

- Credits and Attribution

Jupyter Notebook

Credits and Attribution



- Python is an open-source language. Tools and packages used here are all open-source.
- This module adapted part of its contents from open-source and public domain sources, which include but not limited to:
 - Python for Data Analysis Book - Wes McKinney
 - Whirlwind Tour of Python - Jake VanderPlas
 - HDB Resale Price – data.gov.sg
- Specific URLs are provided in Jupyter Notebook to acknowledge the sources and the authors.

Licenses and Copyright belongs to the respective authors and developers. More information about the licenses and copyright are available within the code.

Jupyter Notebook

- Web-based
- Create and Share Codes
- Functionality:
 - Live Code
 - Live Visualisation
 - Explanatory Text/Comments in Markdown
- Usage:
 - Learn and Try Code (Python, R etc)
 - Data Processing/Transformation
 - Run, See, and Debug instantly.

Jupyter Notebook: Installation and Run

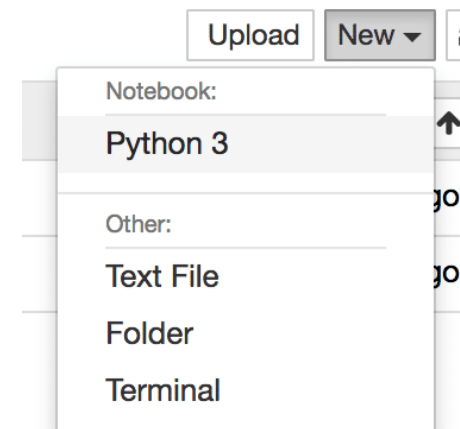
- Install:

- Use the Python's package manager *pip*
- Install the Anaconda distribution

- Run:

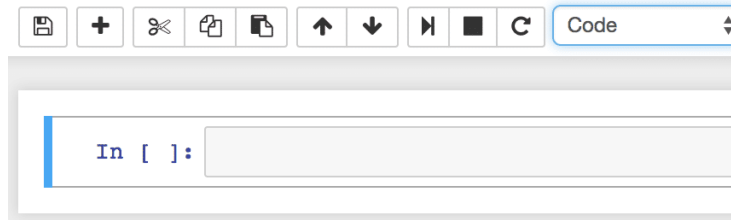
- Jupyter notebook (command prompt)
- Find icon on Windows

- Creating A New Notebook

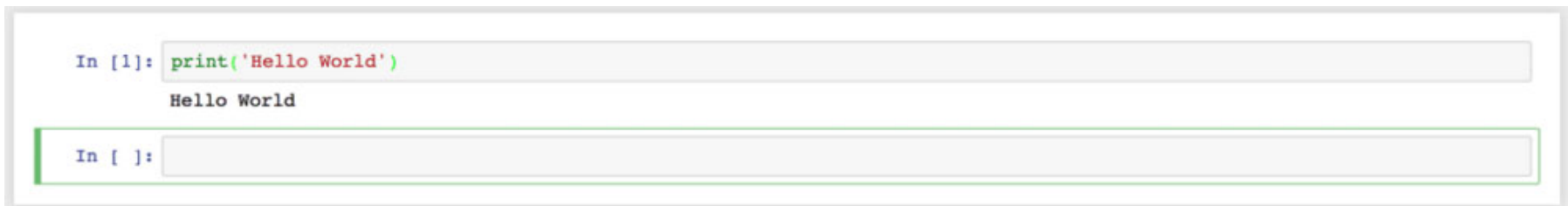


To Code with the Notebook

1. The notebook has cells. An empty cell is created:



2. Start typing in Python code
3. Execute by either clicking on the *run cell* button or hitting Shift + Return keys:

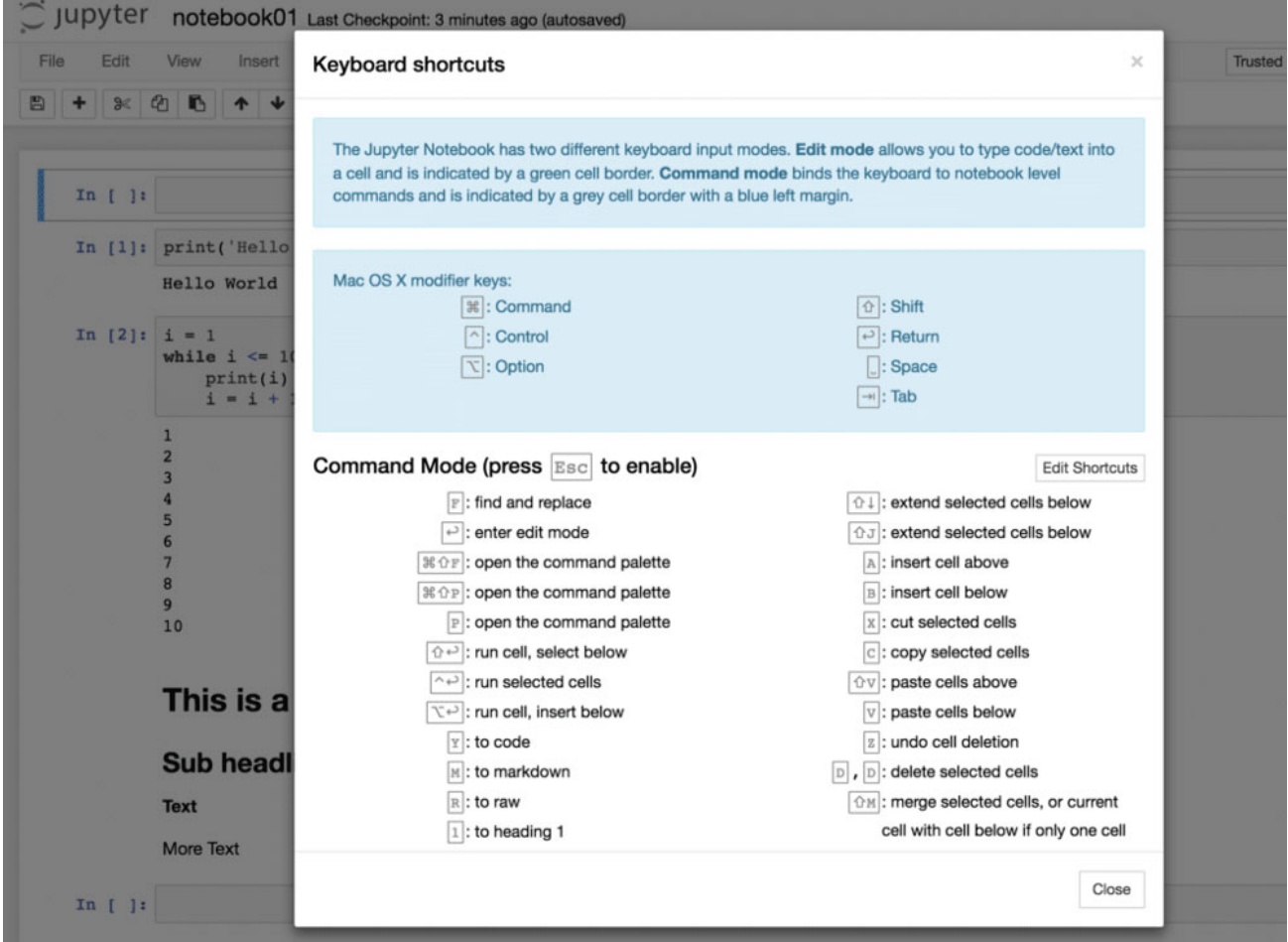


4. The resulting output is shown below the cell.

Shortcuts

Overview of key shortcuts:

- using
menu
entry
- *Help* →
*Keyboard
Shortcuts*



The screenshot shows a Jupyter Notebook window titled 'jupyter notebook01' with a 'Last Checkpoint: 3 minutes ago (autosaved)' status. The notebook contains three code cells. The first cell is empty. The second cell contains `print('Hello World')` and has been executed, showing the output 'Hello World'. The third cell contains a loop: `i = 1; while i <= 10: print(i); i = i + 1`, with line numbers 1 through 10 visible on the left. A 'Keyboard shortcuts' dialog box is open in the foreground. It explains that Jupyter has two input modes: Edit mode (green border) and Command mode (grey border with blue margin). It lists Mac OS X modifier keys: Command (⌘), Control (⌃), Option (⌥), Shift (⇧), Return (↵), Space (␣), and Tab (⇥). The 'Command Mode (press Esc to enable)' section lists various shortcuts for navigation, editing, and cell management, such as find and replace (⌘F), enter edit mode (⇧⌘P), open the command palette (⇧⌘P), run cell (⇧⌘↵), and insert/delete cells (⌘A, ⌘B, ⌘X, ⌘C, ⌘V, ⌘Z, ⌘D, ⌘M).

Keyboard shortcuts

The Jupyter Notebook has two different keyboard input modes. **Edit mode** allows you to type code/text into a cell and is indicated by a green cell border. **Command mode** binds the keyboard to notebook level commands and is indicated by a grey cell border with a blue left margin.

Mac OS X modifier keys:

- ⌘: Command
- ⌃: Control
- ⌥: Option
- ⇧: Shift
- ↵: Return
- ␣: Space
- ⇥: Tab

Command Mode (press `Esc` to enable)

`⌘F`: find and replace
`⇧⌘P`: enter edit mode
`⇧⌘P`: open the command palette
`⇧⌘P`: open the command palette
`⇧⌘P`: open the command palette
`⇧⌘↵`: run cell, select below
`⌘↵`: run selected cells
`⇧⌘↵`: run cell, insert below
`Y`: to code
`M`: to markdown
`R`: to raw
`I`: to heading 1

`⇧⌘↓`: extend selected cells below
`⇧⌘↑`: extend selected cells above
`A`: insert cell above
`B`: insert cell below
`X`: cut selected cells
`C`: copy selected cells
`V`: paste cells above
`V`: paste cells below
`Z`: undo cell deletion
`D, D`: delete selected cells
`⇧⌘M`: merge selected cells, or current cell with cell below if only one cell

[Edit Shortcuts](#)

[Close](#)

When reading the coursebook...

- Run each line and observe the
 - In:
 - Out:
- Understand the purpose of the line
- Experiment with each line

Please feel free to find us if you have any questions!