

# Introduction to the Jupyter Hub environment

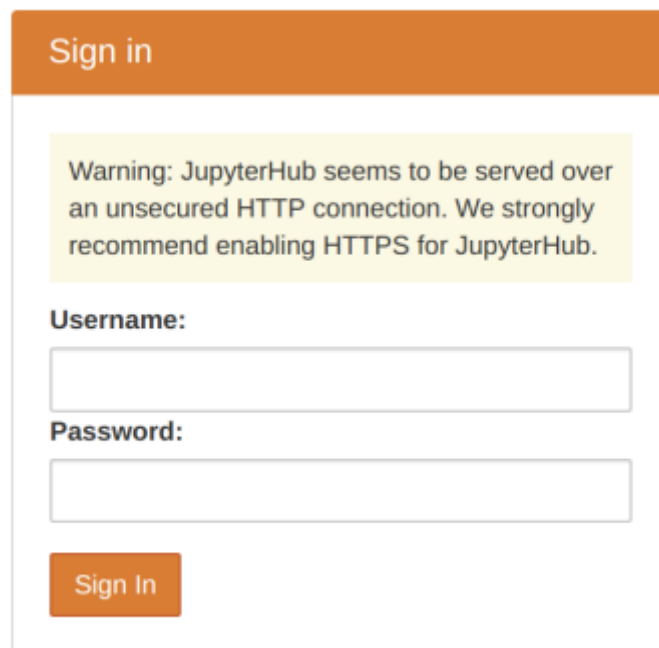
---

## Introduction

Jupyter Hub is an interactive multi-user environment to run Python Notebooks and Linux terminal sessions. It contains a file manager, a file editor, a notebook interface and a terminal emulator.

## Logging in Jupyter Hub

To access Jupyter Hub, go to the following link: <http://comppbio20202.ddns.net> (the browser will probably complain about the site being insecure, but you can safely ignore the warnings and proceed). Upon opening the above link, you will be presented with the following panel:



The image shows a login panel for Jupyter Hub. It has an orange header with the text "Sign in". Below the header is a yellow warning box that reads: "Warning: JupyterHub seems to be served over an unsecured HTTP connection. We strongly recommend enabling HTTPS for JupyterHub." Under the warning box are two input fields: "Username:" and "Password:". Below the password field is an orange button labeled "Sign In".

where you should use the username and password provided by the instructor.

If you wish to change the password please ask the instructor help on how to do so.

## The Jupyter Hub file manager interface









Upon logging in, you will be presented with the **file manager interface** interface of the Hub.

Select items to perform actions on them.

Upload

New ▾



<input type="checkbox"/> 0 ▾	 /	Name ▾	Last Modified	File size
<input type="checkbox"/>	comparative_modelling		2 hours ago	
<input type="checkbox"/>	Untitled Folder 1		2 hours ago	
<input type="checkbox"/>	 Programmatic_Access_PDB.ipynb	Running	an hour ago	4.96 kB
<input type="checkbox"/>	 Python_Short_Tutorial.ipynb		7 days ago	37.5 kB
<input type="checkbox"/>	 Python_Short_Tutorial_New.ipynb	Running	2 days ago	67 kB
<input type="checkbox"/>	 Python_Short_Tutorial_v3.ipynb	Running	a day ago	135 kB
<input type="checkbox"/>	 Untitled1.ipynb	Running	a day ago	27.3 kB
<input type="checkbox"/>	 test.dat		11 days ago	91 B
<input type="checkbox"/>	 untitled.txt		11 days ago	0 B

The file manager displays a list of a user's current files. Three types of file can be seen in the above image: folders (folder icon), notebook files (book icon) and a data files (sheet icon). Different things will happen when you click each type of file:

- **Folder:** clicking these will bring a new list presenting the files contained in the folder you just clicked.
- **Notebook:** clicking these will open the *Notebook Interface* for the file you just clicked (more of on this below).
- **Datafile:** clicking these will open the *Editor Interface* for the file you just clicked (see below).

On the left-hand side, there is a pull-down name "New", where you can create new Notebooks, Data Files, Folders or Terminal Sessions (we will discuss Terminal Sessions in greater detail below).

## The Notebook Interface

If you clicked on a file with the notebook icon, it will open in the Notebook Interface, where you can edit, add, remove or execute *cells* in the notebook. Think of cells in the Notebook a little like snippets of programming code or data that can produce and output when executed. The best way to learn how notebooks work is to actually use one: try openin the notebook named "Introduction\_to\_Python.ipynb".

For now, notice that cells have a "In [number]" preceding them. You can "execute" (run the code) in a cell by clicking on it and pressing the "Shift" and "Enter" keys simultaneously. A new box will appear below the executed will, with an "Out [number]" preceding it. As you may have guessed, this is the *output* produced by the cell upon running the code contained in it.

Not all cells have Python code commands. Some have just (almost) plain text, and are used to write descriptions, explanations, headings, etc. The two types of cells named *code cells* and *markdown cells*. To switch between the two types, click on a cell, press the "Esc" key, and the Y key (code cell) or M key (markdown cell) to switch the cell type.

On the menus you have various commands to insert, delete, copy and paste cells, and also to run sets of cells or execute the entire notebook.

It is important to note that code executed in a cell can affect the result of cells below it. You must keep this in mind when trying to execute cells that are in the middle of a notebook (they may produce error messages complaining about missing variables, functions or libraries).

## The Terminal Interface

If choose "Terminal" on the "New" pull-down of the File Manager Interface, a new tab will open in the browser with a black screen with a prompt (something like "jupyter-xxxx@jlab-mini:~\$") and a blinking cursor. This may appear daunting at first, but it is actually quite simple: the terminal is waiting for your commands. The is called a "command line interface".

Try typing "ls" and pressing the "ENTER" key. A list of the files in you Jupyter Hub home folder will appear, followed by a new line starting with the same prompt and blinking cursor - the system executed the "ls" command and is now waiting for a new command. That's all that there is to it, really: just typing commands, and watching the resulting output. Let's list a few very basic (and important) commands:

- `cd <folder>` - changes to the named folder below your current folder (don't actually type the "<" and ">", they are a notation meaning that you should actually replace "" with the name of your folder
- `ls <file/folder>` - without arguments (just `ls`) this commmand will list the contents of the current folder. If given a folder as argument, it will list the contents of that folder. If given a file as argument, it will simple list that file (or produce an erroer message if the file doesn't exist).
- `cat <file>` - list the contents of a file. Be careful not to run this command on binary files (not containing text) or very large files, or you will most likely get your terminal stuck.
- `pwd` - "pwd" stands for "print working directory" and that is exactly what it does: it prints the name of your current directory (another name for folder). Good to avoid getting lost amidst multiple folder changes.
- `nano <file>` - brings up a simple editor to edit the contents of named file.
- `cp <file> <file/folder>` - copy the first file to a new file in the same folder (it will have to have a different name) or to another folder, depending on whether the second argument is a file or a folder.

To navigate the folder tree, there are two important special folder names:

- `.` - means "the folder we are in"
- `..` - means "the previous folder"

Complete file paths are specified thus: folder1/folder2/folder3

For instance:

```
cd ..
```

will will move one folder up (or down, depending on your perspective) the folder tree, leaving us in "folder1".

On very important fact about the command-line interface: Python code can be run directly on the command line interface, for instance:

```
python my_script.py
```

will run the instructions contained in the "my\_script.py" file (the ".py" extension is used to tell the computer these are Python files).

In this course we will use the command line interface to run various software applications and programs.

## **What do to next**

Go to the browser tab where you have loaded the "Introduction\_to\_Python.ipynb" and study it through.