# Jupyter Notebooks in VS Code

Practical 1, part C

## Contents



Jupiter Notebooks allow to keep the code, results and comments in the same file (somewhat resembling R-Markdown documents); you may look at the basics here (in just 6 min!): https://code.visualstudio.com/docs/datascience/jupyter-notebooks

Jupiter Notebooks are very popular in bioinformatics because they provide an excellent environment for interactive exploration, at the same time helping to keep track of the analysis facilitating reproducible research.

Jupiter Notebooks ***extension*** should be installed in VS Code when you install Python extension. If you installed Python with Anaconda (not Miniconda) you also have the ***original*** Jupiter Notebook installed. However, we will not use the ***original*** Jupiter Notebook directly. We will only use its implementation in VS Code.

Close the previous project folder. Make a new project folder called `i_love_jupyter`.

## Appearance of Jypiter Notebooks in VS Code

Add a new file, call it `i_love_jupyter.ipynb`.

The file extension stands for **I**nteractive **PY**ton **N**ote**B**ok.

VS Code shall recognize this extension and show a screen similar to this:

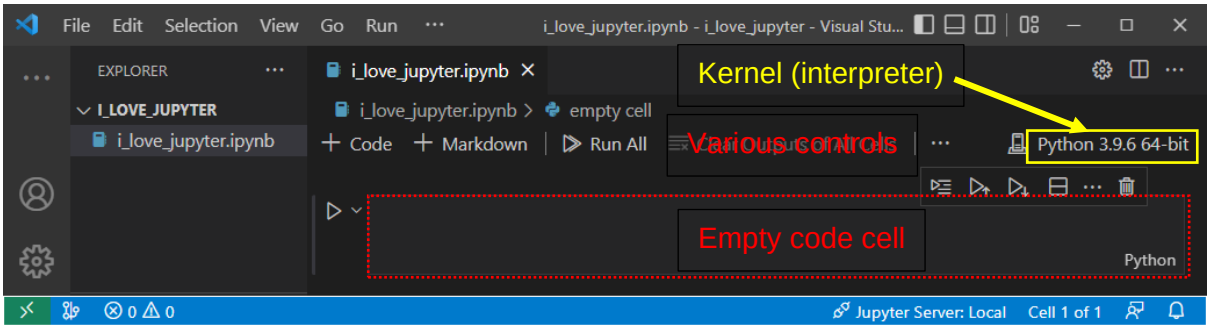



Figure 1: Empty Jupyter Notebook in VS Code.

If needed, click on "+ code" button to create an empty code cell. If you see "Select Kernel" prompt in the top right corner, click and select a Python environment (any environment will be OK for now).

Type `print("Hello Jupyter!")` in the code cell and click the arrow on the left of the cell:

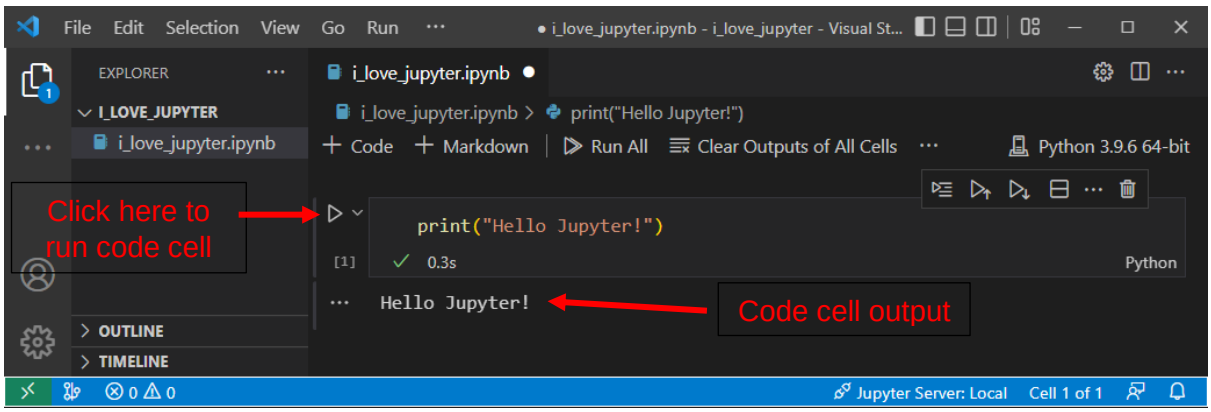



Figure 2: Hello Jupyter notebook

VS Code may ask you to install `ipykernel` package into the current environment to work with Jupyter Notebooks: it's OK. It is safe to add `ipykernel` even to the global or base Python environments. Also, Windows may ask your permission to allow some network access: agree to it (it might be necessary to access your local Jupyter server).

## Working with cells in the Jupyter notebook

Jupyter notebooks organise code by "cells". In addition to the ***code*** cells, notebooks also allow "***markdown***" cells, which are used for text, not for code. The content of markdown cells is not executed, and it can be easily formatted, i.e. rendered in different styles (e.g. bold, large, header, list styles etc).

Let's add a markdown cell above our code cell. You may add it by clicking on "+ Markdown" button. If your version of VS Code doesn't show this button, click on "…" symbol at the top right corner of the code cell. Then click on "Insert cell", then "Insert Markdown Cell Above":

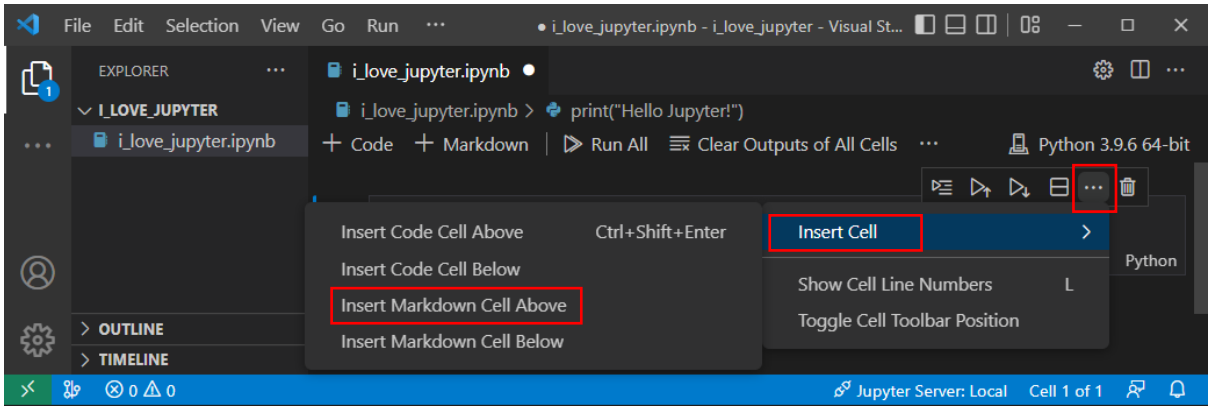



Figure 3: Adding a cell to Jupyter notebook

A new cell will appear as shown on Figure 4. You can see the type of the cell ("Markdown") in the right bottom corner of the cell. Type `# My first Jupyter Notebook`, then click tick-mark in the top right corner of the cell: the cell will be rendered. To edit the markdown cell again, you may click the pencil symbol in the cell controls group (on the top right corner of the cell) that appears instead of the tick-mark after the cell has been rendered (Figure 5).

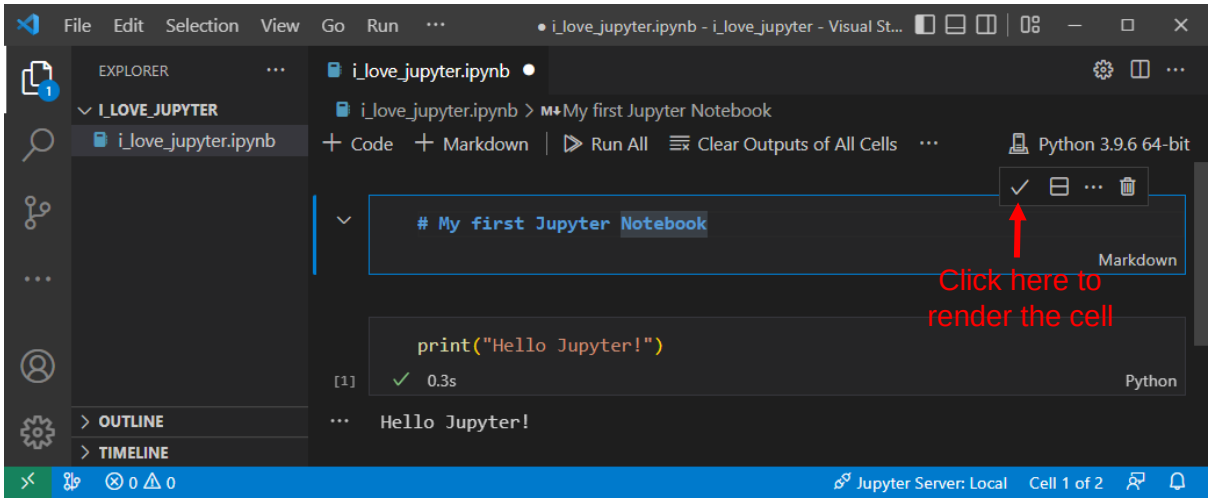



Figure 4: Editing Markdown cell in Jupyter notebook

Compare cell control groups on Figures 2, 4 and 5: you may see that the controls change depending on the type of the cell and cell state: whether it's a code or markdown cell, whether the code cell is active, or whether the markdown cell is being edited or rendered.

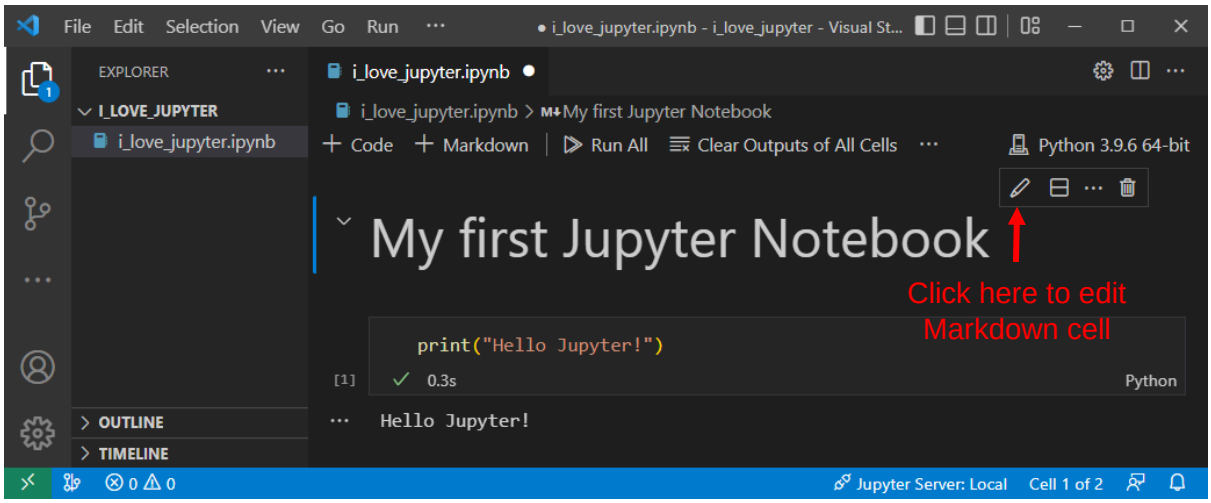



Figure 5: Rendered Markdown cell in Jupyter notebook

Let's edit the markdown cell. Remove "`#`" symbol at the beginning of your text in the markdown cell and then render it again. What happened? Return the "`#`" symbol and render again to use Heading 1 style.

Add one more markdown cell below the code cell. Type "`### I love Jupyter`". Three # symbols is not a typo: it's to use Heading **3** style. Render and see how it looks.

## Adding a cell with emojised code ♡

Add a ***code cell*** at the bottom of your notebook. Type the code, as shown below:

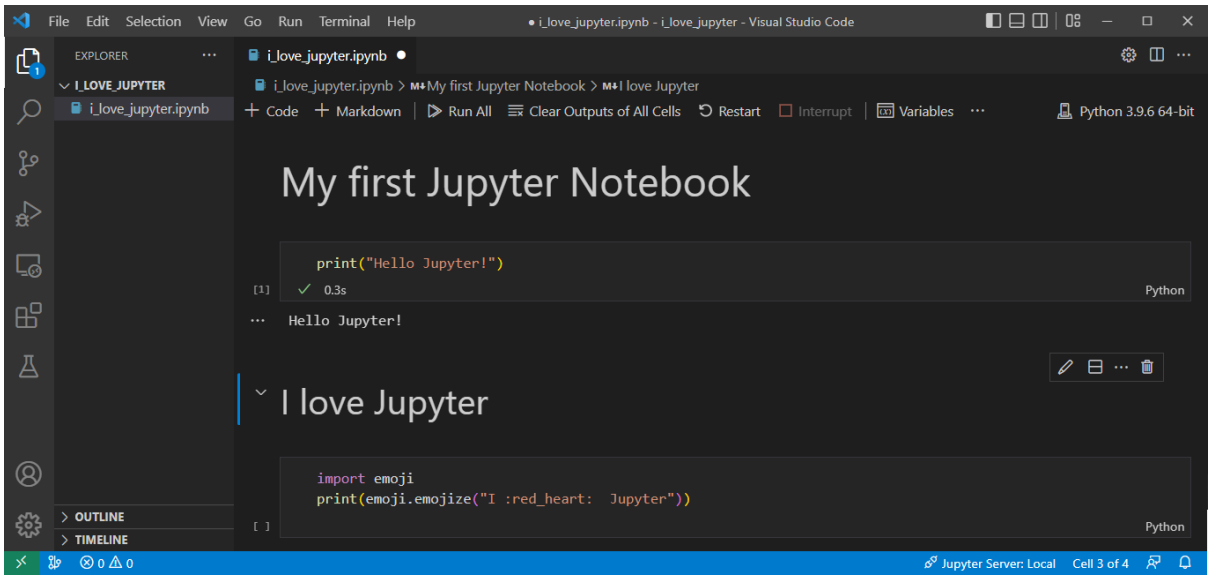



Figure 6: A new code cell with emojized code

However, if you try to run this cell, it is likely that you will get an error: missed emoji package.

## Selecting Python kernels

You may remember that to use emojised code we need the environment with **emoji** package installed. We already have such environment in Conda. Click on the currently used Kernel on the **top right** corner of the Jupyter notebook, and select the **emoji_env** Conda environment, which we created during the previous exercise:

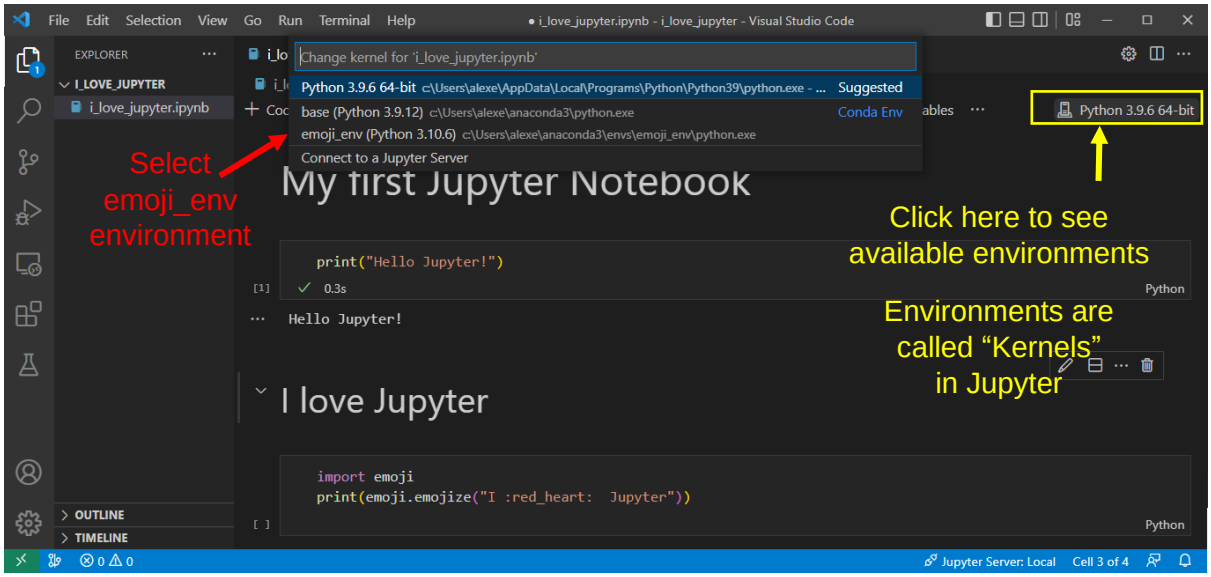



Figure 7: Selecting environment in Jupyter notebook

Note that in Julyter world Python ***environments*** are called Python ***kernels***.

After you selected **emoji-env**, it should be shown on the top right corner of the notebook:

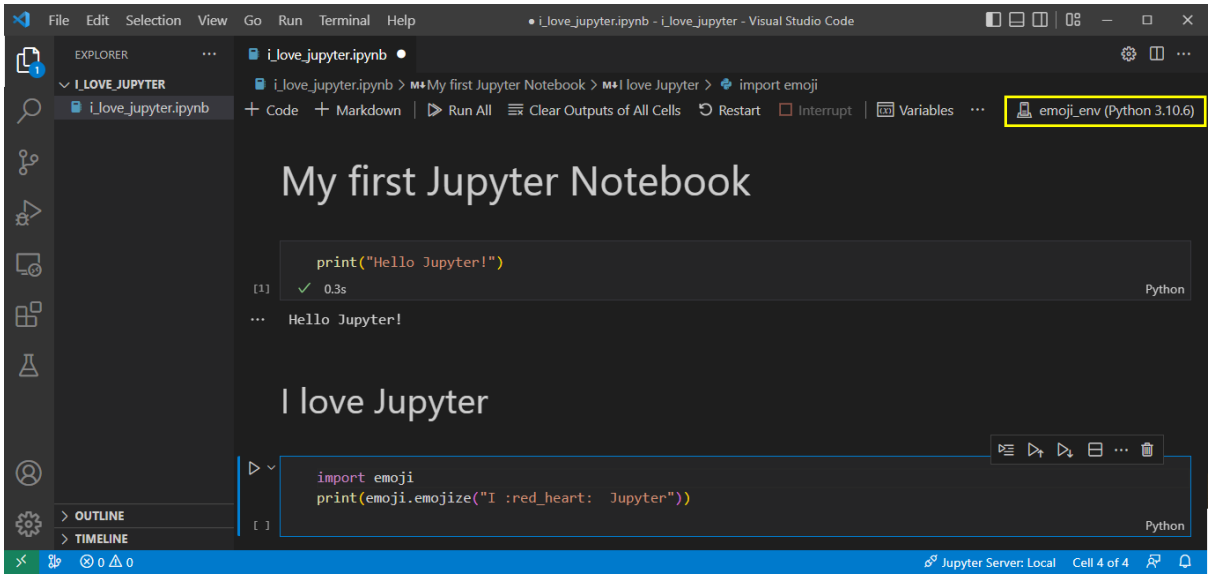



Figure 8: Kernel with emoji package in Jupyter notebook

Now you can run the emojized code (hint: click on the triangle on the left of the cell).

When you run Jupyter Notebooks with the **emoji_env** for the first time, you may be asked to install `ipykernel` again. If this happens, just install it again: previously you might have run the Jupyter Notebook with a different Python kernel (e.g. in the global environment). So, `ipykernel` package may still be missed in the **emoji_env**. After you installed all packages in your current environment, then you may see the code output shown on Figure 9.

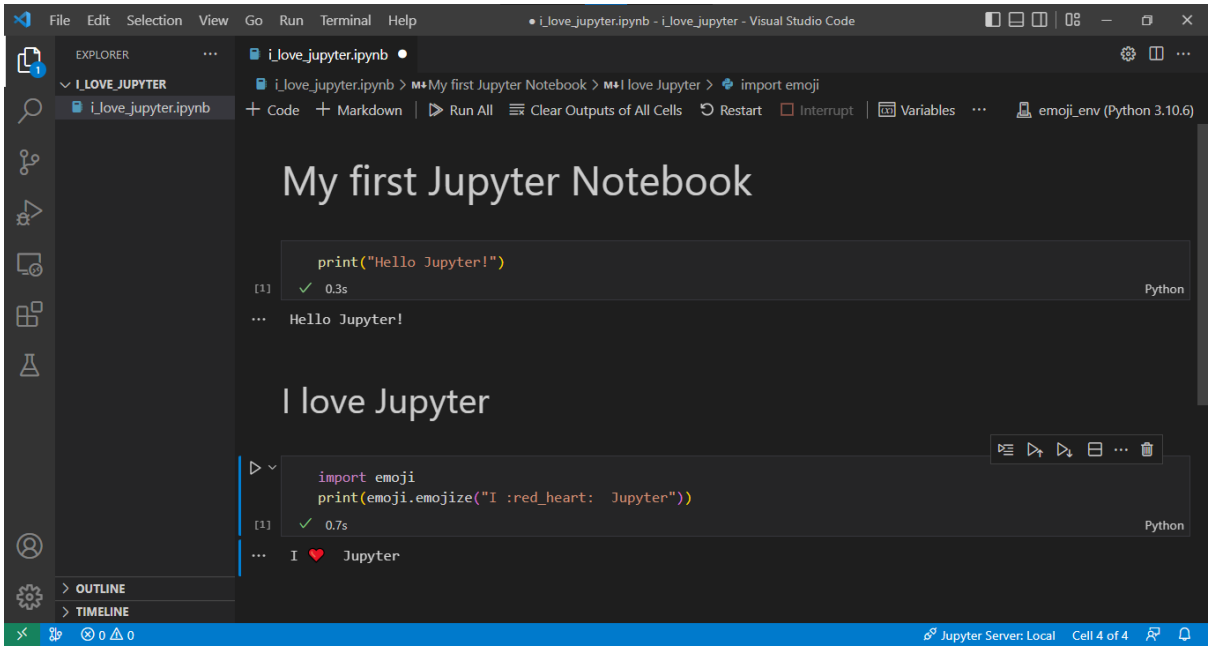



Figure 9: Your first Jupyter notebook complete !

## Exporting Jupyter notebook content to HTML file

Exporting the analysis results to HTML file format, which can be viewed in any computer that has an HTML browser, is a ***very*** handy feature of the Jupyter notebooks. Also, keeping nice HTML-formatted logs of your analysis is good for reproducible research.

Click on "…" symbol in top right corner of the notebook, and select "Export" as shown in the figure below. Then select HTML file format (don't select PDF because it requires installing some additional libraries system-wide).

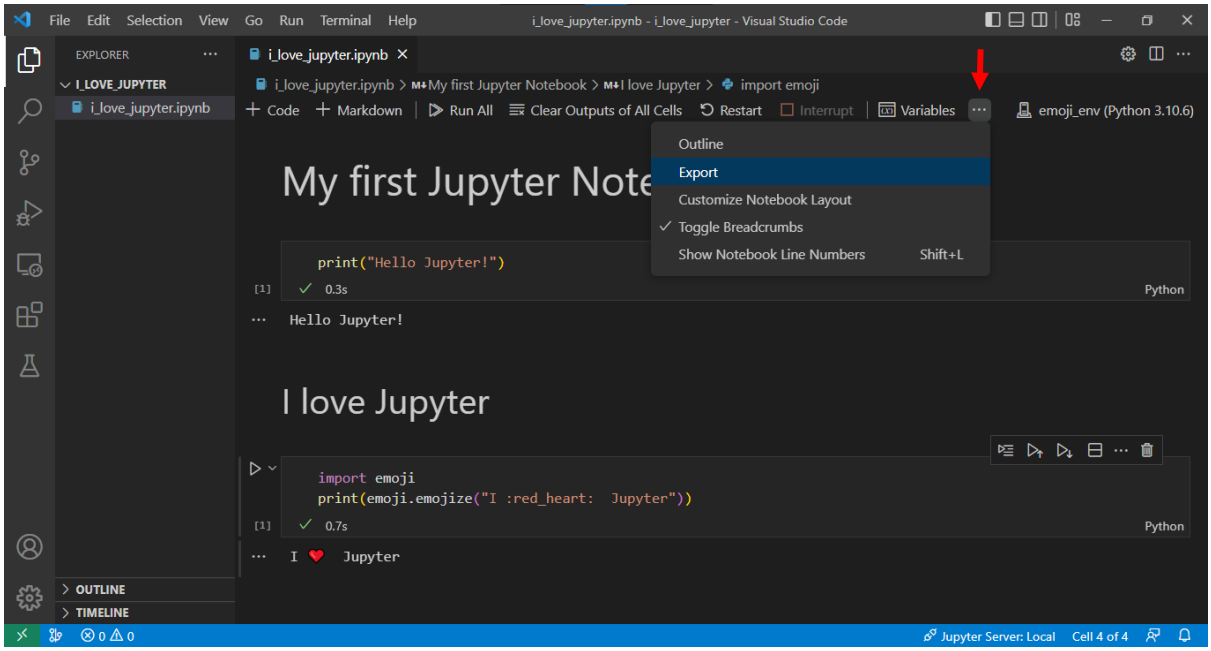
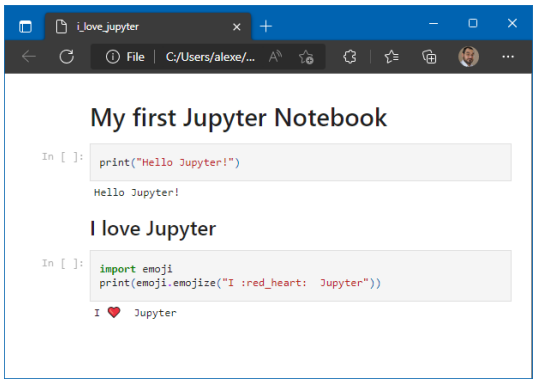



Figure 10: Exporting your Jupyter notebook results

When exporting to HTML for the first time, VS Code may need to add some more packages to your **emoji** environment (e.g. `notebook` and `nbconvert` packages). If the first attempt failed because of the missed packages, just try it again after all the packages installed[1].




When the export is successfully completed, a new html file should appear in your project folder, which you can open by any internet browser available on your PC:

**Well done! You have finished the 3rd part of the Practical 1**

[1] If the error persists, you may need to install `nbconvert` package manually. Open terminal, activate your emoji environment (e.g. `conda activate emoji-env` ) and install the package: `conda install nbconvert` . Then try to export Notebook to HTML again.