Jupyter Notebook Notebook shortcuts

START HERE

Shift + **Enter**
Run cell, select next

B
Insert cell below

Esc
Command mode

D › **D**
Delete cell

Enter
Edit mode

M
Markdown cell

A
Insert cell above

Ctrl + **Shift** + **C**
Command palette

Command mode keys act on cells → **Enter** → **Edit mode** keys type into a cell → **Esc** → **Command mode** back to cells

CONCEPTS

Esc leaves a cell (blue bar, command mode); Enter goes back in (green bar, edit mode). Single letters only work in command mode.

One kernel runs all cells in order of execution, not of position; the number in [ ] shows that order.

Shortcuts are for Notebook 7 on Linux/Windows. On macOS read Ctrl as Cmd where the Accel key is used (run, select all, palette).

Classic Notebook 6 and JupyterLab differ in places: Notebook 6 used Ctrl+Shift+P for the palette and I I / 0 0 for interrupt and restart.

MODES & RUNNING

Shift + **Enter**
Run cell and select next

Ctrl + **Enter** Run cell and stay

Alt + **Enter**
Run cell and insert below

Esc / **Ctrl** + **M**
Enter command mode

Enter Enter edit mode

Shift + **R**
Toggle side-by-side render

NAVIGATE & SELECT

Up / **K** Select cell above

Down / **J** Select cell below

Shift + **Up** / **Shift** + **K**
Extend selection up

Shift + **Down** / **Shift** + **J**
Extend selection down

Shift + **Home**
Extend selection to top

Shift + **End**
Extend selection to bottom

Ctrl + **A** Select all cells
Command mode; Cmd+A on macOS.

Left
Collapse heading / go to parent

NAVIGATE & SELECT (cont.)

Right Expand heading

Ctrl + **Shift** + **Left**
Collapse all headings

Ctrl + **Shift** + **Right**
Expand all headings

CELLS

A Insert cell above

B Insert cell below

D › **D** Delete cell

Z Undo cell action

Shift + **Z** Redo cell action

C Copy cell

X Cut cell

V Paste cell below

Shift + **M**
Merge selected cells

Ctrl + **Backspace**
Merge with cell above

Ctrl + **Shift** + **M**
Merge with cell below

Ctrl + **Shift** + **-**
Split cell at cursor
Edit mode.

Ctrl + **Shift** + **Up**
Move cell up

Ctrl + **Shift** + **Down**
Move cell down

O Toggle cell output

Shift + **L** Toggle line numbers

Shift + **A**
Insert heading above

Shift + **B**
Insert heading below

CELL TYPES

M Change to Markdown

Y Change to code

R Change to raw
Raw cells pass through nbconvert untouched.

1-6 Change to heading 1-6
Command mode, Markdown cell.

CELL TYPES (cont.)

Alt + **Up**
Previous history entry
Edit mode.

Alt + **Down** Next history entry
Edit mode.

KERNEL & PALETTE

Ctrl + **Shift** + **C**
Open command palette
Type a command name: restart kernel, run all, clear outputs, toggle theme.

Tab Complete name
Edit mode.

Kernel menu (no default keys in Notebook 7)

Menu item	Effect
Interrupt	Stop a running cell
Restart	Fresh process, state lost
Restart and Run All	Clean top-to-bottom check
Change Kernel	Switch environment

Restart and Run All before sharing: it proves the notebook works top to bottom.

MAGICS

`%timeit <expr>`
Time a statement

`%%time`
Time the whole cell

`%%bash`
Run cell as a bash script

`!<cmd>`
Run a shell command

`%matplotlib inline`
Plots inside the notebook

`%load_ext autoreload`
Enable module reloading
Follow with %autoreload 2 to reload edited modules before each run.

`%who`
List variables

`%run <file.py>`
Run a script in this kernel

`%pip install <pkg>`
Install into the running kernel
Prefer over !pip: it targets the kernel's environment.

MAGICS (cont.)

`%lsmagic`
List all magics

`<name>?`
Show help for an object
Add ?? for source code.

CLI

`jupyter notebook`
Start the Notebook server

`jupyter lab`
Start JupyterLab

`jupyter nbconvert --to <fmt> <nb>`
Convert to html, pdf or script

`jupyter nbconvert --execute --to notebook <nb>`
Run and save outputs

`jupyter kernelspec list`
List installed kernels

`python -m ipykernel install --user --name <env>`
Register an environment as kernel

`jupyter kernelspec remove <name>`
Remove a kernel

`jupyter notebook list`
Show running servers and URLs

`jupyter --paths`
Show config and data paths

`jupyter notebook --no-browser --port <n>`
Headless on a chosen port

RECIPES

Use a venv in Jupyter

```
python -m venv .venv
source .venv/bin/activate
pip install ipykernel
python -m ipykernel install --
user --name myenv
```

Notebook to clean script

```
jupyter nbconvert --to script
analysis.ipynb
# writes analysis.py
```

Run remote server over SSH

```
# on the server
jupyter notebook --no-browser --
port 8888
# on your laptop
ssh -L 8888:localhost:8888
<user>@<host>
```

Reload edited modules

```
%load_ext autoreload
%autoreload 2
```