# Connecting Cultural Heritage Collections Data with Linked Art

*Software Installation Instructions*

# Introduction

We will support you in a hands-on session looking at the processes involved in transforming cultural heritage data to Linked Data. We will use code notebooks to transform, reconcile, and visualise cultural heritage data using the Linked Art data model. In preparation, you will need to have some software installed on your laptop.

# Setup

We will use Anaconda that will include JupyterLab where we will work with Jupyter notebooks containing Python code from the Linked Art project

- → Anaconda
    - → JupyterLab
        - → Jupyter notebooks (Linked Art)
            - → Python

# Linked Art Notebooks

The notebooks that we will be using are available on GitHub - github.com/tgra/Linked-Art

Download the notebooks zip file
https://github.com/tgra/Linked-Art/archive/refs/heads/main.zip

# Software Installation Instructions

### Anaconda

We recommend installing Anaconda, an all-in-one installer, that will include Python and Jupyter Notebook. Please make sure you install Python version 3.x (e.g., 3.6 is fine).

Download Anaconda via https://www.anaconda.com/products/distribution

If you do not want to install Anaconda, please refer to the additional installation instructions below for how to install Python, JupyterLab and Jupyter Notebook separately.

### OpenRefine

The reconciliation exercise notebook includes an example of using OpenRefine to reconcile collection data. You do not need to install OpenRefine, however, if you'd like to install it, please find instructions on the OpenRefine website.

### Binder

If you experience problems with installation of software, you can still view the notebooks interactively using Binder, a service for viewing/interacting with Jupyter Notebooks. This service will not allow you to write files to disc.

To use Binder to open the Linked Art notebooks use the following link:
    https://mybinder.org/v2/gh/tgra/Linked-Art/HEAD

## Jupyter Notebook

For those new to Jupyter Notebooks, we recommend you watch this short 7 minute introduction:
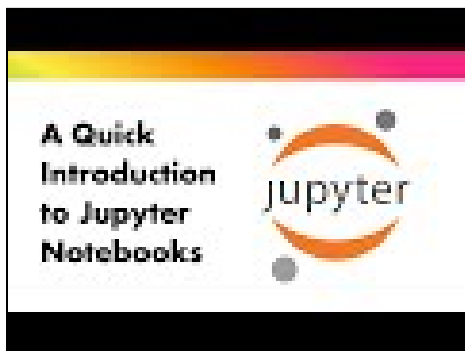


The following section introduces you to the notebook dashboard. (source)

## The Notebook dashboard

When you first start the notebook server, your browser will open to the notebook dashboard. The dashboard serves as a home page for the notebook. Its main purpose is to display the notebooks and files in the current directory. For example, here is a screenshot of the dashboard page for the examples directory in the Jupyter repository:
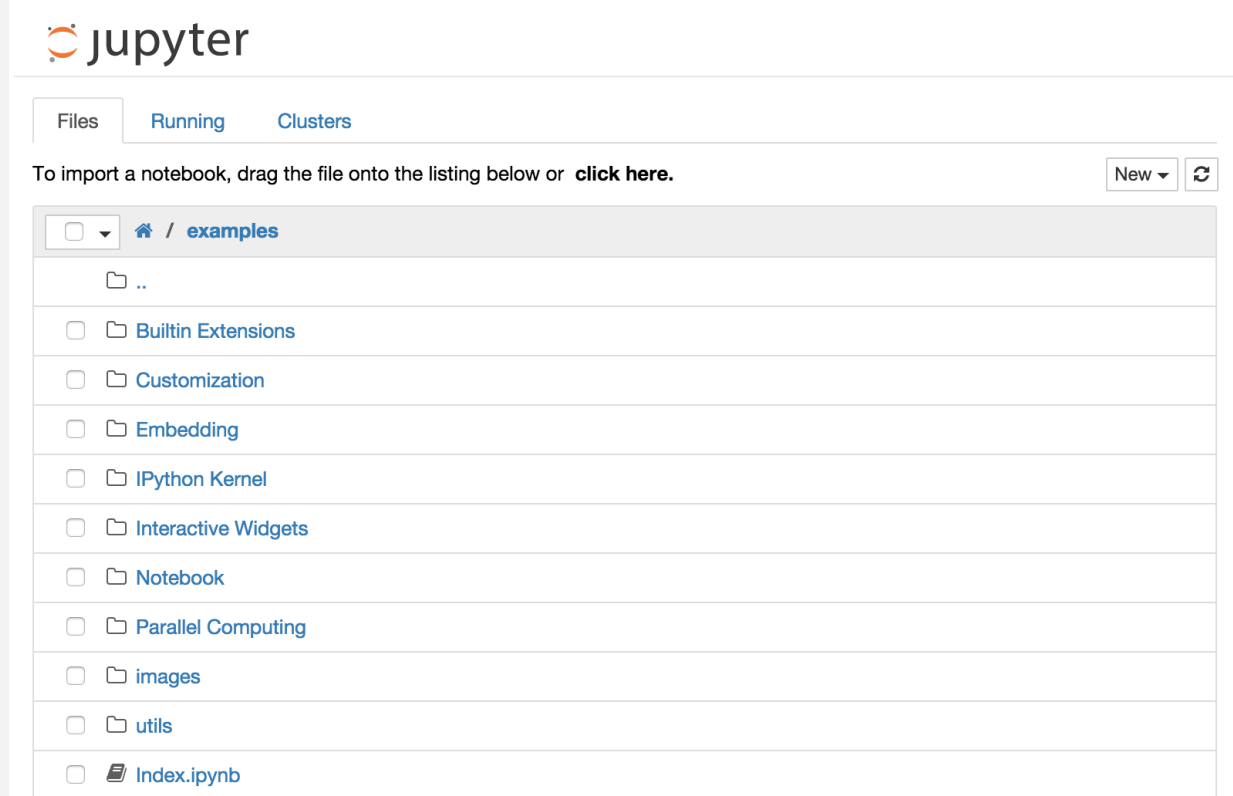


The top of the notebook list displays clickable breadcrumbs of the current directory. By clicking on these breadcrumbs or on sub-directories in the notebook list, you can navigate your file system.

To create a new notebook, click on the "New" button at the top of the list and select a kernel from the dropdown (as seen below). Which kernels are listed depend on what's installed on the server. Some of the kernels in the screenshot below may not exist as an option to you.
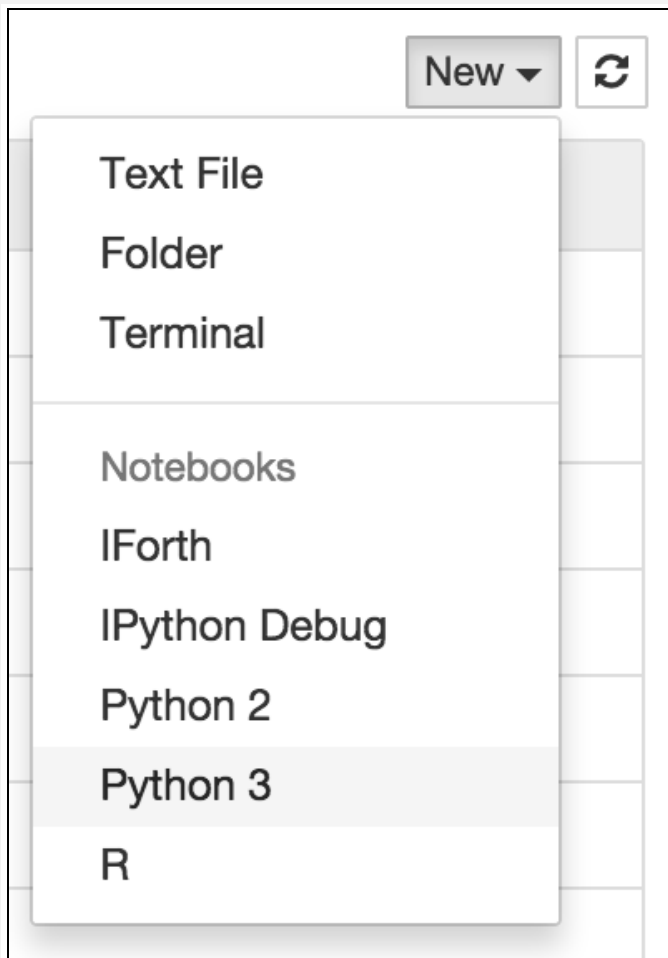
Notebooks and files can be uploaded to the current directory by dragging a notebook file onto the notebook list or by the "click here" text above the list.

The notebook list shows green "Running" text and a green notebook icon next to running notebooks (as seen below). Notebooks remain running until you explicitly shut them down; closing the notebook's page is not sufficient.
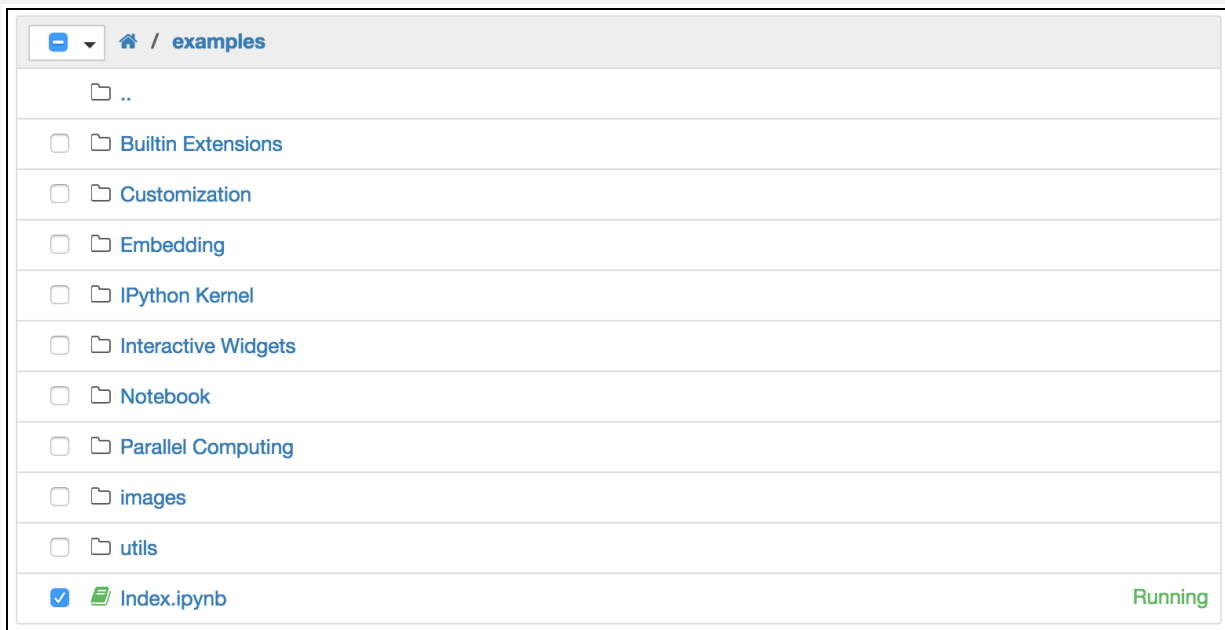
To shutdown, delete, duplicate, or rename a notebook check the checkbox next to it and an array of controls will appear at the top of the notebook list (as seen below). You can also use the same operations on directories and files when applicable.



To see all of your running notebooks along with their directories, click on the "Running" tab:

This view provides a convenient way to track notebooks that you start as you navigate the file system in a long running notebook server.

## Overview of the Notebook UI

If you create a new notebook or open an existing one, you will be taken to the notebook user interface (UI). This UI allows you to run code and author notebook documents interactively. The notebook UI has the following main areas:

- Menu
- Toolbar
- Notebook area and cells

The notebook has an interactive tour of these elements that can be started in the "Help:User Interface Tour" menu item.

## Modal editor

Starting with IPython 2.0, the Jupyter Notebook has a modal user interface. This means that the keyboard does different things depending on which mode the Notebook is in. There are two modes: edit mode and command mode.

### Edit mode

Edit mode is indicated by a green cell border and a prompt showing in the editor area:

```
In [1]: a = 10|
```

When a cell is in edit mode, you can type into the cell, like a normal text editor.

Enter edit mode by pressing `Enter` or using the mouse to click on a cell's editor area.

## Command mode

Command mode is indicated by a grey cell border with a blue left margin:

```
In [1]: a = 10
```

When you are in command mode, you are able to edit the notebook as a whole, but not type into individual cells. Most importantly, in command mode, the keyboard is mapped to a set of shortcuts that let you perform notebook and cell actions efficiently. For example, if you are in command mode and you press c, you will copy the current cell - no modifier is needed.

Don't try to type into a cell in command mode; unexpected things will happen!

Enter command mode by pressing `Esc` or using the mouse to click *outside* a cell's editor area.

## Mouse navigation

All navigation and actions in the Notebook are available using the mouse through the menubar and toolbar, which are both above the main Notebook area:

| File | Edit | View | Insert | Cell | Kernel | Help | | Python 3 O |
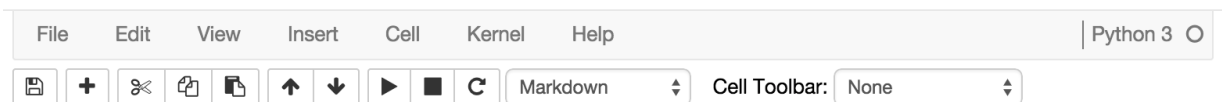
The first idea of mouse based navigation is that **cells can be selected by clicking on them.** The currently selected cell gets a grey or green border depending on whether the notebook is in edit or command mode. If you click inside a cell's editor area, you will enter edit mode. If you click on the prompt or output area of a cell you will enter command mode.

If you are running this notebook in a live session (not on http://nbviewer.jupyter.org) try selecting different cells and going between edit and command mode. Try typing into a cell.

The second idea of mouse based navigation is that **cell actions usually apply to the currently selected cell**. Thus if you want to run the code in a cell, you would select it and click the button in the toolbar or the "Cell:Run" menu item. Similarly, to copy a cell you would select it and click the button in the toolbar or the "Edit:Copy" menu item. With this simple pattern, you should be able to do most everything you need with the mouse.

Markdown cells have one other state that can be modified with the mouse. These cells can either be rendered or unrendered. When they are rendered, you will see a nice formatted representation of the cell's contents. When they are unrendered, you will see the raw text source of the cell. To render the

selected cell with the mouse, click the button in the toolbar or the "Cell:Run" menu item. To unrender the selected cell, double click on the cell.

## Keyboard Navigation

The modal user interface of the Jupyter Notebook has been optimized for efficient keyboard usage. This is made possible by having two different sets of keyboard shortcuts: one set that is active in edit mode and another in command mode.

The most important keyboard shortcuts are Enter, which enters edit mode, and Esc, which enters command mode.

In edit mode, most of the keyboard is dedicated to typing into the cell's editor. Thus, in edit mode there are relatively few shortcuts. In command mode, the entire keyboard is available for shortcuts, so there are many more. The Help->Keyboard Shortcuts dialog lists the available shortcuts.
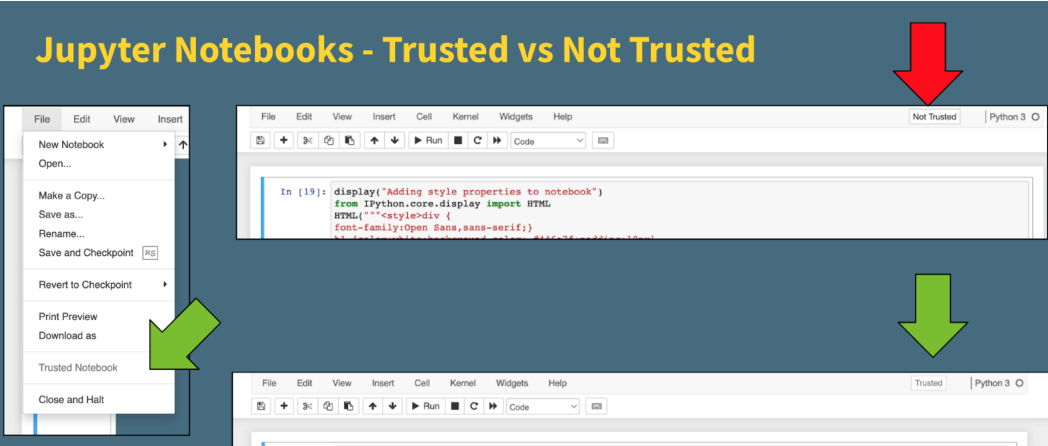
We recommend learning the command mode shortcuts in the following rough order:

1. Basic navigation: enter, shift-enter, up/k, down/j
2. Saving the notebook: s
3. Change Cell types: y, m, 1-6, t
4. Cell creation: a, b
5. Cell editing: x, c, v, d, z
6. Kernel operations: i, 0 (press twice)

# Trusted Jupyter Notebooks

When viewing a notebook, ensure that the notebook is in the 'trusted' state.

**How do I trust a notebook?**

Users can explicitly trust a notebook in two ways:

1.  At the command-line, with: jupyter trust /path/to/notebook. Ipynb
2.  After loading the untrusted notebook, with File / Trust Notebook

# Jupyter Notebook Cheatsheet

The following cheatsheet shows shortcut commands to use with the Jupyter Notebook interface.[Source](#)

# JUPYTER NOTEBOOKS CHEATSHEET

Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text.

See full documentation for Jupyter Notebooks **https://jupyter.readthedocs.io/**

## INSTALLATION

| | |
|---|---|
| Installing Anaconda Distribution installs Jupyter Notebook | https://anaconda.com/downloads |

## LAUNCH

| | |
|---|---|
| Windows | START - click Jupyter Notebook launcher icon. |
| macOS, Linux | In Terminal, type `jupyter notebook` |

## NOTEBOOK BASICS

| | |
|---|---|
| Take the 5-minute tour | http://bit.ly/jupyter-tour |

## KEYBOARD NAVIGATION

| | |
|---|---|
| Basic navigation | `ENTER` (edit mode), `ESC` (command mode) |
| Save notebook | `s` |
| Change cell types | `y, m, 1-6` |
| Cell creation | `a, b` |
| Cell editing | `x, c, v, d, z` |
| Kernel operations | `i. 0` (press twice) |
| **More Keyboard Shortcuts** | http://bit.ly/jupytershortcuts |

## INSTALL PYTHON 2 KERNEL

| | |
|---|---|
| Windows | `conda create -n ipykernel_py2 python=2 ipykernel`<br>`activate ipykernel_py2`<br>`python -m ipykernel_py2 install --user` |
| macOS, Linux | `conda create -n ipykernel_py2 python=2 ipykernel`<br>`source activate ipykernel_py2`<br>`python -m ipykernel_py2 install --user` |

## INSTALL R LANGUAGE KERNEL

| | |
|---|---|
| All operating systems | `conda install r-essentials` |

## KERNEL SHUT DOWN

| | |
|---|---|
| In the associated notebook's top navigation menu | `File - close and halt` |

## SHUT DOWN

| | |
|---|---|
| Closing the browser tab does not shut down Jupyter Notebook. To completely shut it down, close it in the terminal. | `CTRL-C` |

## MORE RESOURCES

| | |
|---|---|
| Support | anaconda.com/support |
| Training | anaconda.com/training |
| Consulting | anaconda.com/consulting |

# JupyterLab Cheatsheet

## JUPYTERLAB USER CHEATSHEET

JupyterLab is the next generation Interactive Development Environment (IDE) for data science.

See full documentation for JupyterLab **https://github.com/jupyterlab/jupyterlab**

### INSTALLATION

| | |
|---|---|
| Install Anaconda3 | https://anaconda.com/downloads |
| Install JupyterLab | |
| Windows | START - Open Anaconda Prompt，type `conda install -c conda-forge jupyterlab` |
| macOS, Linux | In terminal, type `conda install -c conda-forge jupyterlab` |
| **OPTIONAL:** Install R (all operating systems) | `conda install r-essentials` |

### LAUNCH

| | |
|---|---|
| Windows | START - select Anaconda Prompt, type `jupyter lab` |
| macOS, Linux | In Terminal, type `jupyter lab` |

### QUICK START

| | |
|---|---|
| Open Launcher | Select "+" icon in top left corner |
| Open Python 3 Notebook | Under Notebook, select `Python 3` icon |
| Type Python code in cell | In cell, type `print ("Hello World")` |
| Run Python code in cell | In ribbon, select `play` button or keyboard shortcut `shift + enter` |

### OPEN APPLICATIONS

| | |
|---|---|
| Open Notebook, Text File, Terminal, or Shell | Open Launcher by selecting "+" icon in top left corner, then select Notebook, Console, Terminal, or Text Editor |

### EDIT FILES

| | |
|---|---|
| Open a file in a text editor | From the Files tab, ctrl + click the filename to open the drop-down menu, then select Open With..., then select Editor. |

### INSTALL PYTHON 2 KERNEL

| | |
|---|---|
| Windows | `conda create -n ipykernel_py2 python=2 ipykernel`<br>`activate ipykernel_py2`<br>`python -m ipykernel_py2 install --user` |
| macOS, Linux | `conda create -n ipykernel_py2 python=2 ipykernel`<br>`source activate ipykernel_py2`<br>`python -m ipykernel_py2 install --user` |

### KERNEL SHUT DOWN

| | |
|---|---|
| Close Terminal or Kernel Sessions | Navigate to the `Running` tab on the left side of the screen, then select `Shutdown` |

### SHUT DOWN

| | |
|---|---|
| Closing browser tab does not stop JupyterLab. To completely shut it down, close it in the terminal. | `CTRL-C` |

# Linked Art

### Linked Art Workshop

Linked Art ran a 2 hour workshop at the [CIDOC 2020 conference](#) providing a wealth of background motivation and detail, a recording of which can be found below. The workshop included:

- An overview of the Linked Art initiative and community (00:00)
- A technical introduction to the Linked Art profile and overview of the core principles (from 00:14) (also available separately in the next section)
- The Van Gogh Worldwide project and its use of the Linked Art model (from 00:34) Followed by audience Q&A
- A live encoding of data from the Rijksmuseum into Linked Art (from 01:35)
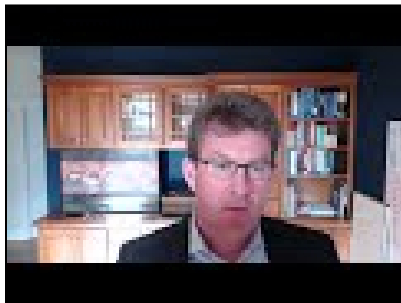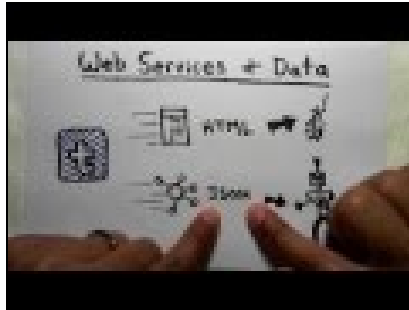


Video length is 1 hour 53 minutes.

### Linked Art Profile

A short (20 minute) technical introduction to the Linked Art profile.



# JSON-LD

JSON-LD is serialisation format for Linked Data that is used to represent descriptions of artworks as Linked Art. This video gives a quick 13 minute introduction to JSON-LD

▶ What is JSON-LD?

# Linked Art Notebook Download using Git

If you prefer, you can use the 'git clone' command or fork the repository in GitHub.

   https://github.com/tgra/Linked-Art

Further information on Git and GitHub