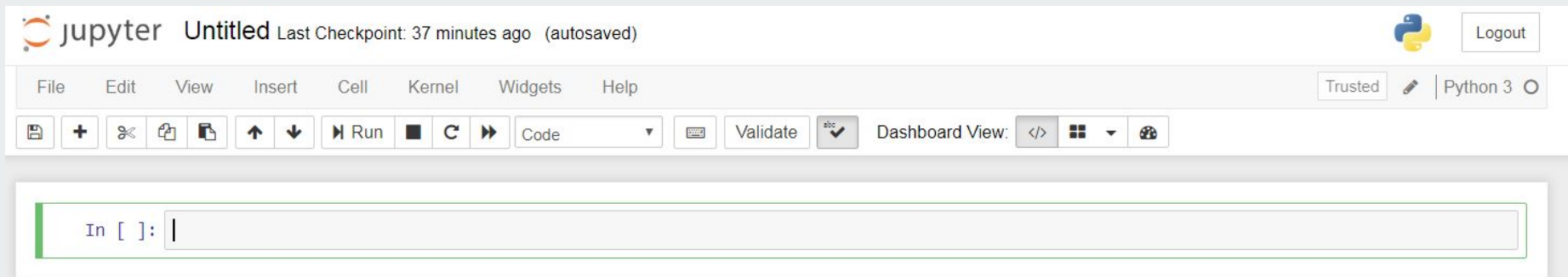
Opening Jupyter Notebook

- In the upper right click on New - and then Python 3 from the dropdown menu



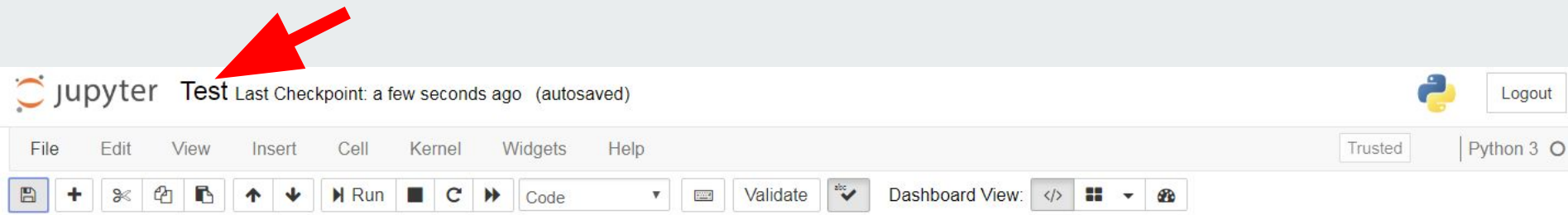
Opening Jupyter Notebook

- That should bring up a new tab in the browser that looks like this:



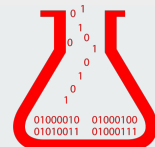
Coding in a Jupyter Notebook

- Change the title: test



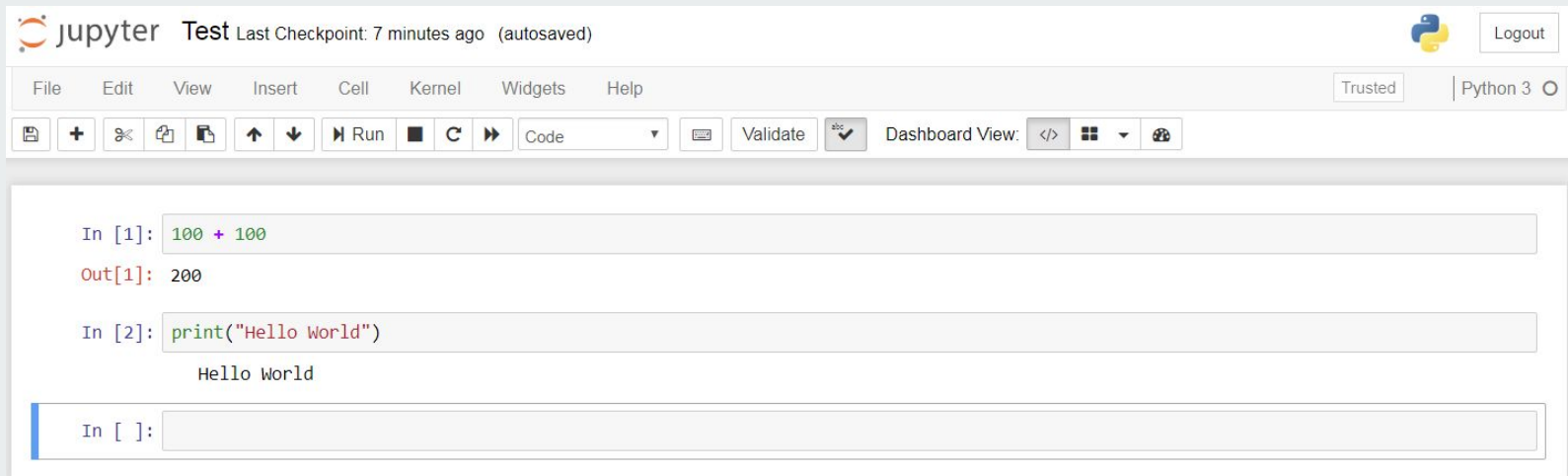
In []:

Coding Cell



Coding in a Jupyter Notebook

- In the coding cell type in `100 + 100`
- Click the 'Run' button (or SHIFT-ENTER) to run the code cell

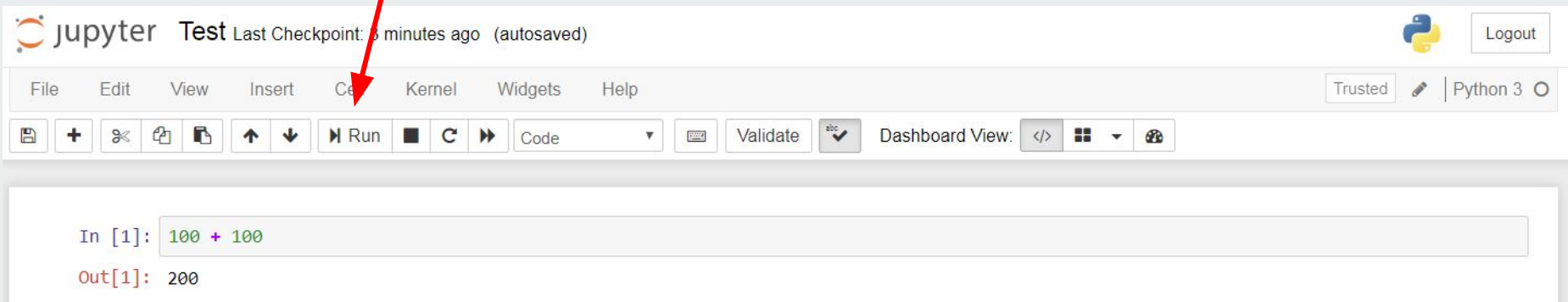


The screenshot displays the Jupyter Notebook interface. At the top, the header shows the Jupyter logo, the name 'Test', and the status 'Last Checkpoint: 7 minutes ago (autosaved)'. On the right, there is a Python logo and a 'Logout' button. Below the header is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. To the right of the menu bar are 'Trusted' and 'Python 3' indicators. A toolbar below the menu bar contains icons for saving, adding cells, undo, redo, and other functions, along with a 'Run' button (a play icon) and a 'Validate' button. The main area shows two code cells. The first cell contains the code `In [1]: 100 + 100` and its output `Out[1]: 200`. The second cell contains the code `In [2]: print("Hello World")` and its output `Hello World`. A third, empty code cell is at the bottom, labeled `In []:`.



Coding in a Jupyter Notebook

- In the next coding cell type in `print("Hello World")`
- Again Click the 'Run' button (or SHIFT-ENTER) to run the code cell



The screenshot displays the Jupyter Notebook interface. At the top, the header shows 'jupyter Test' with a status 'Last Checkpoint: 5 minutes ago (autosaved)' and a 'Logout' button. Below the header is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. A red arrow points to the 'Run' button in the 'Cell' menu. The 'Run' button is represented by a play icon. To the right of the 'Run' button are buttons for 'Code', 'Validate', and 'Dashboard View'. Below the menu bar is a code cell containing the text 'In [1]: 100 + 100'. Below the code cell is the output 'Out[1]: 200'.

jupyter Test Last Checkpoint: 5 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code Validate Dashboard View: </>

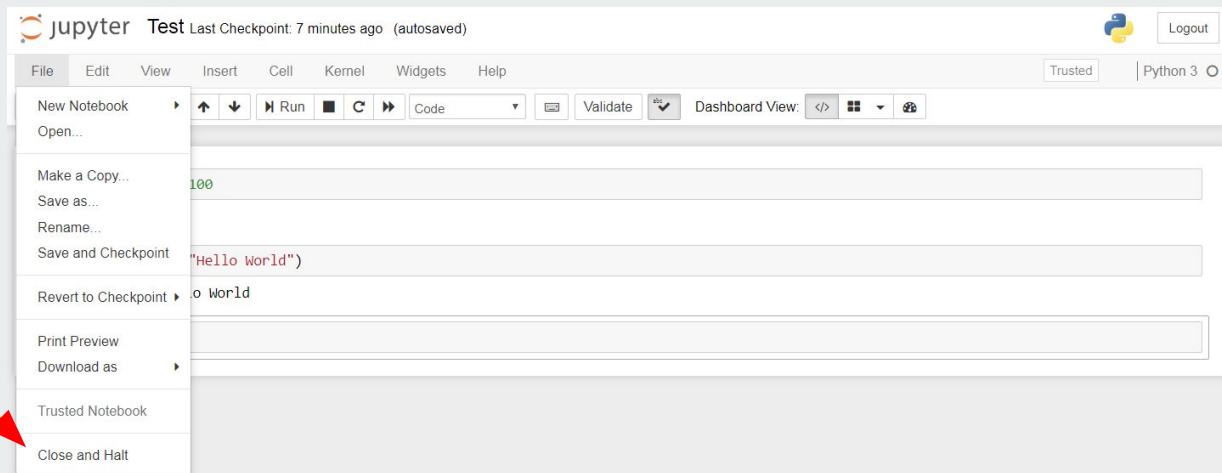
In [1]: 100 + 100

Out[1]: 200



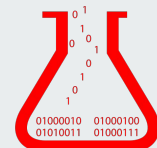
Exiting Jupyter Notebook

- To exit jupyter notebook - click on the File menu; select Close and Halt
- It is important to exit this way, if you just 'X' out of the tab - the notebook **will still be running** in the background (and this can chew up system resources!)



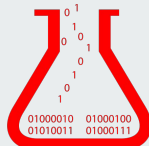
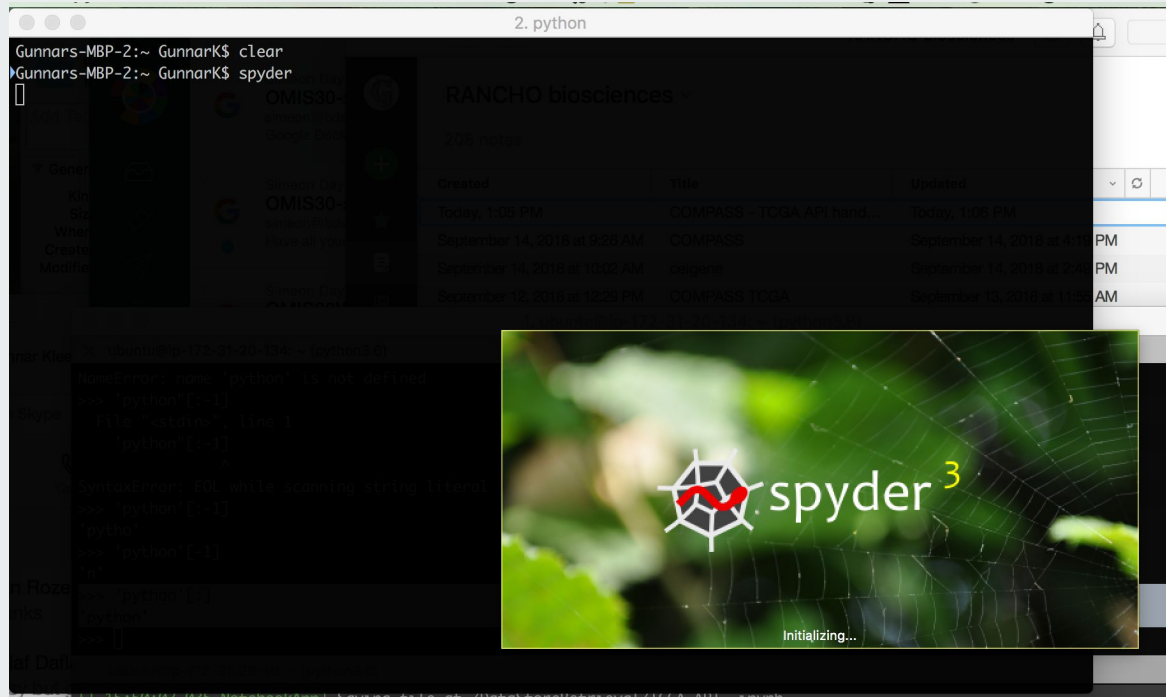
Using Spyder

- What is an IDE and how do we use it?
- When do we use an IDE vs a notebook?



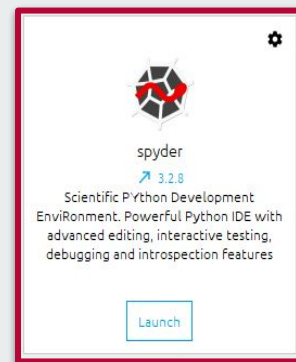
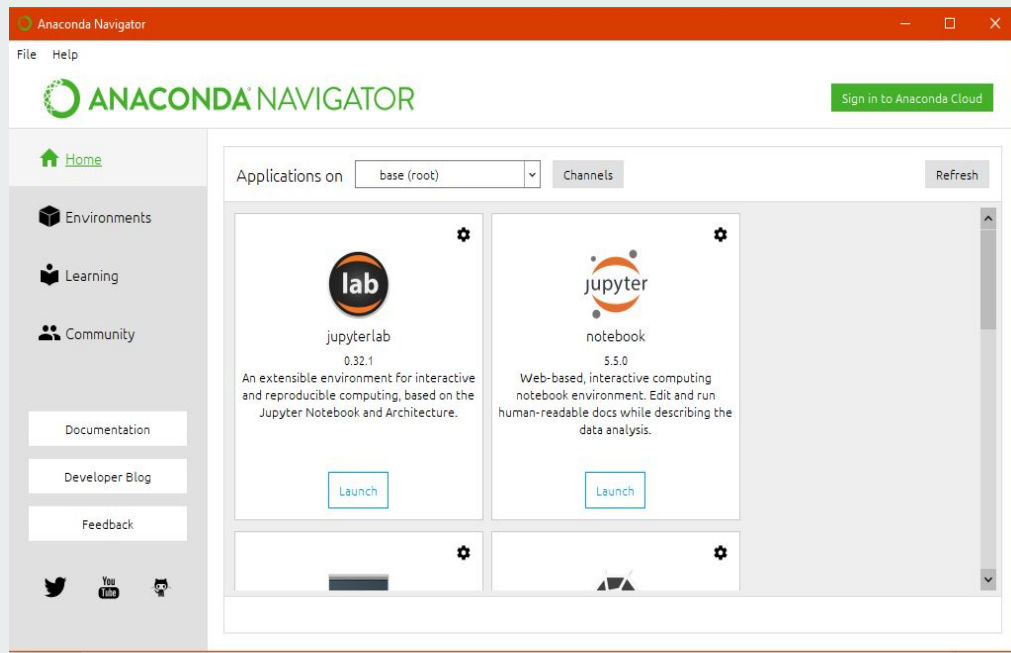
Opening Spyder

- Command Line/Terminal - type 'spyder' from the command line

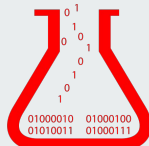
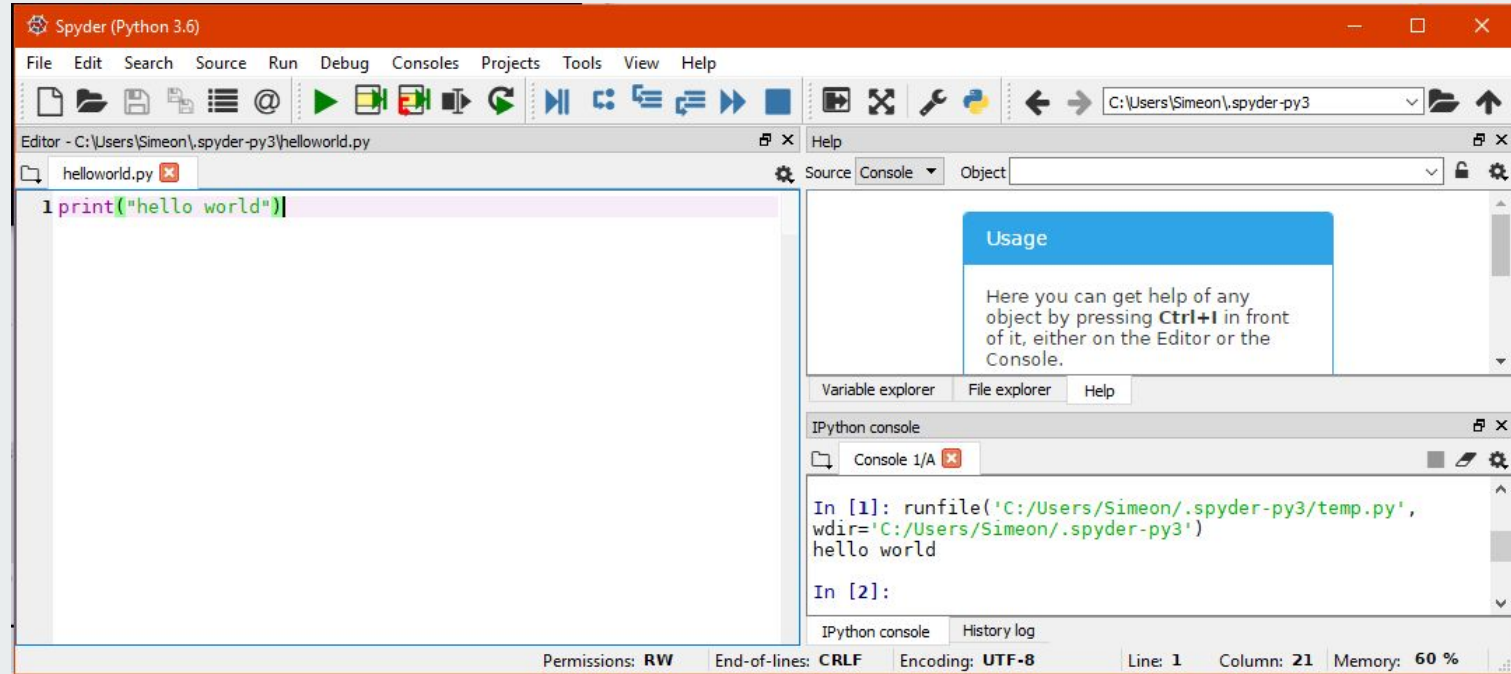


Opening Spyder

- Anaconda Navigator



Using Spyder




Exiting Spyder

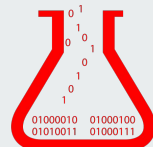


- If you're running spyder from your terminal or CMD, you can close the program by typing control+c
- If opened in Anaconda Navigator close by exiting the window



Adding a new Python library with Anaconda

- 
- Using Anaconda Navigator - open the Navigator
 - Click on Environments on the left bar
 - Make sure the dropdown bar in the middle is not installed (or All)
 - Type in the library you want to install
 - Click the box on the library to install
 - Click Apply in the button right corner



Adding a new Python library with Anaconda

The screenshot shows the Anaconda Navigator web interface. On the left sidebar, the 'Environments' tab is selected, indicated by a red arrow and the number 1. The main panel displays a list of environments: 'base (root)', 'Tableau-Python-Server', and 'altair'. The 'altair' environment is selected, indicated by a red arrow and the number 4. Above the environment list, there is a search bar and a dropdown menu showing 'Not installed', with a red arrow and the number 2 pointing to it. To the right of the dropdown is a search bar containing 'pandas', with a red arrow and the number 3 pointing to it. Below the search bar is a table of packages available for the selected environment. The table has columns for 'Name', 'Description', and 'Version'. The 'pandas' package is highlighted, indicated by a red arrow and the number 4. At the bottom right of the interface, there is an 'Apply' button, with a red arrow and the number 5 pointing to it.

ANACONDA NAVIGATOR

Sign in to Anaconda Cloud

Home

Environments

Learning

Community

Documentation

Developer Blog

Create Clone Import Remove

Search Environments

base (root)

Tableau-Python-Server

Tableau-Python-Server

altair

Not installed Channels Update index... pandas X

Name	Description	Version
<input type="checkbox"/> autovizwidget	An auto-visualization library for pandas dataframes	0.12.6
<input type="checkbox"/> blaze	Numpy and pandas interface to big data	0.9.1
<input type="checkbox"/> geopandas	Geographic pandas extensions.	0.4.0
<input checked="" type="checkbox"/> pandas	High-performance, easy-to-use data structures and data analysis tools.	0.9.1
<input type="checkbox"/> pandas-datatreader	Up to date remote data access for pandas, works for multiple versions of pandas	0.7.0
<input type="checkbox"/> pandas-profiling	Generate profile report for pandas dataframe	1.4.1
<input type="checkbox"/> pandasql	Sql/d for pandas	0.7.3
<input type="checkbox"/> qgrid	Pandas dataframe viewer for jupyter notebook	1.1.1
<input type="checkbox"/> streamz	Manage streaming data, optionally with dask and pandas	0.5.0

9 packages available matching "pandas" 1 package selected

Apply Clear

Adding a new Python library with Anaconda

- Using Anaconda Prompt
- Type conda install <library name>
 - For example: conda install pandas

```
cmd C:\Windows\system32\cmd.exe - conda install pandas
```

```
(base) D:\>conda install pandas  
Solving environment: |
```

- After it finishes 'Solving environment' - hit 'y' to proceed



Questions?



Contact:

Denis Vrdoljak
denis@bds.group

Practice Problems

- Create a “Hello World” program in a Jupyter Notebook
- Create a “Hello World” program as a standalone .py script, and run it from the command line
- Modify the Jupyter Notebook version to ask the user to input their name, then say hello to them.

