

COMP10020

Introduction to Programming II

Dr. Brian Mac Namee

brian.macnamee@ucd.ie

School of Computer Science
University College Dublin

JUPYTER & IPYTHON NOTEBOOKS

Python As A Collection Of Libraries

Python has a large **standard library** that includes support for GUIs, database connections, networking, handling regular expressions, ...

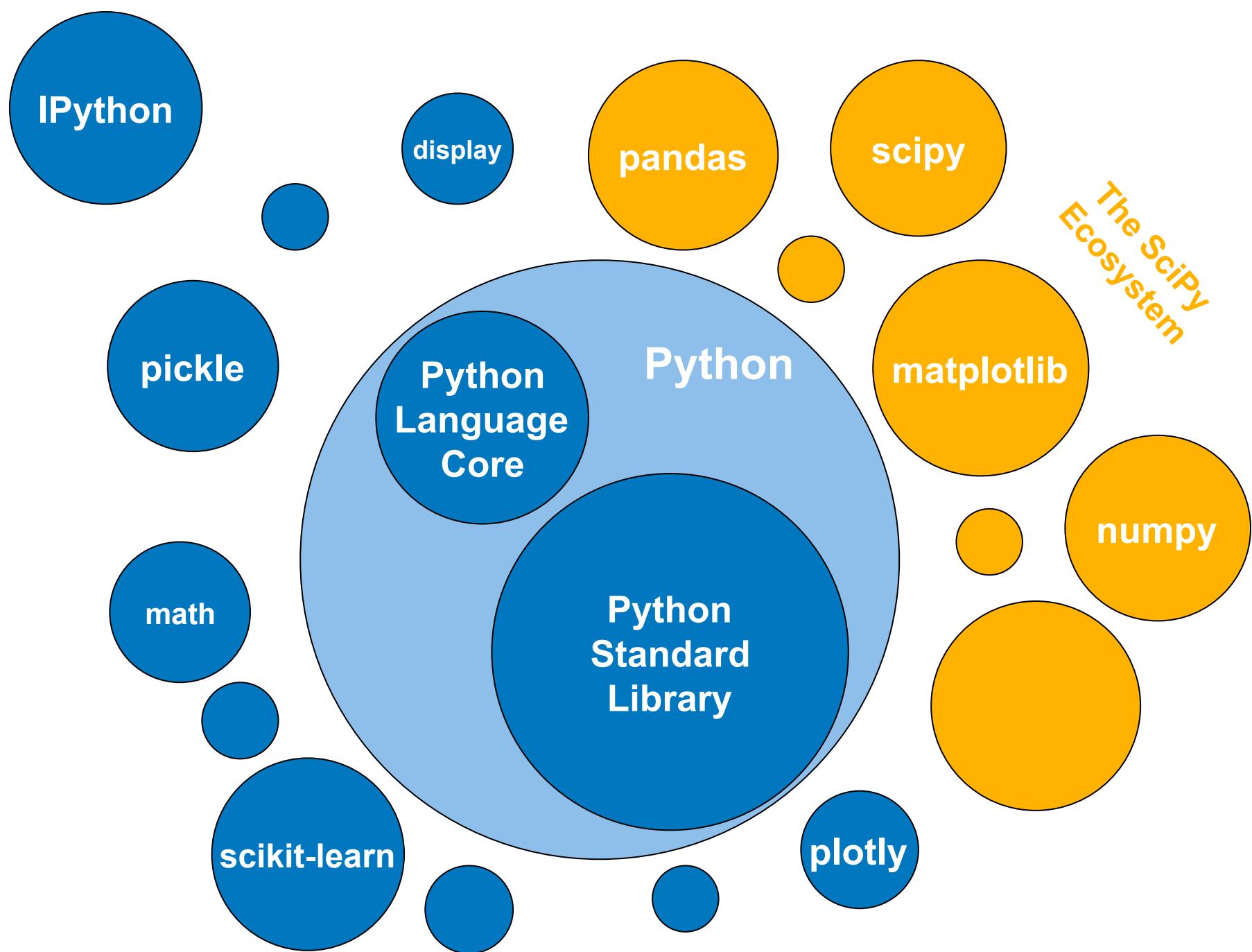
One of the biggest attractions of Python is the fact that it is easily extended through the development of **libraries (or packages)**

Libraries can be written by anyone but are available centrally through installers such as **pip**

Just about everything you can imagine is implemented in some Python package!

- Currently at pypi.python.org there are 82,737 packages listed!

The SciPy Ecosystem



Jupyter

The Jupyter project (www.jupyter.org) is a web application for interactive data science and scientific computing

Jupyter allows us to create **notebooks** that contain text, code, and output all in one place – ideal for data science projects



A screenshot of a Jupyter Notebook interface. The title bar says "IPy IPython Dashboard" and "IP[y]: Notebook". The main area shows a section titled "Simple spectral analysis" with the subtitle "An illustration of the Discrete Fourier Transform". It contains a mathematical formula for the DFT:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \quad k = 0, \dots, N - 1$$

Text explaining windowing for frequency content is present. A code cell [1] shows:

```
In [1]: from scipy.io import wavfile  
rate, x = wavfile.read('test_mono.wav')
```

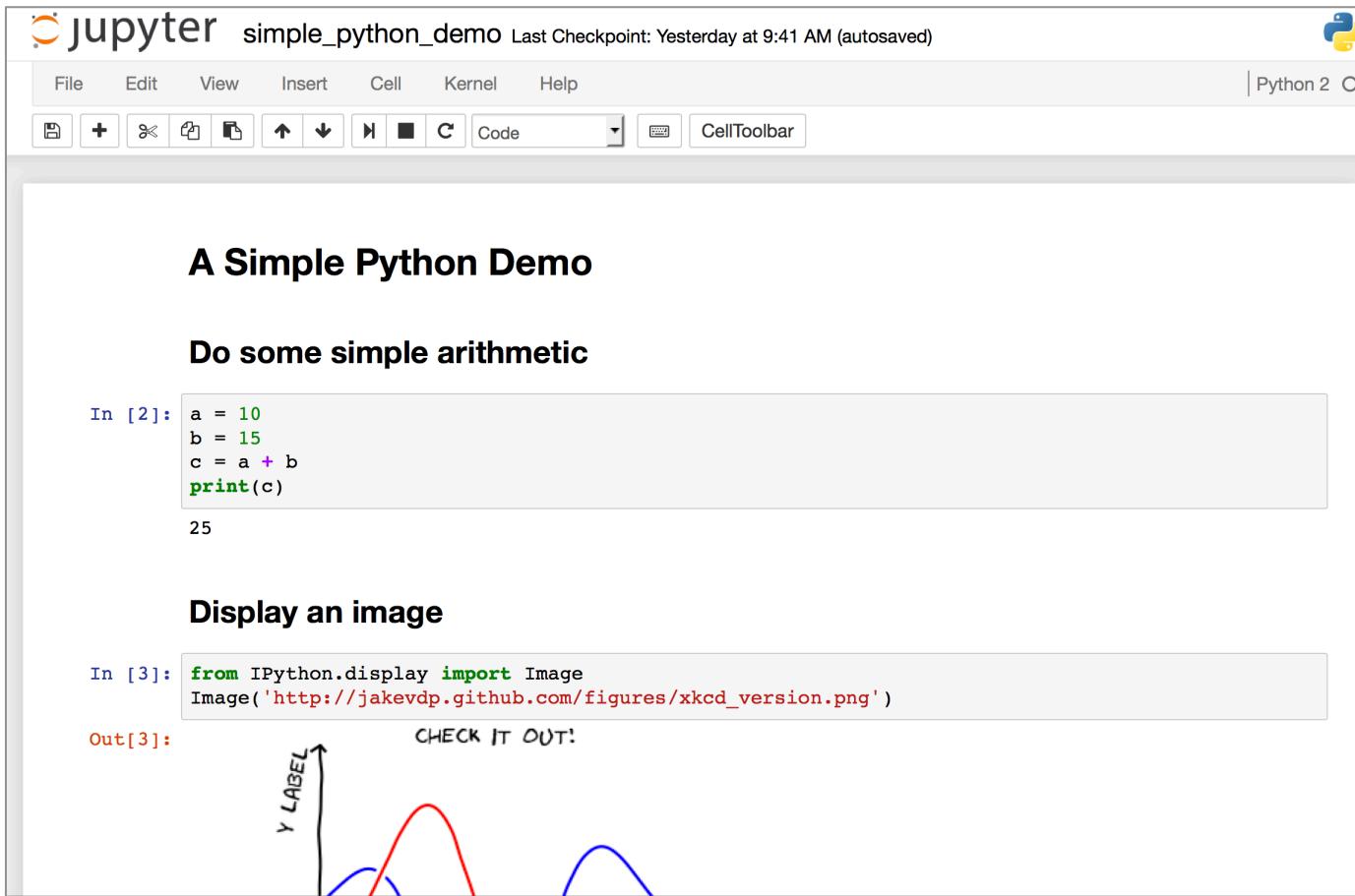
Text explaining the use of matplotlib's specgram routine follows. A code cell [2] shows:

```
In [2]: fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))  
ax1.plot(x); ax1.set_title('Raw audio signal')  
ax2.specgram(x); ax2.set_title('Spectrogram');
```

Two plots are shown: "Raw audio signal" (a blue waveform plot) and "Spectrogram" (a heatmap plot showing frequency over time).

Jupyter

IPython Notebooks provide an engine for running Python code under Jupyter



The screenshot shows a Jupyter Notebook window titled "jupyter simple_python_demo". The toolbar includes File, Edit, View, Insert, Cell, Kernel, Help, and a Python 2 kernel icon. Below the toolbar is a toolbar with various icons for cell operations like run, cell type, and cell toolbar.

The notebook content starts with a title "A Simple Python Demo" and a section "Do some simple arithmetic". In cell [2], the following Python code is run:

```
In [2]:  
a = 10  
b = 15  
c = a + b  
print(c)
```

The output of this cell is "25".

Next, there is a section "Display an image". In cell [3], the following code is run:

```
In [3]:  
from IPython.display import Image  
Image('http://jakevdp.github.com/figures/xkcd_version.png')
```

The output of this cell is an image of a comic strip from xkcd. The comic features two overlapping bell curves, one red and one blue, on a graph with a y-axis labeled "Y LABEL". The text "CHECK IT OUT!" is written above the curves.

File Edit View Insert Cell Kernel Help

Python 2



A Simple Python Demo

Do some simple arithmetic

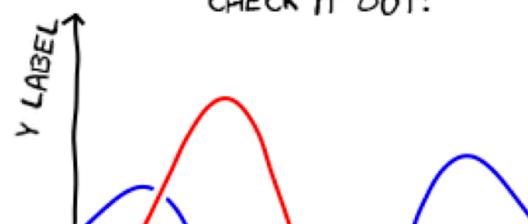
```
In [2]: a = 10  
       b = 15  
       c = a + b  
       print(c)
```

25

Display an image

```
In [3]: from IPython.display import Image  
Image('http://jakevdp.github.com/figures/xkcd_version.png')
```

Out[3]:



File Edit View Insert Cell Kernel Help

Python 2



A Simple Python Demo

Do some simple arithmetic

Code Cell

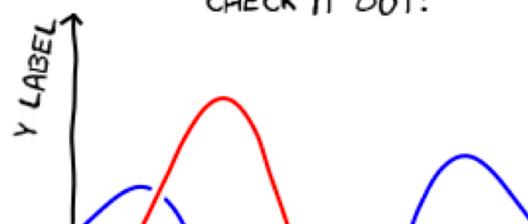
```
In [2]: a = 10  
       b = 15  
       c = a + b  
       print(c)
```

25

Display an image

```
In [3]: from IPython.display import Image  
Image('http://jakevdp.github.com/figures/xkcd_version.png')
```

Out[3]:



File Edit View Insert Cell Kernel Help

Python 2



A Simple Python Demo

Do some simple arithmetic

```
In [2]: a = 10  
       b = 15  
       c = a + b  
       print(c)
```

```
25
```

Code Output

Display an image

```
In [3]: from IPython.display import Image  
Image('http://jakevdp.github.com/figures/xkcd_version.png')
```

```
Out[3]:
```



File Edit View Insert Cell Kernel Help

Python 2



CellToolbar

Markdown Cell

A Simple Python Demo

Do some simple arithmetic

```
In [2]: a = 10  
       b = 15  
       c = a + b  
       print(c)
```

25

Display an image

```
In [3]: from IPython.display import Image  
Image('http://jakevdp.github.com/figures/xkcd_version.png')
```

Out[3]:



iPython Notebooks

Entire books have been written as iPython notebooks

The screenshot shows a Jupyter notebook interface. At the top left is the "jupyter nbviewer" logo. At the top right are links for "JUPYTER", "FAQ", and download icons. The main content area has a header bar with the path "Probabilistic-Programming-and-Bayesian-Methods-for-Hackers / Chapter1_Introduction". Below this is the title "Probabilistic Programming and Bayesian Methods for Hackers" in large bold letters. A welcome message follows: "Welcome to *Bayesian Methods for Hackers*. The full Github repository is available at [github/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers](#). The other chapters can be found on the project's [homepage](#). We hope you enjoy the book, and we encourage any contributions!" There is also a section titled "Looking for a printed version of Bayesian Methods for Hackers?" with a link to the book's page on Amazon. At the bottom is a thumbnail image of the book cover, which features abstract geometric shapes in green, blue, and black against a dark background, with the Addison Wesley Data & Analytics Series logo.

http://nbviewer.jupyter.org/github/CamDavidsonPilon/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers/blob/master/Chapter1_Introduction/Chapter1.ipynb

INSTALLING & STARTING IPYTHON NOTEBOOKS

Downloading iPython Notebooks

The easiest way to set up iPython notebooks is to download Anaconda from Continuum Analytics

The screenshot shows the Continuum Analytics website. At the top, there's a navigation bar with links for Log In, Support, Search, and Contact, followed by ANACONDA, COMMUNITY, CONSULTING, TRAINING, ABOUT, and RESOURCES. On the left, there's a large green "DOWNLOAD ANACONDA NOW" button with sub-links for "Download For" Windows, Mac, and Linux. To the right is a large green circular logo with a grid pattern, surrounded by small colored triangles (green, blue, and yellow). Below the download section is a purple banner with text about AnacondaCON 2017 and a "REGISTER NOW" button. At the bottom left, there's a "GET SUPERPOWERS WITH ANACONDA" section and a note about Anaconda being the leading open data science platform. On the right, a box contains a question about which Python version to download.

ANACONDA
Powered by Continuum Analytics

Log In Support Search Contact

ANACONDA COMMUNITY CONSULTING TRAINING ABOUT RESOURCES

**DOWNLOAD
ANACONDA NOW**

Download For

Supercharge your Data Science team with two-for-one tickets to
AnacondaCON 2017!

REGISTER NOW

**GET SUPERPOWERS
WITH ANACONDA**

Anaconda is the leading open data science platform powered by Python. The

Which version should I download and install?
With Anaconda you can run multiple versions of Python in isolated environments, so choose the download with the Python version that you use

<https://www.continuum.io/downloads>

Downloading iPython Notebooks

Install Anaconda with Python3.5

The screenshot shows the Anaconda download page for Windows. At the top, there are three download links: "Download for Windows" (selected), "Download for OSX", and "Download for Linux". Below this, the heading "Anaconda 4.2.0" is followed by "For Windows". A note states: "Anaconda is BSD licensed which gives you permission to use Anaconda commercially and for redistribution." A "Changelog" link is provided. Three steps for installation are listed: 1. Download the installer, 2. Optional: Verify data integrity with MD5 or SHA-256 (with a "More info" link), and 3. Double-click the .exe file to install Anaconda and follow the instructions on the screen. A note for users behind a firewall suggests using zipped Windows installers. A message at the bottom encourages using the Anaconda installer archive for older versions and contacting support for long-term support.

Download for Windows Download for OSX Download for Linux

Anaconda 4.2.0
For Windows

Anaconda is BSD licensed which gives you permission to use Anaconda commercially and for redistribution.

[Changelog](#)

1. Download the installer
2. Optional: Verify data integrity with [MD5](#) or [SHA-256](#) [More info](#)
3. Double-click the .exe file to install Anaconda and follow the instructions on the screen

Behind a firewall? Use these [zipped Windows installers](#)

For older versions of Anaconda installers, see the [Anaconda installer archive](#)
For long-term support of the packages found in the Anaconda archives, please [contact us](#).

Python 3.5 version

64-BIT INSTALLER (391M)

32-BIT INSTALLER (333M)

Python 2.7 version

64-BIT INSTALLER (381M)

32-BIT INSTALLER (324M)

GET STARTED

Downloading iPython Notebooks

Install Anaconda with Python3.5

The screenshot shows the Anaconda 4.2.0 download page for OSX. At the top, there are three download links: "Download for Windows", "Download for OSX" (which is selected), and "Download for Linux". Below these, the "Anaconda 4.2.0" section is displayed. This section includes a "For OSX" heading, a brief description about the BSD license, and links for "Changelog" and "Graphical Installer". The "Graphical Installer" link leads to the current page. The "Python 3.5 version" section is highlighted with an orange box. It contains two download options: a green "GRAPHICAL INSTALLER (407M)" button and a white "COMMAND-LINE INSTALLER (349M)" button. Both buttons have a thin green border. Below the buttons, the text "64-Bit" is centered. To the right of the highlighted section, there is another section for "Python 2.7 version" with its own "GRAPHICAL INSTALLER (403M)" and "COMMAND-LINE INSTALLER (346M)" buttons, also labeled "64-Bit". A small circular arrow icon with an upward-pointing arrow is located on the right side of the page. At the bottom left, there is a URL: "s://www.continuum.io/downloads#osx".

Download For Windows Download for OSX Download for Linux

Anaconda 4.2.0

For OSX

Anaconda is BSD licensed which gives you permission to use Anaconda commercially and for redistribution.

[Changelog](#)

Graphical Installer

1. Download the graphical installer
2. Double-click the downloaded **.pkg** file and follow the instructions

Command Line Installer

1. Download the command-line installer
2. Optional: Verify data integrity with [MD5](#) or [SHA-256](#) [More info](#)
3. In your terminal window type one of the below and follow the instructions:
Python 3.5 version

```
bash Anaconda3-4.2.0-MacOSX-x86_64.sh
```

Python 2.7 version

```
bash Anaconda2-4.2.0-MacOSX-x86_64.sh
```

s://www.continuum.io/downloads#osx

Python 3.5 version

GRAPHICAL INSTALLER (407M)

COMMAND-LINE INSTALLER (349M)

64-Bit

Python 2.7 version

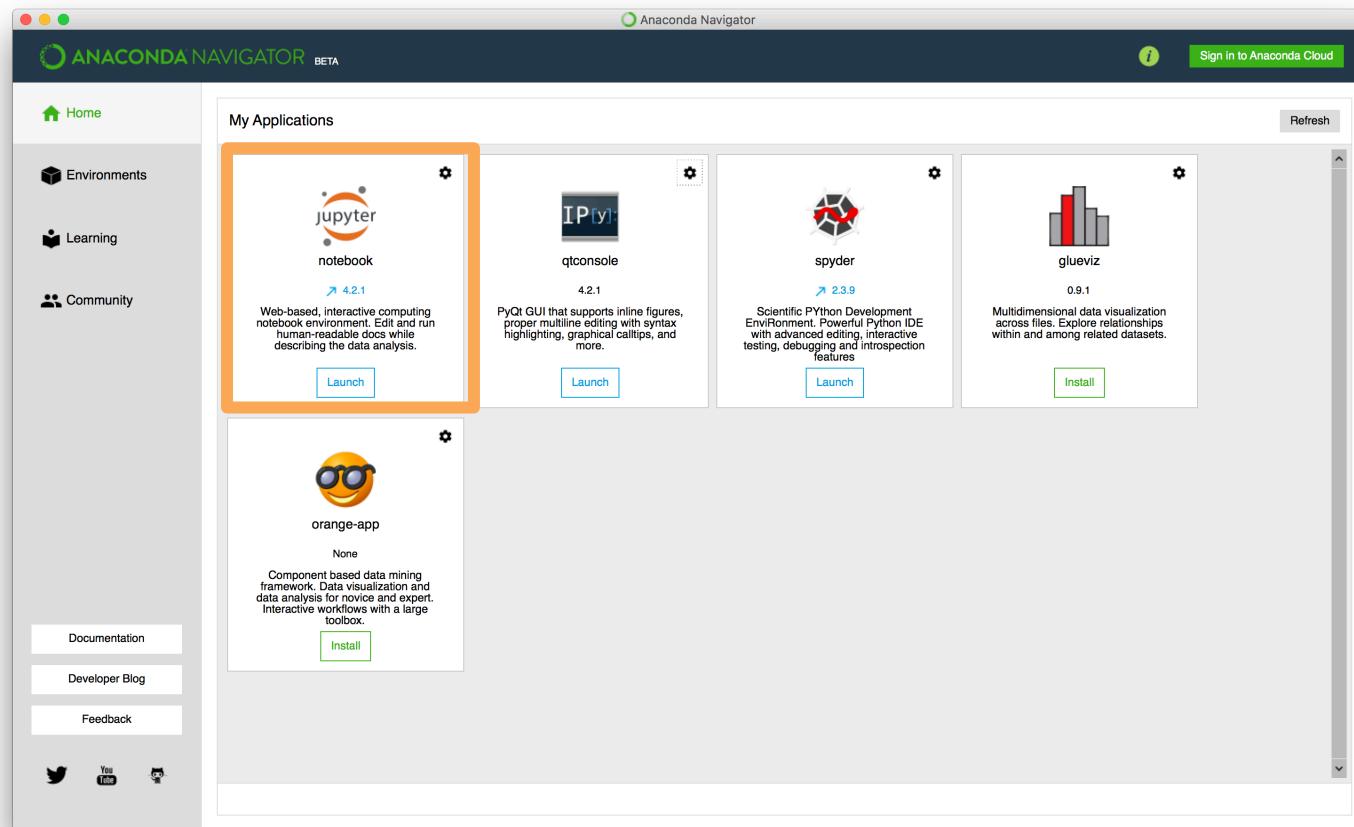
GRAPHICAL INSTALLER (403M)

COMMAND-LINE INSTALLER (346M)

64-Bit

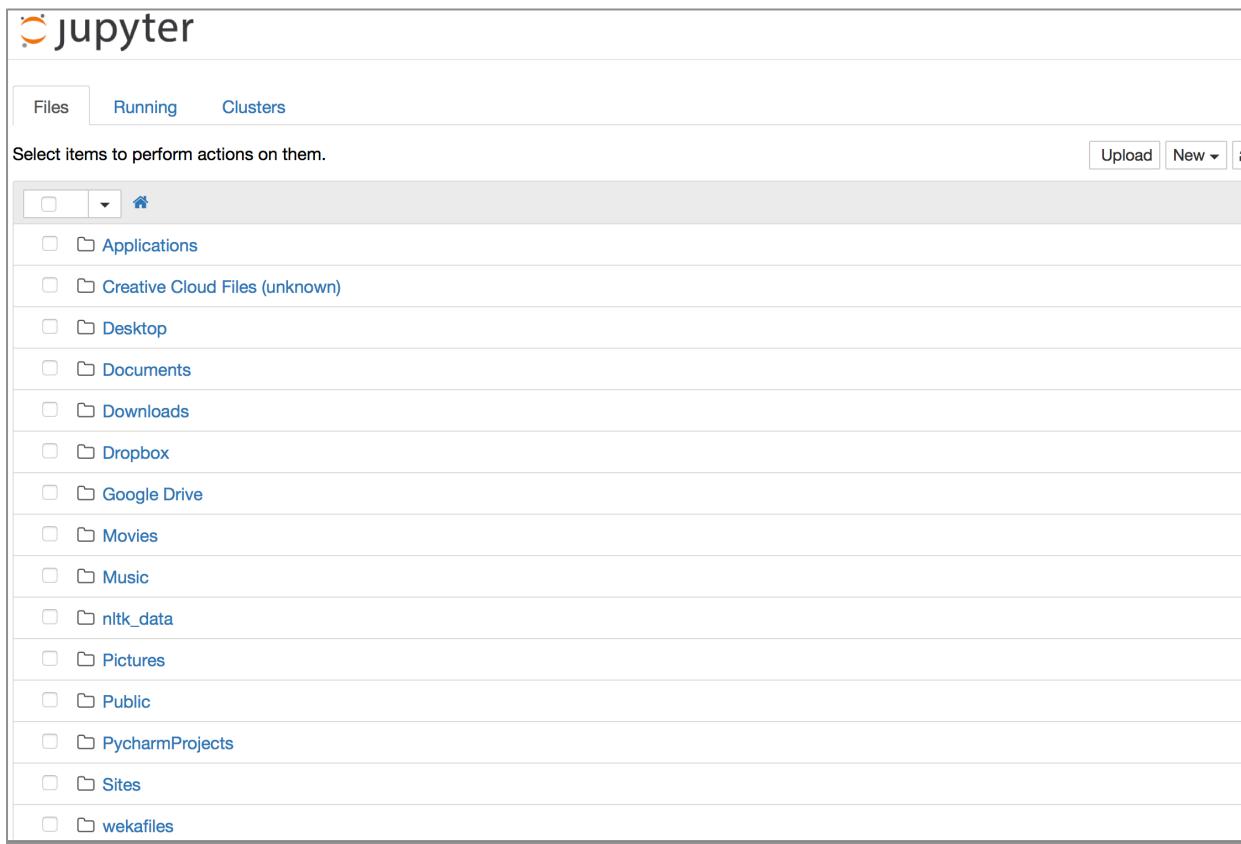
Starting iPython Notebooks

From wherever you installed iPython notebooks
start the Navigator application and launch jupyter
notebook



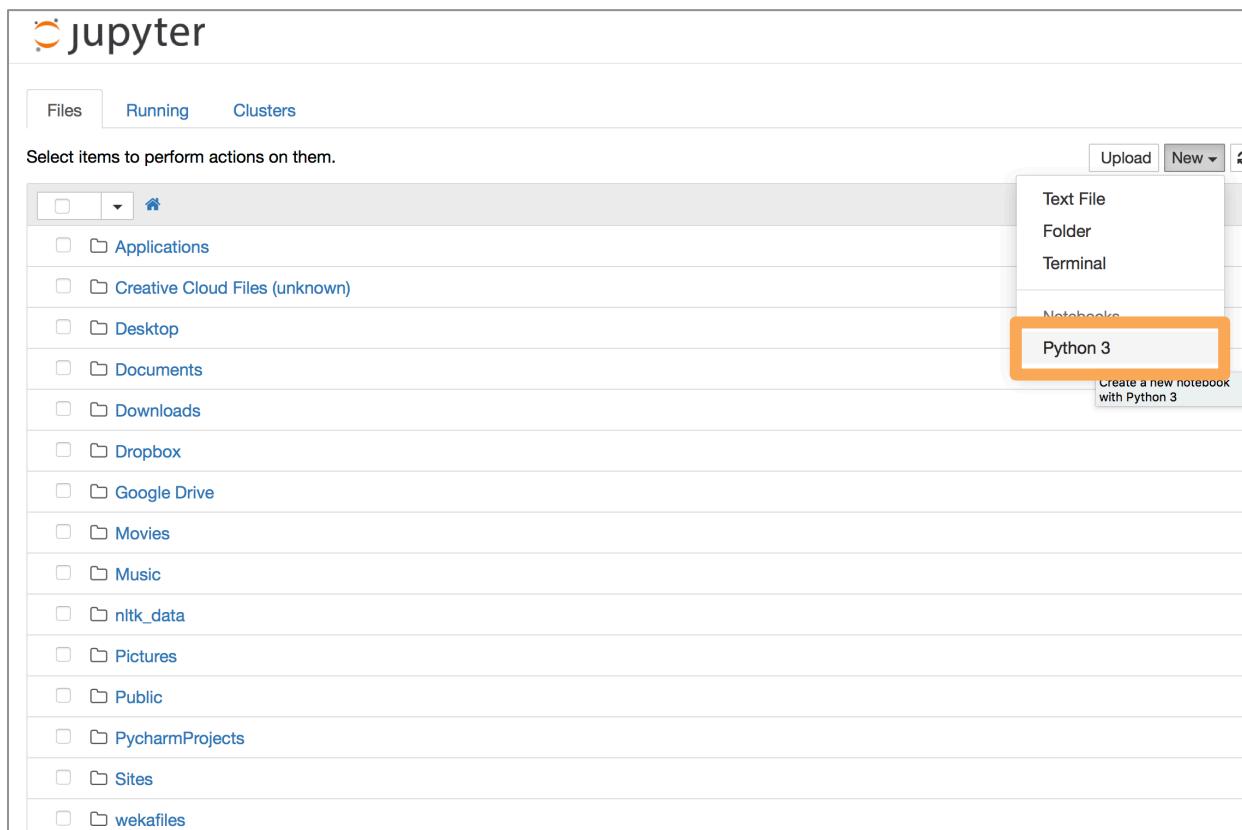
Starting iPython Notebooks

This starts the iPython notebooks server and opens a web interface to allow access to your iPython notebooks and other resources



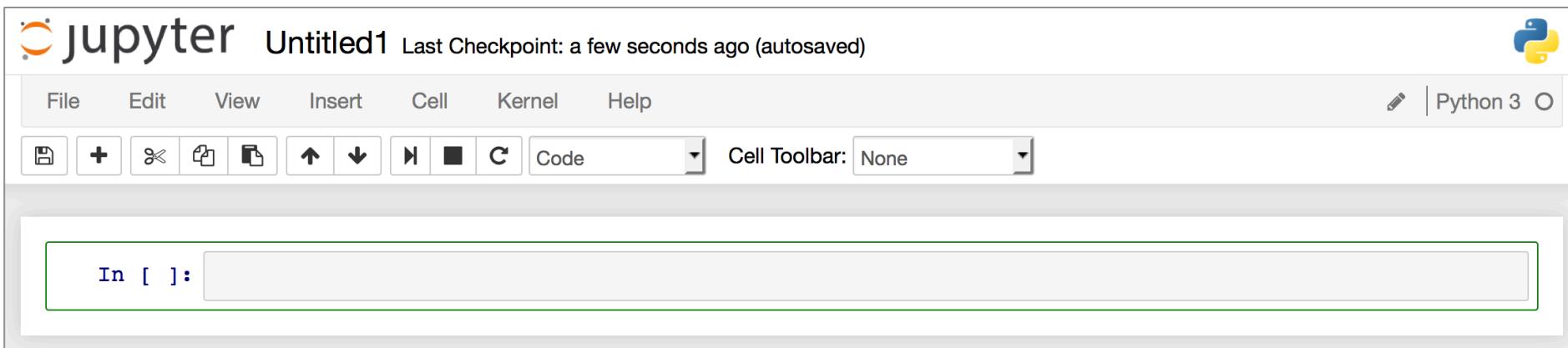
Starting iPython Notebooks

Create a new notebook by clicking on New and selecting Python 3



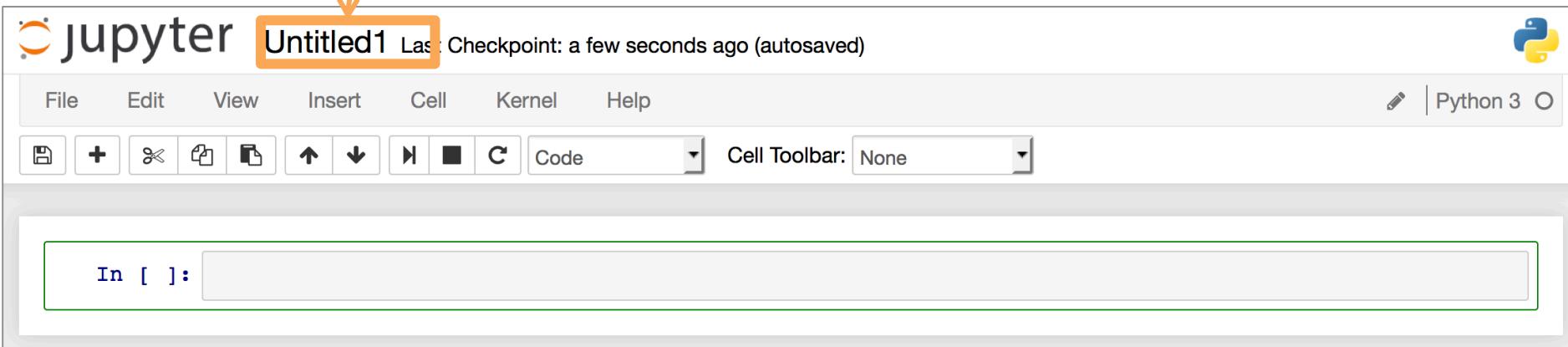
Starting iPython Notebooks

This is your new blank notebook

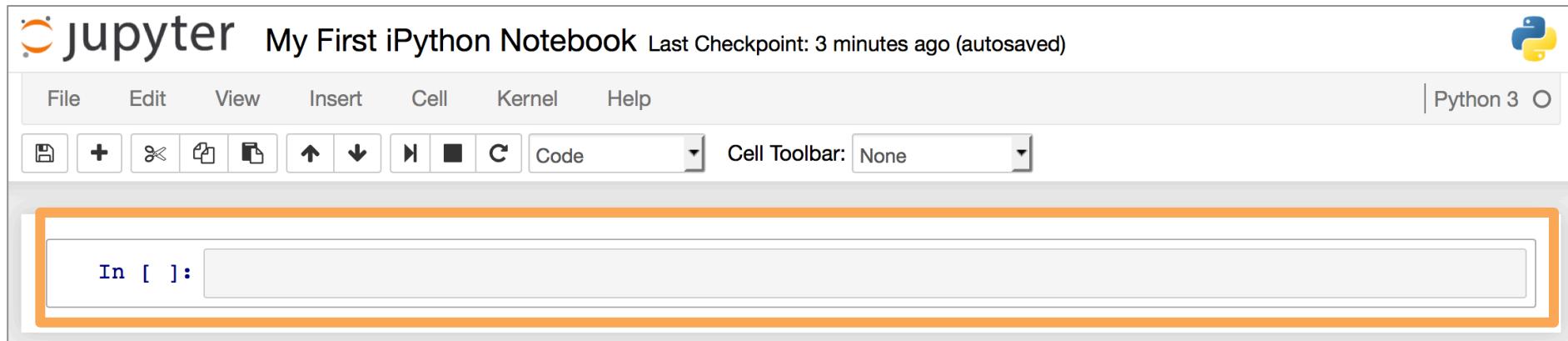


Using iPython Notebooks

Click to give your notebook a useful name



Using iPython Notebooks

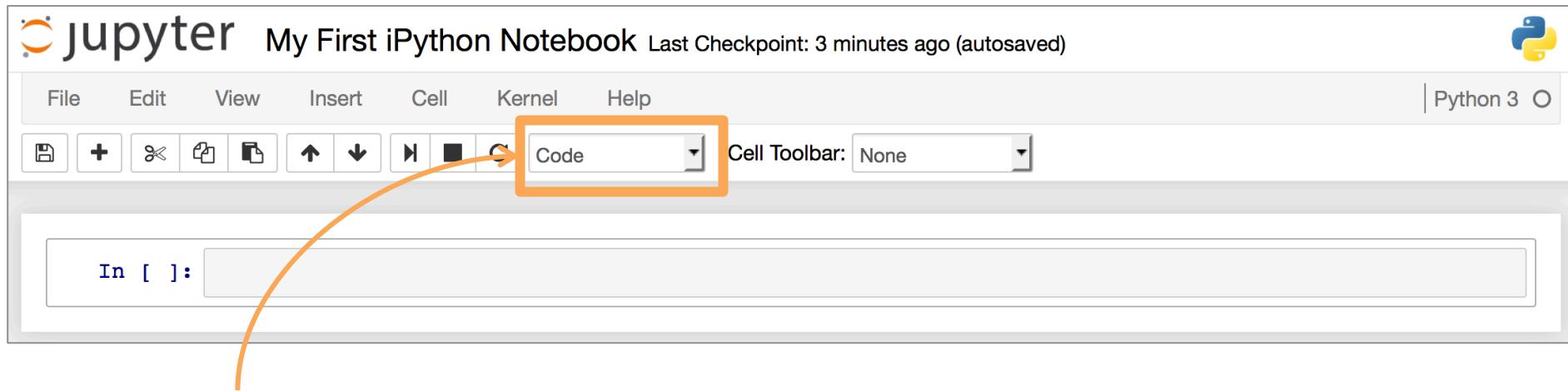


This is a cell. they can be of 2 main types

- **Code**
- **Markdown**

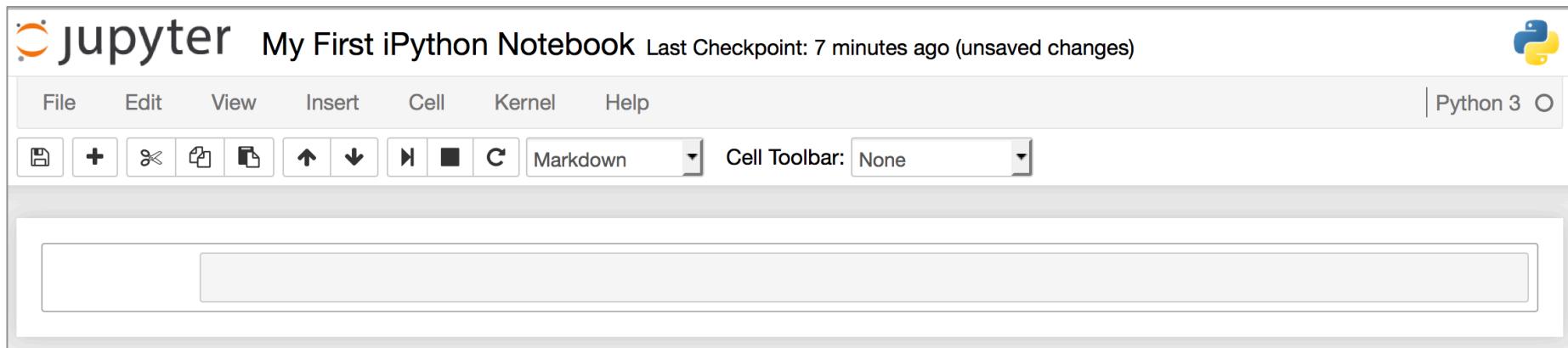
MARKDOWN

Using iPython Notebooks



Change the
type of a
cell here.

Markdown



Markdown is a simple markup language that allows us to write nicely formatted text blocks - it's a little like stripped down HTML

Markdown - Headings

We create headings in markdown by inserting # symbols before text

The number of # symbols determines the level of the heading

#Heading 1

Heading 1

##Heading 2

Heading 2

###Heading 3

Heading 3

####Heading 4

Heading 4

Markdown - Styling

It is easy to style text as italics (one * symbol) or bold (two * symbols)

Normal text

Italic text

Bold text

Normal text

Italic text

Bold text

Markdown - Lists

* or - symbols create unordered lists

This is a list

- * Item 1
- * Item 2
- * Item 3

This is a list

- Item 1
- Item 2
- Item 3

Putting numbers followed by a . symbol before lines creates ordered lists

This is a list

1. Item 1
2. Item 2
3. Item 3

This is a list

1. Item 1
2. Item 2
3. Item 3

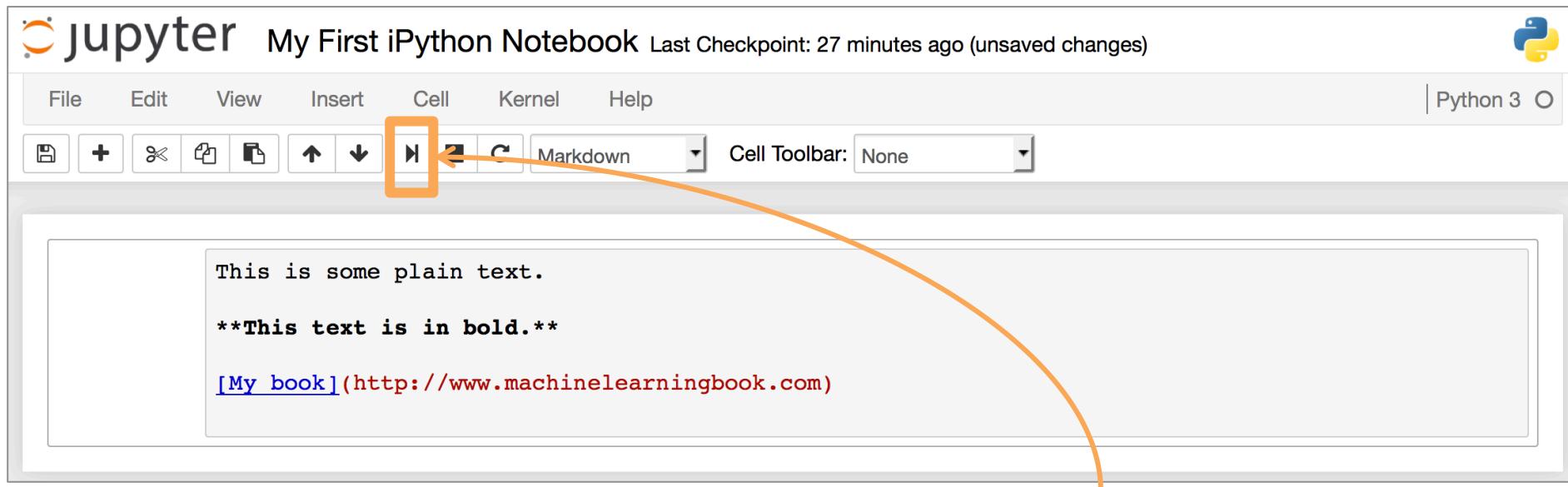
Markdown - Links

Links are created by wrapping the anchor text in square brackets, [], and the URL in regular brackets, ()

[My book](http://www.machinelearningbook.com)

[My book](#)

Markdown



Press this button to run the cell,
which renders the markdown,
and creates a new cell

Markdown

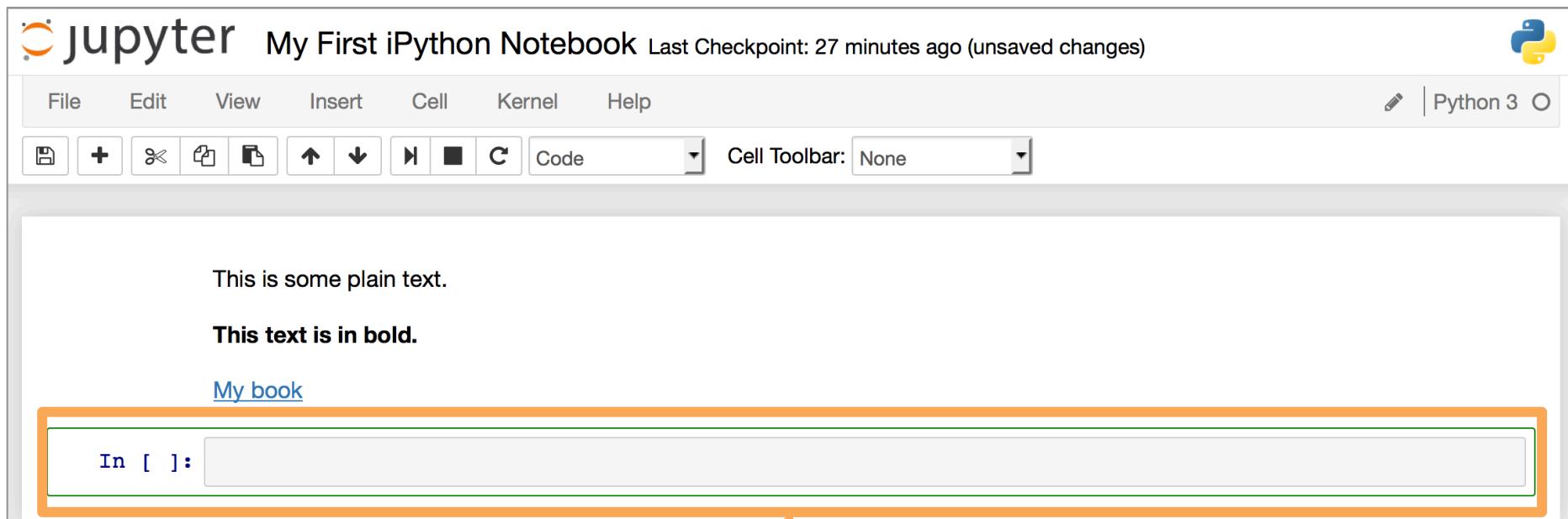
This screenshot shows a Jupyter Notebook interface with the title "My First iPython Notebook" and a status bar indicating "Last Checkpoint: 27 minutes ago (unsaved changes)". The toolbar includes standard file operations like Open, Save, and New, along with cell management icons (Run, Cell, Kernel, Help) and a Cell Toolbar dropdown set to "None". The main content area displays the following Markdown text:

```
This is some plain text.  
This text is in bold.  
My book
```

The input cell at the bottom is labeled "In []:".

CODE CELLS

Code Cells



We can write any Python code
we like in a code cell

Code Cells

This is some plain text.

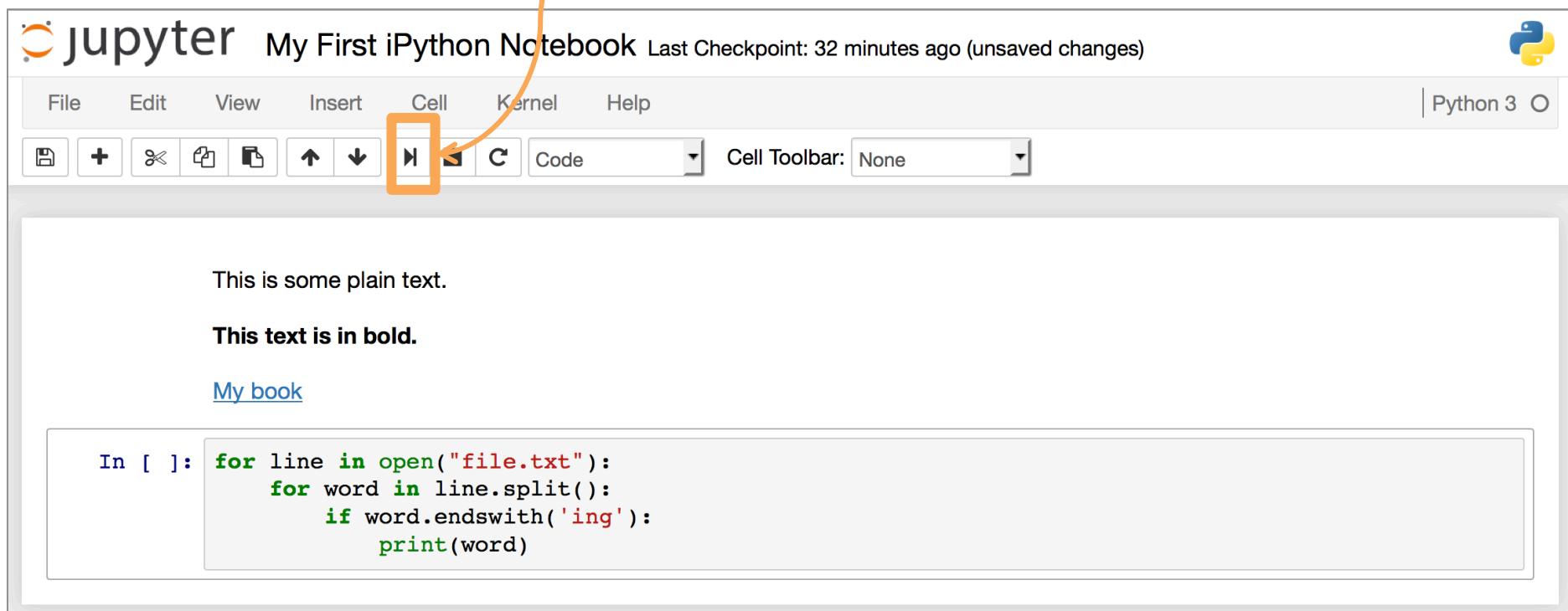
This text is in bold.

[My book](#)

```
In [ ]: for line in open("file.txt"):
          for word in line.split():
              if word.endswith('ing'):
                  print(word)
```

Code Cells

Press this button to
run the code in the cell



Code Cells

This screenshot shows a Jupyter Notebook interface. At the top, there's a toolbar with various icons for file operations like opening, saving, and running cells. The title bar says "jupyter My First iPython Notebook Last Checkpoint: 34 minutes ago (unsaved changes)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, and Help. On the right, there's a Python 3 logo.

The main area contains the following content:

- A text cell with the plain text: "This is some plain text."
- A text cell with bold text: "**This text is in bold.**"
- A text cell with a link: "[My book](#)"
- A code cell labeled "In [1]:" containing the following Python code:

```
for line in open("file.txt"):
    for word in line.split():
        if word.endswith('ing'):
            print(word)
```
- The output of the code cell, which lists words ending in 'ing':

```
nothing
taking
starring
fizzing
trying
everything
Making
something
feeling
following
being
thing
distrusting
everything
thing
everything
```

On the right side of the slide, there is an orange text overlay that reads: "Any output from code appears directly below the cell".

Code Cells

jupyter simple_python_demo Last Checkpoint: Yesterday at 9:41 AM (autosaved) Python 3

In [3]:

```
from IPython.display import Image
Image('http://jakevdp.github.com/figures/xkcd_version.png')
```

Out[3]:

CHECK IT OUT!

DAMPED SINE
DAMPED COSINE

click to expand output;
double click to hide

Output can be interesting things like images or graphs

Code Cells

This is some plain text.

This text is in bold.

[My book](#)

```
In [1]: for line in open("file.txt"):
    for word in line.split():
        if word.endswith('ing')
            print(word)

File "<ipython-input-1-df471fc3f1dd>", line 3
    if word.endswith('ing')
               ^
SyntaxError: invalid syntax"
```

In []:

If there are any errors in your code, you get an error message instead of output

Code Cells

```
In [1]: for line in open("file.txt"):
    for word in line.split():
        if word.endswith('ing')
            print(word)
```

```
File "<ipython-input-1-df471fc3f1dd>", line 3
    if word.endswith('ing')
               ^
SyntaxError: invalid syntax
```

Code Cells

The screenshot shows a Jupyter Notebook interface with the title "My First iPython Notebook" and a status bar indicating "Last Checkpoint: a few seconds ago (unsaved changes)". The toolbar includes File, Edit, View, Insert, Cell, Kernel, Help, and a Python 3 kernel icon. Below the toolbar are standard notebook controls for file operations and cell navigation.

In [2]:

```
a = 12
b = 15
```

Any variables created in one code cell are accessible in any other code cell in the notebook. We say that they exist within the **iPython kernel**.

In [4]:

```
c = a + b
print (c)
```

27

In []:

Python objects created in one cell
in a notebook can be accessed
from any subsequent cell

SUMMARY

Summary

This has been a short introduction to iPython notebooks

They are great for data science or any other scientific computing projects

Notebooks can include other formats such as Latex, R code,

Summary

More info

- iPython Official Documentation:
[http://ipython.readthedocs.org/en/stable/
overview.html](http://ipython.readthedocs.org/en/stable/overview.html)
- Markdown Cheat Sheet:
[http://nestacms.com/docs/creating-content/
markdown-cheat-sheet](http://nestacms.com/docs/creating-content/
markdown-cheat-sheet)