# 00 - Introduction to Jupyter Notebooks

February 21, 2023

## 1 Practical Sessions

Vincenzo Scotti

vincenzo.scotti@polimi.it

## 2 A (very) brief introduction to jupyter notebooks

---

You can find many detailed introductions online: e.g. https://realpython.com/jupyter-notebook-introduction/. This introduction is partially taken from there.

Official documentation of Jupyter Notebook: https://jupyter-notebook.readthedocs.io/en/stable/

---

### 2.1 Pre-requisites

It requires that you have `python` installed; in this course we will be using **`python3`**. If you don't have it, please install it; one of the most common distributions is Anaconda.

---

### 2.2 Installation

**Option 1: with the Anaconda distribution (suggested if you don't have python)** From here you should be able to download the Anaconda installer (be careful to download the correct installer for your OS).

`jupyter notebook` comes preinstalled with the Anaconda python distribution.

**Option 2: using `pip` (if you already have python)** Write the following command in the cmd line:

```
pip install jupyter
```

---

### 2.3 Launching the jupyter server

**Option 1 (with Anaconda):**

- open Anaconda Navigator
- Launch `jupyter`

**Option 2:**

- open terminal
- type: `jupyter notebook`

Regardless of the chosen option, you will see something like this:

At this point, you are not actually running a Notebook, but instead you are just running the Notebook server!

---

### 2.4 Creating a new notebook

- click `New`, then `Python 3`, as in the following image
- it will open a new browser tab with an empty notebook that you can start work on.

---

### 2.5 Opening an existing notebook

- when you lauch the jupyter server, you might see some existing notebooks, as in the following image
- to open a notebook, just click on it

---

## 3 Naming

To rename the Notebook, just click on the title. You should now see an in-browser dialog titled Rename Notebook that lets you rename it.

---

## 4 Running Cells

A Notebook is made of "cells" of code which use the kernel that you chose when you started your Notebook (i.e. Python 3, in this example). In this case, we started the Notebook with Python 3 as kernel, so that means we can write Python code in the code cells.

```
[1]: print("Welcome to the AI for Security course!")
```

```
Welcome to the AI for Security course!
```

To execute a cell, you can just select the cell and click the Run button that is in the row of buttons along the top.

If you prefer, you can just press *Shift+Enter*.

**The order in which you run the code cells is important!**

In general, you should always run the cells sequentially, starting from the top.

Example:

```
[ ]: print("Hello world")
```

```
[ ]: a = 1
     b = 2
```

```
[ ]: print("Value of variable a = %d" % a)
     print("Value of variable b = %d" % b)
```

```
[ ]: print(a + b)
```

---

## 5 “Editing mode” vs. “browsing mode”

Check the color on the left (and around) the selected cell:

- if green, you are in “editing mode”: what you write will be inserted in the cell, and if you move with the arrow keys you move withing the cell
- if blue, you are in “browsing mode”: you can use more keyboard shortcuts, browse cells, etc.

How to move between modes: - “editing” -> “browsing” - run the cell (e.g. Shift+Enter) - press Esc - “browsing” -> “editing”: - press Enter - double click with mouse

---

## 6 Text cells

There are two types of cells: - **code** - **markdown** (I will often refer to them as “text cell”)

(actually, there are more but you will just need these two).

---

## 7 How to add a new cell

- with the “+” in the row of buttons along the top (adds cell below the current one)
- by pressing “b” to add a cell below the current cell (if in “browsing mode”)
- by pressing “a” to add a cell above the current cell (if in “browsing mode”)

By default, newly created cells are code cells

---

# 8 Change cell type

- by selecting the type in the row of buttons along the top
- press “y” to convert to code cell (if in “browsing mode”)
- press “m” to convert to markdown cell (if in “browsing mode”)

Example:

```
[ ]: print("Hello world")
```

---

# 9 Styling Your Text

For the text cells, Jupyter Notebook uses Markdown, which is a markup language that is a superset of HTML. Thus, you can use it to style your cells, create headers, etc.

---

## 9.1 Headers

In Markdown, you write:

```
# Headers
## Header2
### Header3
#### Header4
```

And the outcome is: # Headers ## Header2 ### Header3 #### Header4

---

## 9.2 Creating Lists

You can create a list (bullet points) by using dashes, plus signs, or asterisks.

An example:

```
- first element
- second element
    - inner element
- third element
```

becomes: - first element - second element - inner element - third element

---

### 9.3 Code and Syntax Highlighting

To insert a code example that you don't to actually run, you can use Markdown. For inline code highlighting, just surround the code with backticks (‘). If you want to insert a block of code, you can use triple backticks and also specify the programming language.

An example:

```
print("Hello world")
```

---

## 10 Exporting notebooks

You can do so by doing *File* -> *Download as* and then choosing the desired format. We usually suggest, depending on your needs, *.ipynb* or *.html.*

---