Setting up your Jupyter Notebook IDE

First ensure you understand how to open the Jupyter Notebook environment. To do this you must open the Anaconda launcher and then open Jupyter Notebooks. Follow the instructions in the course notes until you reach the JupyterLab screen in your web browser as seen below:

|  |
| --- |
| *JupyterLab -* [*http://localhost:8888/lab*](http://localhost:8888/lab) |

To organize your files it is imperative that you follow a folder and naming scheme that allows you easily access the files for each relevant assignment. For the purposes of this course you are advised to have have the following file structure:

|  |  |
| --- | --- |
|  |  |

Throughout the course you will be issued assignments as *.ipynb* files. After downloading the file from the Noroff Moodle website you should copy the file to the corresponding assignment folder’s src folder. From here you should be able to open the Notebook inside the JupyterLab IDE simply by double clicking on the file within its corresponding assignment src folder.

# How to do assignments

The source file you will be given for submission assignments will be a Jupyter Notebook file (*.ipynb*). The assignment objectives will be text cells. When doing the assignments you are required to type your code in the cell directly beneath the objective and execute the code block.

You are welcome to add additional cell blocks besides the ones provided, however the original text cells must remain in the Notebook so that the marker can have a clear indication between the output of your code. Any blank cells you create will be ignored, though leaving blank cells in the Notebook is considered a bad practice.

Carefully read the objectives in each text block as they will guide you on what to do for that assignment, as well as give you a general point value for completed objectives. After completing an assignment you must upload the edited *.ipynb* file back onto the Noroff Moodleboard as indicated.

Cell structure

Let’s go over the typical Jupyter Notebook structure. Each Notebook is made up of several cells which can either be text cells (called Markdown), or Python cells. In the next lesson we will be jumping into how to use the Python cells, but for this lesson we will be covering the use of Markdown.

First, review the course notes and read the **Overview of the text editor** chapter. Practice with a new empty notebook by inserting, moving, and deleting cells. Once you have a general grasp of how to use the interface we can discuss Markdown.

Markdown is a lightweight markup language that uses a plain text formatting syntax to transform plain text to a styled format. For example if we wish to transform text into a heading we’d simply add a hash mark before it.

An example of this can be seen below.

|  |
| --- |
| # My heading, wow! |
| My heading, wow! |

Unlike other text applications like Microsoft Word, or Google Documents, markdown has no user interface that allows you to apply formatting to text. Thus you will need to lookup the desired formatting in a Markdown reference sheet (cheatsheet). For the purposes of this course we won't be extending much further than simple headers, bolds and italics, however we may require you post images and links in your reports.

Follow this link to see a breakdown of what is possible in markdown: <https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>