

---

# M348 JUPYTERLITE LEARN7 / VIRTUAL CAMPUS RESOURCE

**M348 Module Team**  
The Open University

2<sup>nd</sup> Oct, 2025

Developed by  The Open  
University

---

## 1 Introduction

The *M348 JupyterLite Learn7 / Virtual Campus Resource* is a self-contained HTML interactive resource that provides a powerful JupyterLab environment that you can access using your web browser.

All code is executed within the browser, and no files need to be loaded from external websites. This means that the application can be run purely from an Open University Learn7 / Virtual Campus module website.

Files can be mounted into the JupyterLab environment from your computer, edited in the browser, and saved back to your computer.

*If you have an issues or problems running the application, or any comments or feedback about how we might improve it, please pass them on to the module team via your tutor.*

## 2 Accessing the M348 JupyterLite Environment on Learn7 / Virtual Campus

The *M348 JupyterLite Environment* can be accessed from the *Resources* area of the module website and loaded as an embedding environment within an activity page.

### 2.1 Installation testing

Once the application has been launched and accessed in the browser, you should be presented with a view of a JupyterLab environment.

Two notebooks are bundled into the environment: `M348-styling-test.ipynb` and `M348-code-tests.ipynb`.

In the local JupyterLab file browser, double click on the `M348-styling-test.ipynb` notebook to open it. It should render a notebook containing coloured cells and an embedded image, Figure 1.

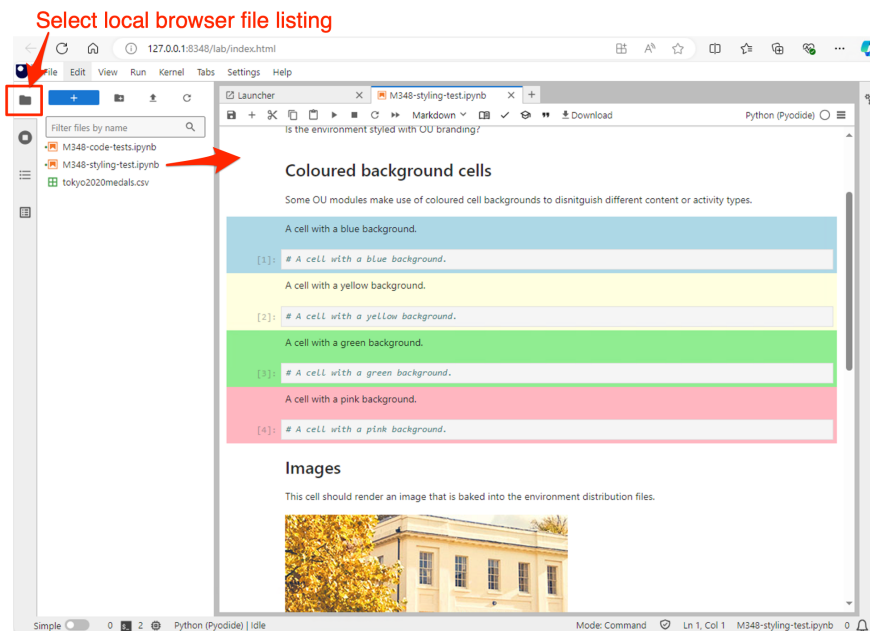


Figure 1: Screenshot of JupyterLab UI showing rendered styling text notebook with coloured background cells and displaying an embedded image

Double click on the `M348-code-tests.ipynb` notebook to open it. From the Run menu, select *Run All Cells*.

Hopefully, each cell will run correctly (indicated by a green block in left hand margin of each cell). *Note the test to read a written CSV data file may fail. Everything else should pass.*

### 2.2 Accessing the Module Notebooks

The JupyterLite / Jupyterlab environment provides an environment for working with Jupyter notebook documents, which have the file suffix `.ipynb`.

These documents will be distributed in their own `.zip` file archive. Unzip the file archive and put the notebooks in a location where they can be saved. You may find it convenient to place the notebooks in an `M348-25J` folder in your user home directory (for example, `C:\Users\<username>\M348-25J` on Windows, or `/Users/<username>/M348-25J` on a Mac).

There are two ways of making the notebooks accessible in the JupyterLite environment:

1. Uploading the notebooks to the JupyterLite environment, described in Subsection 2.2.1 and then manually downloading them again if you want to keep an updated copy on your desktop. *If your browser can save*

*browser sessions, any notebooks you upload to JupyterLite will be saved to browser storage and will be available in future sessions.*

2. Allowing the JupyterLite environment to access a folder on your computer using the *Local Filesystem Access* extension, described in Subsection 2.2.2.

### 2.2.1 Uploading Notebooks to the JupyterLite Environment

To upload files or zipped file archives, click the up arrow ("Upload Files") icon in the JupyterLab file browser toolbar. *It is recommended that you upload files to the top level JupyterLite directory.*

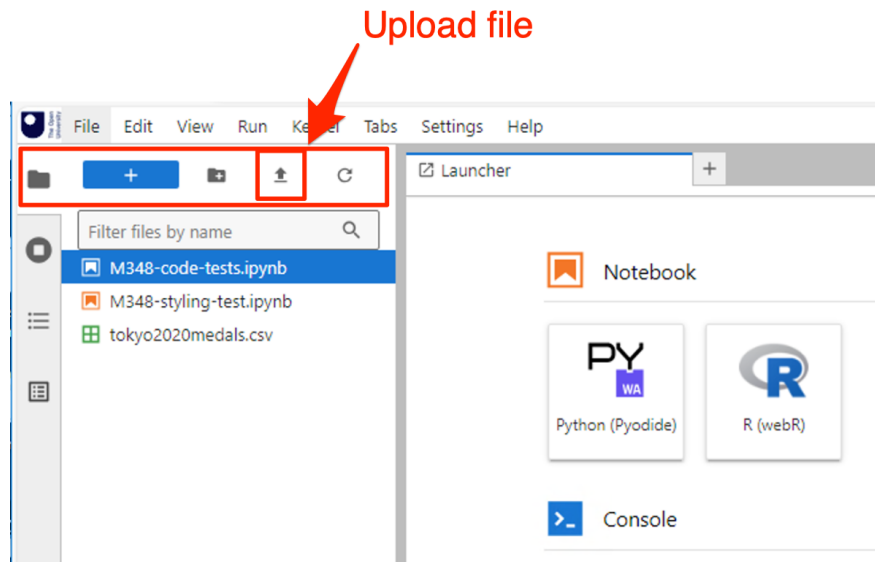


Figure 2: JupyterLab file browser, file upload button (up arrow icon)

If you upload a zip file archive, you will need to unzip it in order to access the files it contains.

A zip file uploaded to the top level JupyterLite directory can be unzipped in the following way:

1. Open a new R notebook by clicking on the R (webR) icon under Notebook.
2. In the code cell type the following command: `unzip("filename.zip")` where `filename.zip` corresponds to the name of the zip file. (To preview a listing of the files contained in the zip file, run `unzip("filename.zip", list=TRUE)`)
3. Run the code cell by clicking on *Run* to unzip the file. The unzipped files should appear in the file list in the sidebar.

*If you upload the zip file to a subdirectory in JupyterLite, you will need to set the path to the file. Irrespective of which directory a notebook file is in, R also starts to run in the top-level directory, which has the absolute path `/drive`. If your zip file is in the directory `unit_1`, you need to specify the path, such as `unzip("./unit_1/filename.zip")` (relative path) or `unzip("/drive/unit_1/filename.zip")` (absolute path). The relative path is set relative to the current working directory, which can be identified using `print(getwd())`. The current working directory can be set using `setwd('./unit_1')` (set current working directory relative to current working directory) or `setwd('/drive/unit_1')` (set current working directory using an absolute path).*

## 2.2.2 Local Filesystem Access

The [jupyterlab-contrib/jupyterlab-filesystem-access](https://github.com/jupyterlab-contrib/jupyterlab-filesystem-access) extension adds local file system access to the JupyterLab environment (*this may not work in all browsers*).

This extension allows you to select a directory from your local filesystem (which is to say, the files on your own computer) and access that directory from JupyterLite, Figure 3.

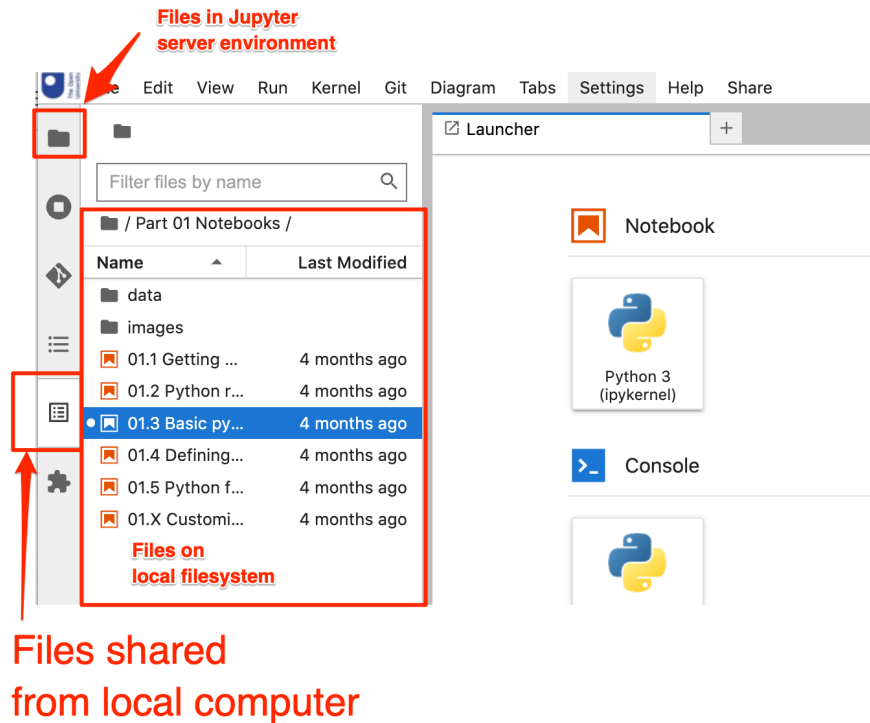


Figure 3: Local file browser.

Screenshot of a sidebar that lists files mounted into the browser from the local file system. *Note that files are shown for a different module.*

From the “Open new folder” button in the *Local File System* tab, open a directory on your computer. *You may be prompted for permission.* Sharing the folder allows the browser to read and open the contents of the folder on your computer in the Jupyter environment, Figure 4.

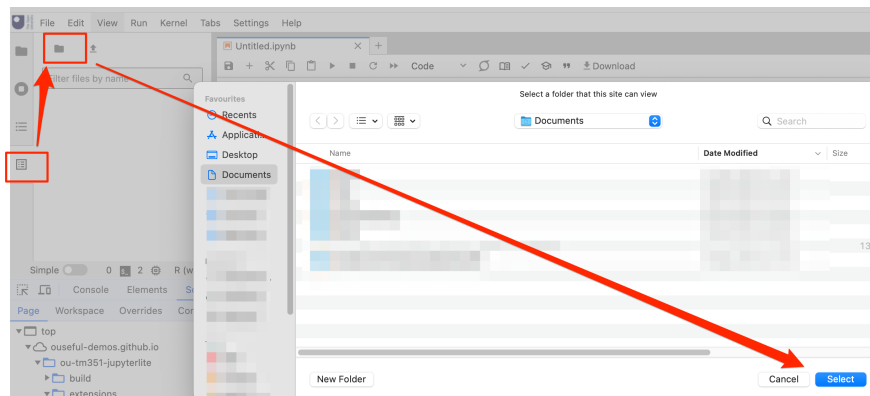


Figure 4: Shared computer filesystem.

Screenshot showing how to mount a directory from your computer into the JupyterLab UI.

*No files are copied: the browser is just given permission to read to them from that directory.*

You should then be able to see files shared from your computer into JupyterLab, Figure 3.

You can also create files in the Jupyter environment and save them back to your computer. For example, in the local file system browser, right click and select *New File*. You may be prompted for permission to write to the shared directory (the browser cannot write outside that directory). The file should appear in the folder on your computer. If you open and edit the file in the Jupyter environment, then save it, it should be saved to your computer.

Read and write permissions over the shared directory on the local filesystem are granted to the JupyterLite environment (which runs inside your browser) for only as long as it is open in your browser.

Note that *whilst you can preview the contents of CSV data files listed in the local filesystem access file listing, you **cannot** read from, or write to, these shared data files from R code.*

## 2.3 Reading and writing CSV data files

You can read and write CSV data files to the JupyterLite file area using the `read.csv(FILENAME)` and `write.csv(DATAFRAME, FILENAME)` commands. If you specify a simple filename, this will be read from or written to that filename in the current working directory. If a file cannot be opened, you will see a “cannot open the connection” error.

The location of the current working directory can be identified using `print(getwd())`. The current working directory can be set using `setwd('./unit_1')` (set current working directory relative to current working directory) or `setwd('/drive/unit_1')` (set current working directory using an absolute path).

If you have mounted files into JupyterLite using the local filesystem access extension, you can *preview* the contents of a CSV file mounted from the desktop in JupyterLite by double clicking on the file in the local filesystem access sidebar listing, *but you **cannot read or write files** to the desktop from R code*. Instead, you will need to *upload* any data files you want to access from R code to JupyterLite and *download* any data files that you write to if you want to keep a copy of them on your desktop.

## 2.4 Downloading Files from the JupyterLite Environment

To download a copy of an open notebook, click on the *Download* link in the notebook toolbar. This *should* automatically save the notebook before downloading it, but we advise that you *always* save a notebook before downloading it, just to be sure.

To download a file from the file browser sidebar, right click on the file and then select *Download*, Figure 5.

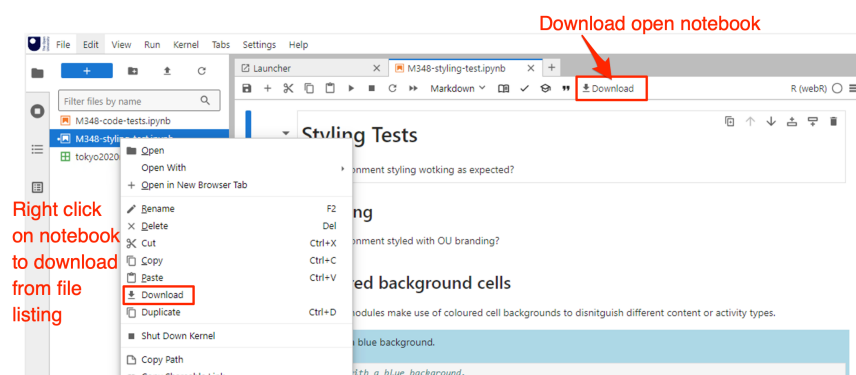


Figure 5: Downloading files from JupyterLab

Screenshot of JupyterLab showing two ways of downloading notebooks: from a Download button in an open notebook; and by right clicking on a notebook in the file browser, then selecting the “Download” menu option.

## **2.5 In Case of Further Issues**

If you encounter any other issues with running the desktop application, using JupyterLite, or working with the notebooks, please raise them with your tutor or the M348 module team.