

# Introduction to Notebooks

[Google Cloud Datalab \(https://cloud.google.com/datalab\)](https://cloud.google.com/datalab) builds on the notebook metaphor popularized by [Jupyter \(http://jupyter.org\)](http://jupyter.org) (formerly known as IPython). If you have used notebooks, you'll find a familiar environment. If you're new to the idea of notebooks, this introduction should cover the basics to get you started.

## Notebook Basics

### Notebooks

A *notebook* is essentially a source artifact, saved as a `.ipynb` file - it can contain descriptive text content, executable code blocks, and associated results (rendered as interactive HTML). Structurally, a notebook is a sequence of cells.

### Cells

A *cell* is a block of input text that is evaluated to produce results. Cells can be of two types:

- **Code cells** - contain code to evaluate. Any outputs or results from executing the code are rendered inline after the input code.
- **Markdown cells** - contain markdown text that is converted to HTML to produce headers, lists and formatted text (like this content).

### Sessions

Each opened notebook is associated with a running *session*. If you're familiar with IPython, this is also referred to as a *kernel*. This session is used to execute all the code entered within the notebook, and manages the state (variables you've created, their values, functions and classes you've defined, and any existing Python modules you've loaded).

You can use the Reset Session command on the toolbar, above, to restart the session and start running code against a clean slate. If your notebook gets disconnected from the session, you can reconnect or recreate it by refreshing the page.

## Working in Notebooks

## Editing Cells

Cells can be individually selected, and up/down keys can be used to navigate the notebook. Once you have the appropriate cell selected, you can switch into edit mode either via the Enter key or a double click gesture.

When you are done editing the cell, you can toggle markdown cells back into rendered mode and execute code cells to produce results, either using the Run button or via the Shift+Enter key stroke. This automatically moves the selection to the next cell. Use Ctrl+Enter to stay on the current cell.

**Lets try some of this.** Following is a markdown cell, and then a code cell with Python code (and results) ready to be re-run, and an empty code cell for you to try something on your own.

[Here is a markdown cell. Go ahead... double click here to modify this text.]

```
def greet():  
    print('Hello World! Welcome to Notebooks!')  
  
greet()
```

Hello World! Welcome to Notebooks!

## Editing Notebooks

In addition to working within a cell, you can add new cells, delete cells, reorder cells or clear past results to organize your notebook content, or complete your task at hand. These editing capabilities are provided by the toolbar above, as well as via equivalent keystroke shortcuts.

Notebooks are automatically saved periodically. However you can choose to save the notebook at any point.

## Code

A notebook provides an environment to author and execute code. In Datalab, you will most often write Python code to work with your data. You've already seen a little bit of that above.

Datalab also introduces the ability to author BigQuery SQL, JavaScript, or even shell commands. This capability allows you to issue command-like instructions or use alternate languages in the same notebook. These code cells start with an escape sequence, `%%`, which instructs the session to treat the content of the cell as something other than Python. If you're familiar with IPython notebooks, you might have come across the term cell magic. Here's an example of a bash cell that lets you invoke a shell command.

```
%%bash
ls -al

total 60
drwxr-x--- 3 249442 5000 4096 May  2 18:20 .
drwxr-x--- 7 249442 5000 4096 May  2 18:19 ..
drwxr-xr-x 2 root   root  4096 May  2 18:20 .ipynb_checkpoints
-rw-r----- 1 249442 5000 8503 May  2 18:15 Introduction to Notebooks.ipynb
-rw-r----- 1 249442 5000 30582 May  2 18:15 Introduction to Python.ipynb
-rw-r----- 1 249442 5000 4000 May  2 18:15 Using Datalab - Accessing Cloud Data.ipynb
```

## User Interface

The Datalab user interface is composed of two screens:

- a notebook page (like this one), where you'll spend most of your time authoring notebook content and running code
- a notebook list page (where you started from), where you'll manage your list of notebooks and active notebook sessions.

## Toolbars

Within the notebook page, you'll find an application-level toolbar with commands for logging in and out, opening the sessions page, which shows all running kernels, as well as submitting feedback or looking up About information.

Then there is the notebook toolbar, which allows you to work with your notebook, for example, to save or rename the file, or download it and convert it another format, such as HTML. The rest of the toolbar commands allow you to modify and run the notebook.

## Sidebar Panes

The sidebar on the right contains multiple panes.

The Navigation pane allows you to navigate your notebook content easily. It provides links to all Level 1 and Level 2 headings that you have authored within markdown cells in your notebook. You can check out the navigation pane for this introduction notebook to jump between sections.

The Help tab allows you to view help content. Help content can be invoked via the Help button in the toolbar, or by invoking it from within a code cell. This will be covered in the next notebook in this series of introduction notebooks.

## Git Integration

Datalab also includes an integrated Git client called ungit, so you can view a visual diff of your changes, stage and commit them without leaving the browser. You can access this client from the git icon in the top right toolbar. You can find more information about ungit [here](https://github.com/FredrikNoren/ungit) (<https://github.com/FredrikNoren/ungit>).

