

# TRIQS Tutorials: Getting Started

- Login via ssh

```
ssh user@jouvence.ccs.usherbrooke.ca
```

- Clone the TRIQS tutorials

```
git clone /soft/public_soft/triqs_tutorials tutorials
```

- Run iPython Notebook on a node

```
/soft/bin/srun_jupyter
```

- You will be presented a URL for the notebook access

```
https://cpXXX-XXXX.jouvence.ccs.usherbrooke.ca/?token=YYYY
```

# TRIQS Tutorials: Getting Started

2

 jupyter

Quit

Logout

Files Running Clusters Nbextensions

Select items to perform actions on them.

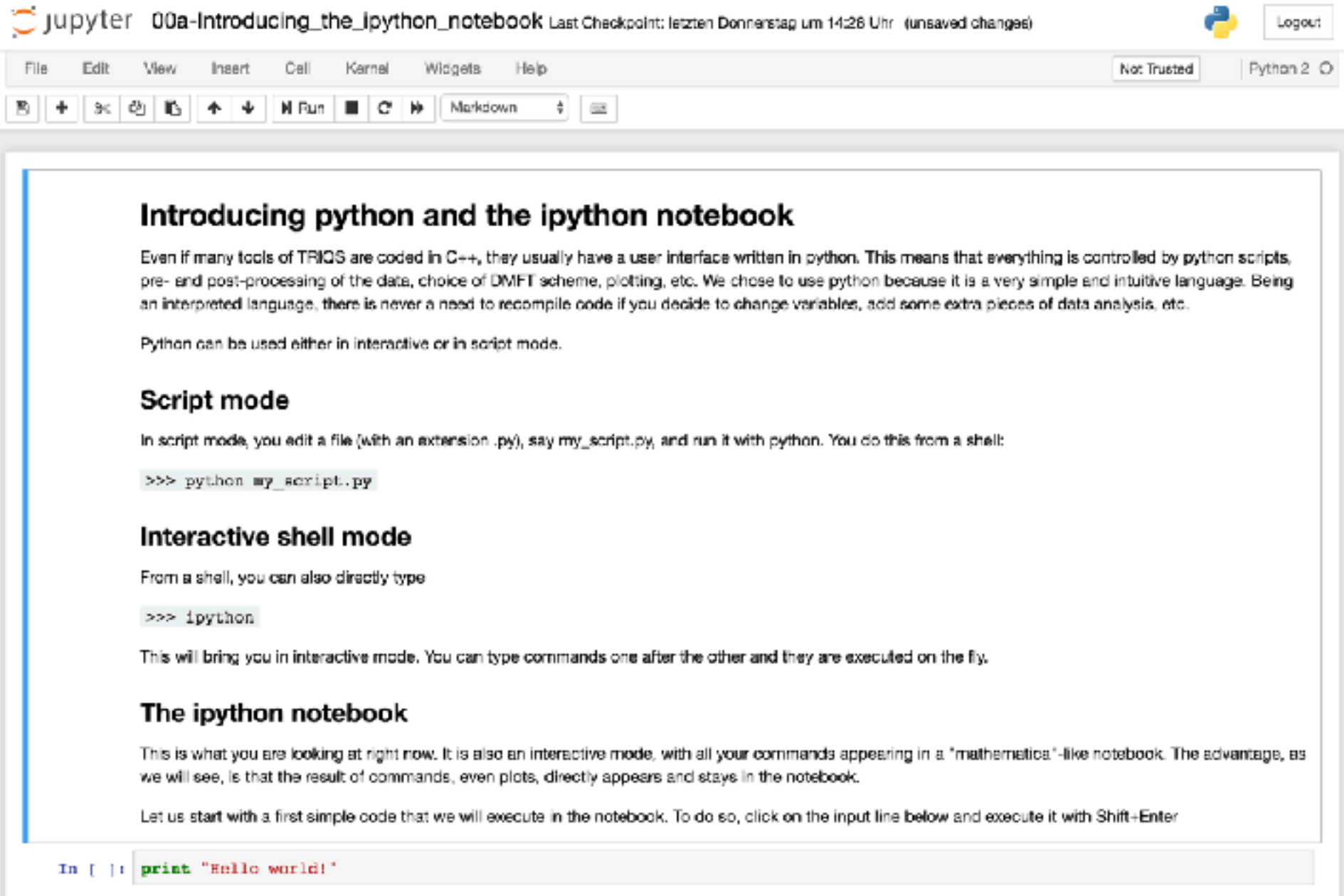
Upload

New



/ clean_home		Name	Last Modified	File size
<input type="checkbox"/>	..		vor ein paar Sekunden	
<input type="checkbox"/>	tutorials		vor einer Minute	
<input type="checkbox"/>	Advanced		vor 4 Minuten	
<input type="checkbox"/>	ctm_tutorial		vor einer Minute	
<input type="checkbox"/>	TRIQSPaperExamples		vor 4 Tagen	
<input type="checkbox"/>	TRIQSTutorialsPython		vor 4 Minuten	
<input type="checkbox"/>	README.md		vor 4 Tagen	125 B
<input type="checkbox"/>	TPSC		vor 10 Stunden	
<input type="checkbox"/>	00a-Introducing_the_ipython_notebook.ipynb		vor 7 Minuten	12.1 kB
<input type="checkbox"/>	00b-Matplotlib_Examples.ipynb		vor 7 Minuten	10.2 kB
<input type="checkbox"/>	01-Greens_functions.ipynb		vor 7 Minuten	24.8 kB
<input type="checkbox"/>	02-Greens_functions_solution.ipynb		vor 7 Minuten	7.83 kB
<input type="checkbox"/>	03-Archiving_your_data.ipynb		vor 7 Minuten	6.12 kB
<input type="checkbox"/>	04-IPT_and_DMFT.ipynb		vor 7 Minuten	8.01 kB
<input type="checkbox"/>	05-IPT_and_DMFT_solution.ipynb		vor 7 Minuten	6.12 kB
<input type="checkbox"/>	06-Operators.ipynb		vor 7 Minuten	4.6 kB
<input type="checkbox"/>	07-Introduction_to_the_CTHYB_solver.ipynb		vor 7 Minuten	10.9 kB
<input type="checkbox"/>	08-Single-orbital_Hubbard_with_CTQMC.ipynb		vor 7 Minuten	6.02 kB
<input type="checkbox"/>	09-Single-orbital_Hubbard_with_CTQMC_solution.ipynb		vor 7 Minuten	13.3 kB
<input type="checkbox"/>	10-Two-orbital_Hubbard_with_CTQMC.ipynb		vor 7 Minuten	9.01 kB
<input type="checkbox"/>	11-Two-orbital_Hubbard_with_CTQMC_solution.ipynb		vor 7 Minuten	9.51 kB
<input type="checkbox"/>	12-VDMFT_Hubbard.ipynb		vor 7 Minuten	11.6 kB
<input type="checkbox"/>	13-VDMFT_Hubbard_solution.ipynb		vor 7 Minuten	7.9 kB

# TRIQS Tutorials: Getting Started



The screenshot shows a Jupyter Notebook interface. At the top, the title bar reads "jupyter 00a-introducing\_the\_ipython\_notebook" with a "Last Checkpoint: letzten Donnerstag um 14:28 Uhr (unsaved changes)" and a "Logout" button. Below the title bar is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". To the right of the menu bar are "Not Trusted" and "Python 2" indicators. Below the menu bar is a toolbar with icons for file operations, running, and a "Markdown" dropdown menu. The main content area contains the following text:

## Introducing python and the ipython notebook

Even if many tools of TRIQS are coded in C++, they usually have a user interface written in python. This means that everything is controlled by python scripts, pre- and post-processing of the data, choice of DMFT scheme, plotting, etc. We chose to use python because it is a very simple and intuitive language. Being an interpreted language, there is never a need to recompile code if you decide to change variables, add some extra pieces of data analysis, etc.

Python can be used either in interactive or in script mode.

### Script mode

In script mode, you edit a file (with an extension .py), say my\_script.py, and run it with python. You do this from a shell:

```
>>> python my_script.py
```

### Interactive shell mode

From a shell, you can also directly type

```
>>> ipython
```

This will bring you in interactive mode. You can type commands one after the other and they are executed on the fly.

### The ipython notebook

This is what you are looking at right now. It is also an interactive mode, with all your commands appearing in a "mathematica"-like notebook. The advantage, as we will see, is that the result of commands, even plots, directly appears and stays in the notebook.

Let us start with a first simple code that we will execute in the notebook. To do so, click on the input line below and execute it with Shift+Enter

```
In [ ]: print "Hello world!"
```

# TRIQS Tutorials: Useful Links

- Binder: Notebook Access on the Web

[mybinder.org/v2/gh/TRIQS/docker/master](https://mybinder.org/v2/gh/TRIQS/docker/master)

- The TRIQS Documentation

[triqs.github.io/triqs/master/contents](https://triqs.github.io/triqs/master/contents)

- Docker: Run the Notebook on your Computer

[hub.docker.com/r/flatironinstitute/triqs](https://hub.docker.com/r/flatironinstitute/triqs)



- Install TRIQS yourself

[triqs.github.io/triqs/master/install](https://triqs.github.io/triqs/master/install)

Ubuntu Packages (experimental), Easy Cluster Setup with Singularity,  
Compile TRIQS yourself (Advanced)

