Franklin

The MBG jupyter exercise tool

Table of contents

Part I

Jupyter

2 Jupyter

> 🔥 This pages are under construction

```
lak sdflkasj dlfkj asldfj alskdj flaksdj flaksjd flkja sdlfkj alsdkjf laksdj flaksdf jlaksdjf alsdl
```

Jupyter Notebooks are an interactive computing environment that allow users to create and share documents containing live code, equations, visualizations, and narrative text. Originally developed as part of the IPython project, Jupyter (short for Julia, Python, R) now supports over 100 programming languages and has become a standard tool in data science, scientific computing, and education.

At the core of a Jupyter Notebook is a web-based interface that organizes content into "cells." These cells can contain code (typically in Python, but also in other languages via kernels), formatted text using Markdown, LaTeX for equations, and embedded multimedia elements. Users execute code in-place, and outputs such as plots or tables appear directly below the corresponding cells. This structure enables exploratory data analysis and facilitates reproducibility by interleaving code and its results with documentation.

Notebooks are stored in .ipynb files (JSON format), which preserve the code, outputs, and formatting. They can be run locally using the Jupyter server or hosted in cloud environments such as Google Colab or Binder. For scientific workflows, notebooks can integrate with tools for version control, containerization, and workflow management, making them a flexible instrument for open and reproducible research.

Despite their strengths, Jupyter Notebooks are not without limitations. Version control can be challenging due to the JSON-based format, and improper use (e.g., out-of-order execution) can compromise reproducibility. Nevertheless, their advantages in accessibility, interactivity, and communication have made them central to modern computational work.

2.1Keyboard shortcuts

2.1.1 Command Mode in Jupyter Notebooks

In Jupyter Notebooks, **Command Mode** is one of the two main interaction modes—the other being **Edit Mode**. Command Mode is active when the cell border is **blue**, indicating that keyboard commands will be interpreted as notebook-level operations rather than editing the cell's content.

Command Mode allows users to manage cells and perform structural modifications without using the mouse. This enhances efficiency, especially when working with large notebooks. Pressing Esc while in a cell activates Command Mode.

Common Keyboard Shortcuts in Command Mode 2-1

Shortcut	Description
Enter	Switch to Edit Mode in the selected cell
A	Insert a new cell **above** the current cell
B	Insert a new cell **below** the current cell
D, D	Delete the selected cell (press D twice quickly)
Z	Undo the last cell deletion
Y	Change cell type to **code**
M	Change cell type to **Markdown**
C	Copy the selected cell
X	Cut the selected cell
V	Paste cell **below**
Shift + V	Paste cell **above**
Shift + Up/Down	Extend selection to multiple cells
Ctrl + S (or Cmd + S on macOS)	Save the notebook
H	Show all keyboard shortcuts
0, 0	Restart the kernel (press 0 twice quickly)
Shift + M (in Command Mode)	**Merge** selected cells

These shortcuts make navigation and cell management significantly faster, enabling an efficient coding and documentation workflow within Jupyter Notebooks.

2.1.2 Edit Mode in Jupyter Notebooks

Edit Mode is activated when a cell's border turns **green**, allowing the user to directly modify the contents of the cell. You can enter Edit Mode by pressing Enter while a cell

is selected in Command Mode. This mode is primarily used for writing and editing code or Markdown content within cells.

Common Keyboard Shortcuts in Edit Mode 2-2

Shortcut	Description
Ctrl + Enter	Run the current cell and remain in Edit Mode
Shift + Enter	Run the current cell and move to the next cell
Alt + Enter	Run the current cell and insert a new cell below
Esc	Switch to Command Mode
Ctrl + /	Toggle comment on selected lines (code cells only)
Tab	Code completion or indent
Shift + Tab	Show tooltip/help for the object under cursor
Ctrl +]	Indent the current line or selection
Ctrl + [	Dedent the current line or selection
Ctrl + A	Select all content in the cell
Ctrl + Z	Undo the last change
Ctrl + Y	Redo the last undone change
Ctrl + Shift + -	Split the current cell at cursor position into two cells
Ctrl + S (or Cmd + S)	Save the notebook
Ctrl + Shift + -	Split the cell at the current cursor position

These shortcuts are optimized for efficient coding and content editing, significantly reducing reliance on the mouse and improving productivity within Jupyter Notebooks.

Part II

Docker

3 Jupyter

> 🔥 This pages are under construction

```
lak sdflkasj dlfkj asldfj alskdj flaksdj flaksjd flkja sdlfkj alsdkjf laksdj flaksdf jlaksdjf alsdl
```

Jupyter Notebooks are an interactive computing environment that allow users to create and share documents containing live code, equations, visualizations, and narrative text. Originally developed as part of the IPython project, Jupyter (short for Julia, Python, R) now supports over 100 programming languages and has become a standard tool in data science, scientific computing, and education.

At the core of a Jupyter Notebook is a web-based interface that organizes content into "cells." These cells can contain code (typically in Python, but also in other languages via kernels), formatted text using Markdown, LaTeX for equations, and embedded multimedia elements. Users execute code in-place, and outputs such as plots or tables appear directly below the corresponding cells. This structure enables exploratory data analysis and facilitates reproducibility by interleaving code and its results with documentation.

Notebooks are stored in .ipynb files (JSON format), which preserve the code, outputs, and formatting. They can be run locally using the Jupyter server or hosted in cloud environments such as Google Colab or Binder. For scientific workflows, notebooks can integrate with tools for version control, containerization, and workflow management, making them a flexible instrument for open and reproducible research.

Despite their strengths, Jupyter Notebooks are not without limitations. Version control can be challenging due to the JSON-based format, and improper use (e.g., out-of-order execution) can compromise reproducibility. Nevertheless, their advantages in accessibility, interactivity, and communication have made them central to modern computational work.

3.1 Keyboard shortcuts

3.1.1 Command Mode in Jupyter Notebooks

In Jupyter Notebooks, **Command Mode** is one of the two main interaction modes—
the other being **Edit Mode**. Command Mode is active when the cell border is **blue**,
indicating that keyboard commands will be interpreted as notebook-level operations
rather than editing the cell's content.

Command Mode allows users to manage cells and perform structural modifications
without using the mouse. This enhances efficiency, especially when working with large
notebooks. Pressing Esc while in a cell activates Command Mode.

Common Keyboard Shortcuts in Command Mode 3-1

Shortcut	Description
Enter	Switch to Edit Mode in the selected cell
A	Insert a new cell **above** the current cell
B	Insert a new cell **below** the current cell
D, D	Delete the selected cell (press D twice quickly)
Z	Undo the last cell deletion
Y	Change cell type to **code**
M	Change cell type to **Markdown**
C	Copy the selected cell
X	Cut the selected cell
V	Paste cell **below**
Shift + V	Paste cell **above**
Shift + Up/Down	Extend selection to multiple cells
Ctrl + S (or Cmd + S on macOS)	Save the notebook
H	Show all keyboard shortcuts
0, 0	Restart the kernel (press 0 twice quickly)
Shift + M (in Command Mode)	**Merge** selected cells

These shortcuts make navigation and cell management significantly faster, enabling an
efficient coding and documentation workflow within Jupyter Notebooks.

3.1.2 Edit Mode in Jupyter Notebooks

Edit Mode is activated when a cell's border turns **green**, allowing the user to directly
modify the contents of the cell. You can enter Edit Mode by pressing Enter while a cell

is selected in Command Mode. This mode is primarily used for writing and editing code or Markdown content within cells.

Common Keyboard Shortcuts in Edit Mode 3-2

Shortcut	Description
Ctrl + Enter	Run the current cell and remain in Edit Mode
Shift + Enter	Run the current cell and move to the next cell
Alt + Enter	Run the current cell and insert a new cell below
Esc	Switch to Command Mode
Ctrl + /	Toggle comment on selected lines (code cells only)
Tab	Code completion or indent
Shift + Tab	Show tooltip/help for the object under cursor
Ctrl +]	Indent the current line or selection
Ctrl + [	Dedent the current line or selection
Ctrl + A	Select all content in the cell
Ctrl + Z	Undo the last change
Ctrl + Y	Redo the last undone change
Ctrl + Shift + -	Split the current cell at cursor position into two cells
Ctrl + S (or Cmd + S)	Save the notebook
Ctrl + Shift + -	Split the cell at the current cursor position

These shortcuts are optimized for efficient coding and content editing, significantly reducing reliance on the mouse and improving productivity within Jupyter Notebooks.