Jupyter Notebook Notebook shortcuts

CONCEPTS

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

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

> 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).

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

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

Ctrl + A Select all cells

Right Expand heading

CELLS

A Insert cell above

B Insert cell below

D › D Delete cell

CELLS (cont.)

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 + Shift + -
Split cell at cursor

Ctrl + Shift + Up Move cell up

Ctrl + Shift + Down
Move cell down

O Toggle cell output

Shift + L Toggle line numbers

CELL TYPES

M Change to Markdown

Y Change to code

R Change to raw

1-6 Change to heading 1-6

KERNEL & PALETTE

Ctrl + Shift + C
Open command palette

Tab Complete name

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

MAGICS (cont.)

%%bash
Run cell as a bash script

!*<cmd>*
Run a shell command

%matplotlib inline
Plots inside the notebook

%load_ext autoreload
Enable module reloading

%who
List variables

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

%pip install *<pkg>*
Install into the running kernel

*<name>***?**
Show help for an object

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 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
```

Run remote server over SSH

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