



IIT ROORKEE



NPTEL ONLINE
CERTIFICATION COURSE

Data Analytics with Python

Lecture 2: Python – Fundamentals

Dr. A. Ramesh
DEPARTMENT OF MANAGEMENT
IIT ROORKEE



Learning objectives

1. Installing Python
2. Fundamentals of Python
3. Data Visualisation

Python Installation Process

Installation Process –

Step 1: Type <https://www.anaconda.com> at the address bar of web browser.

Step 2: Click on download button

Step 3: Download python 3.7 version for windows OS

Step 4: Double click on file to run the application

Step 5: Follow the instructions until completion of installation process

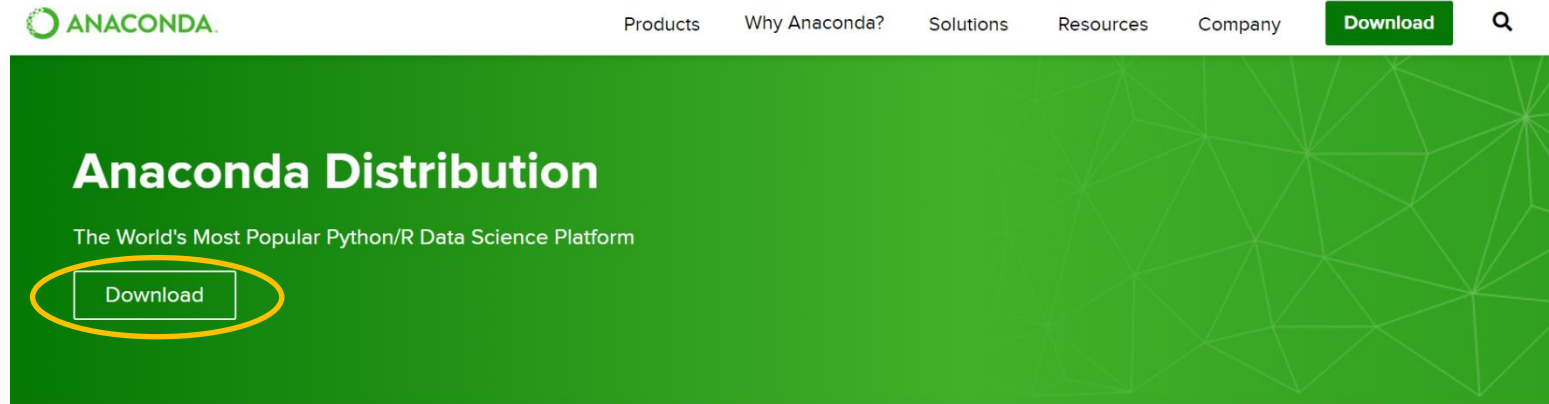
Python Installation Process

Installation Process –

Step 1: Type <https://www.anaconda.com> at the address bar of web browser.

Python Installation Process

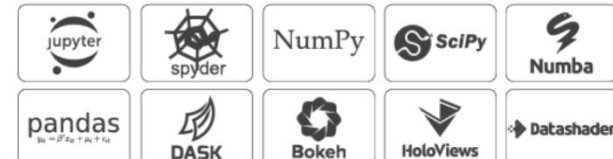
Step 2: Click on download button



The screenshot shows the Anaconda website's header and main banner. The header includes the Anaconda logo, navigation links (Products, Why Anaconda?, Solutions, Resources, Company), a 'Download' button, and a search icon. The main banner has a green background with a geometric pattern. It features the text 'Anaconda Distribution' and 'The World's Most Popular Python/R Data Science Platform'. A 'Download' button is prominently displayed and circled in yellow.

The open-source [Anaconda Distribution](#) is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 11 million users worldwide, it is the industry standard for developing, testing, and training on a single machine, enabling *individual data scientists* to:

- Quickly download 1,500+ Python/R data science packages



Python Installation Process

Step 3: Download python 3.7 version for windows OS

 Windows |  macOS |  Linux

Anaconda 2019.03 for Windows Installer

Python 3.7 version

Download

64-Bit Graphical Installer (662 MB)
32-Bit Graphical Installer (546 MB)

Python 2.7 version

Download

64-Bit Graphical Installer (587 MB)
32-Bit Graphical Installer (493 MB)

Get Started with Anaconda Distribution

Python Installation Process

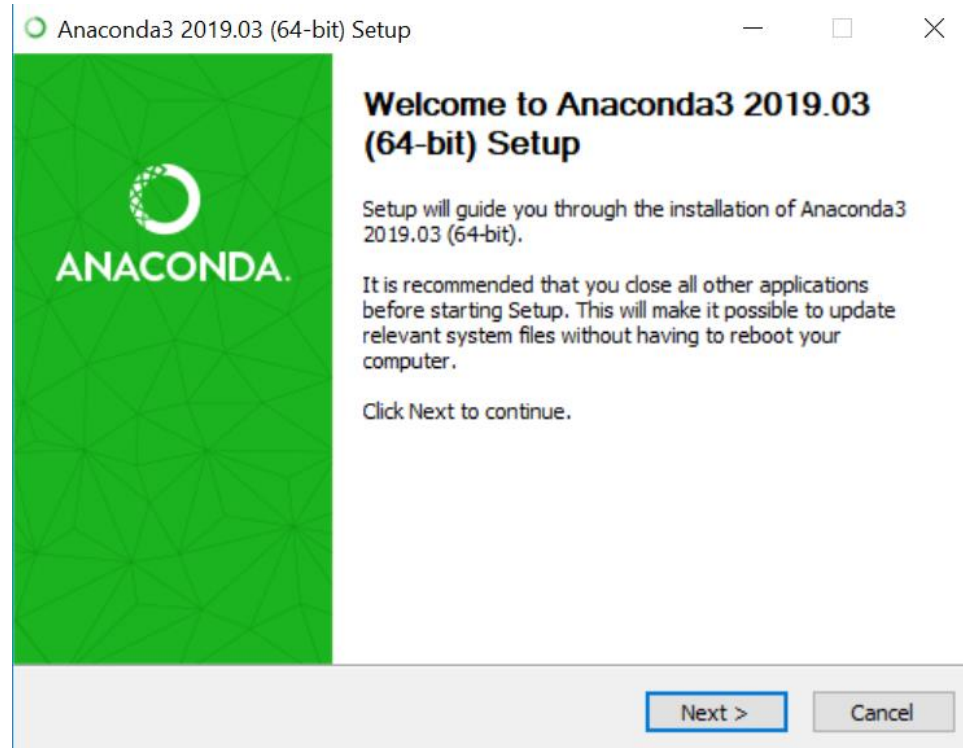
Step 4: Double click on file to run the application



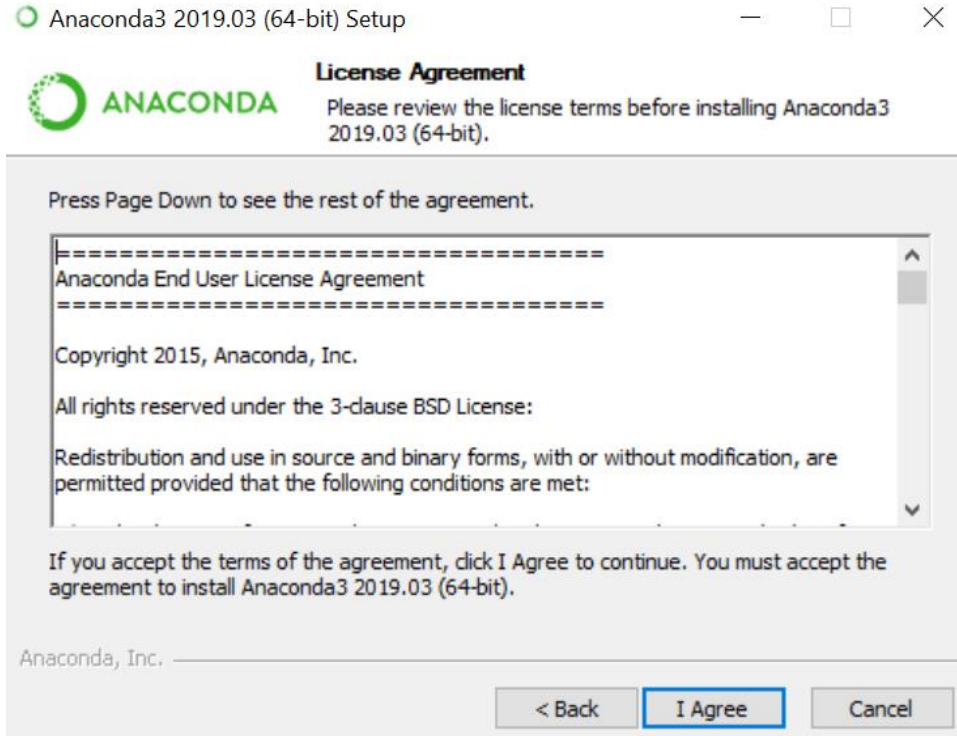
Anaconda3-2018.12-Windows-x86_64_2.exe

F:\Umabharati

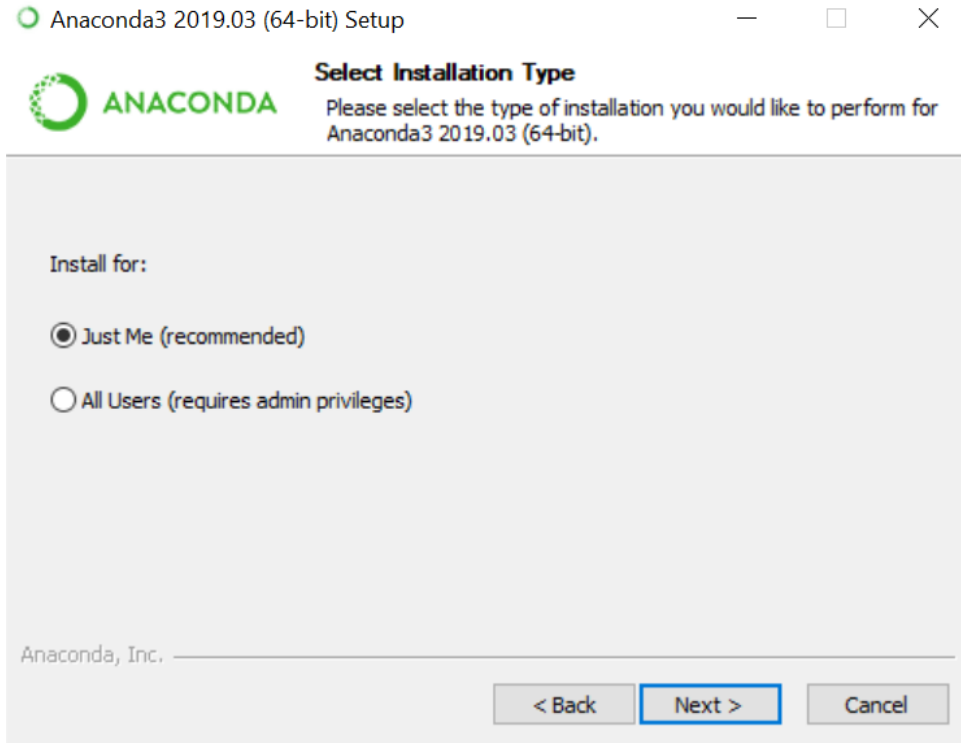
Python Installation Process



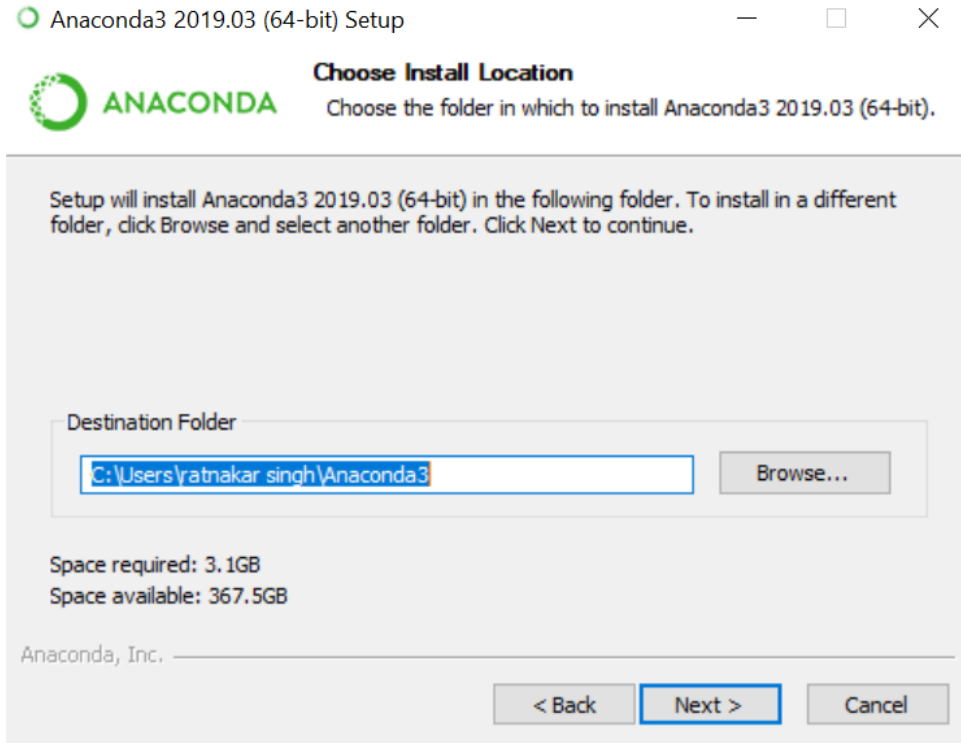
Python Installation Process



Python Installation Process



Python Installation Process



Python Installation Process

Anaconda3 2019.03 (64-bit) Setup



Advanced Installation Options

Customize how Anaconda integrates with Windows

Advanced Options

☒ Add Anaconda to my PATH environment variable

Not recommended. Instead, open Anaconda with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.

☒ Register Anaconda as my default Python 3.7

This will allow other programs, such as Python Tools for Visual Studio, PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.7 on the system.

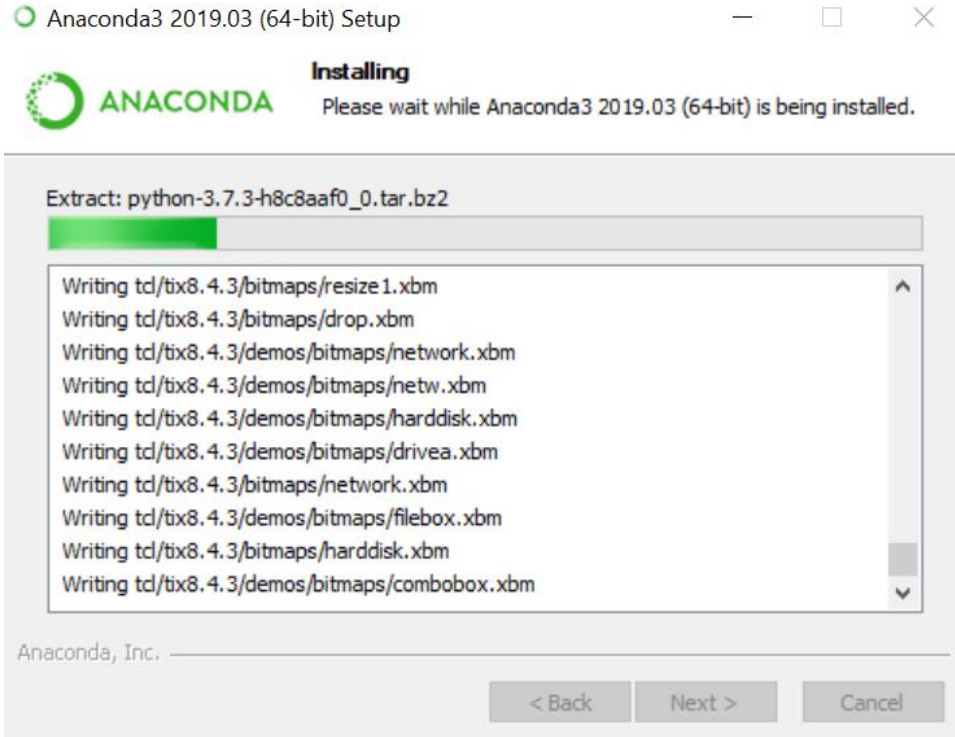
Anaconda, Inc.

< Back

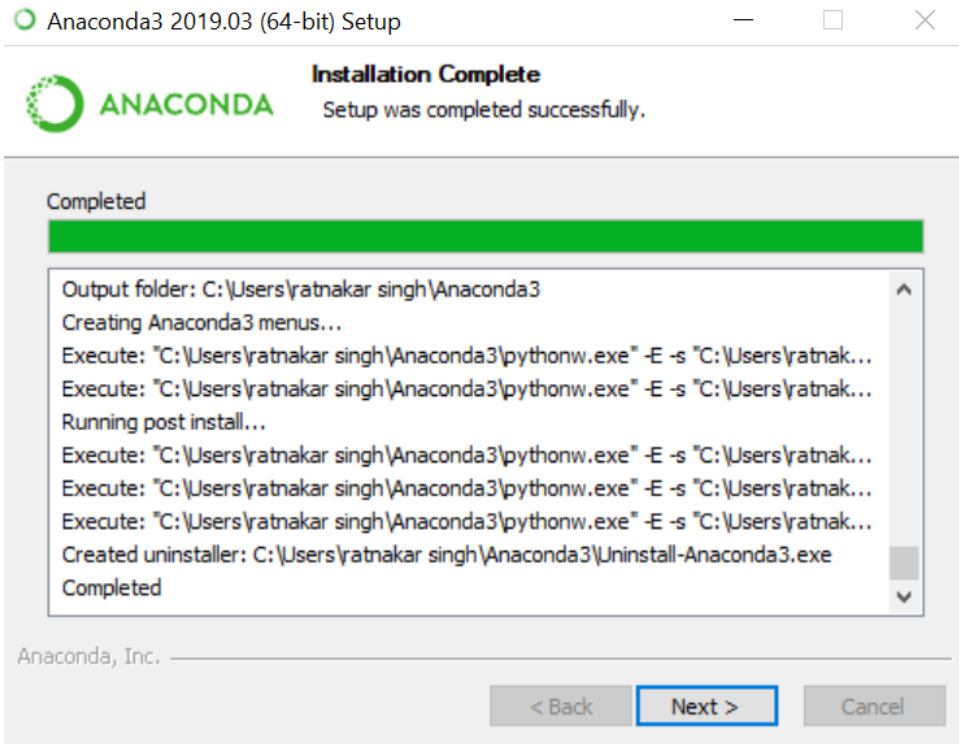
Install

Cancel

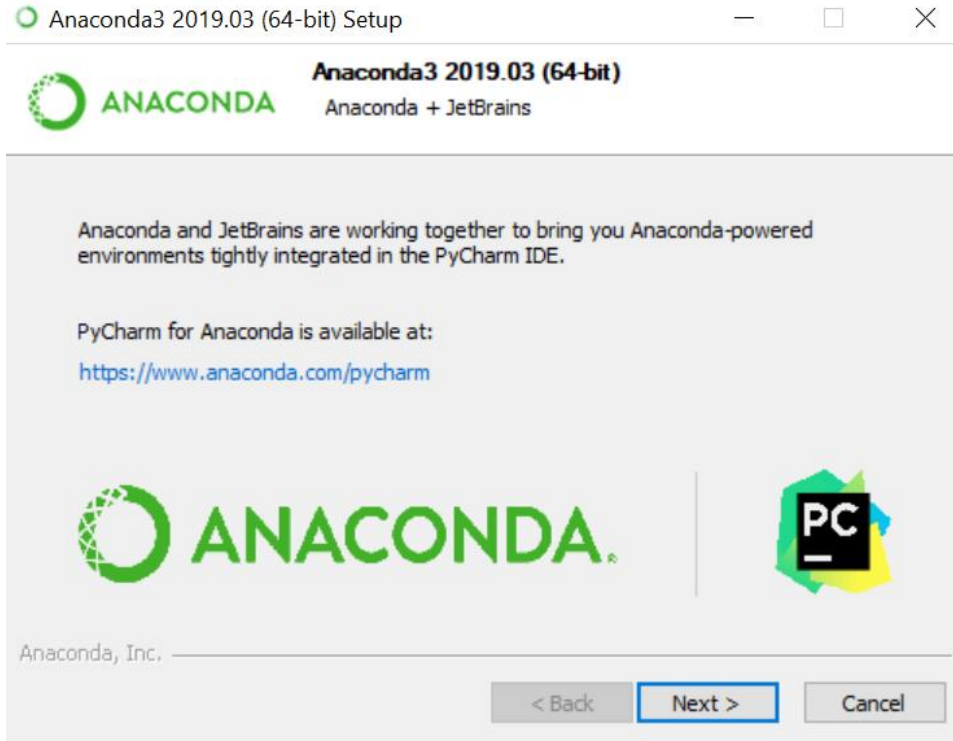
Python Installation Process



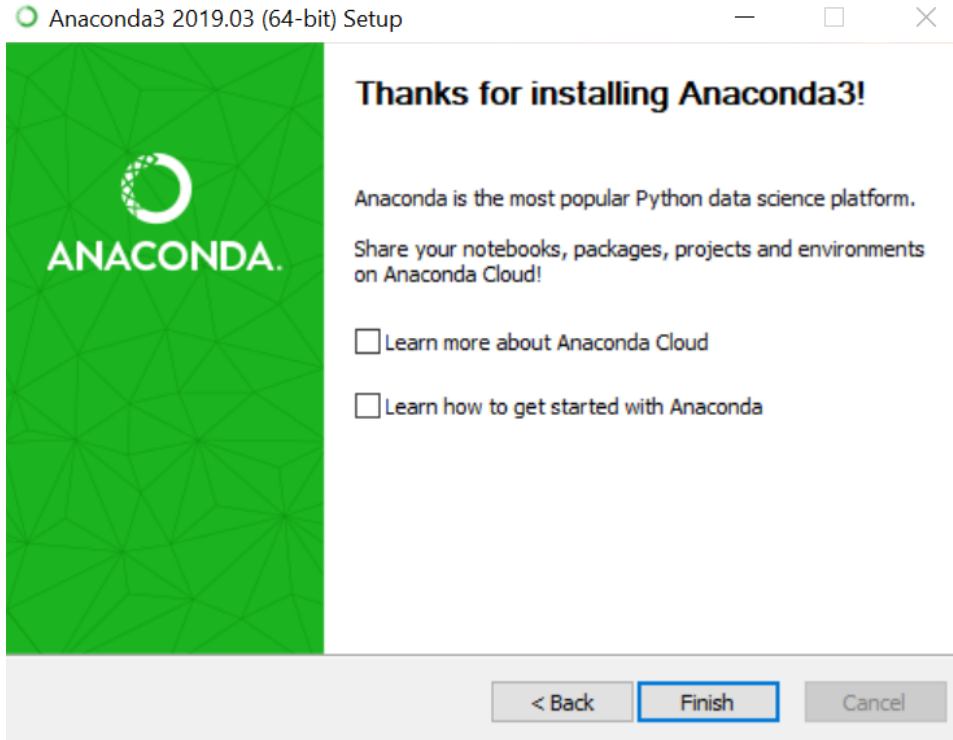
Python Installation Process



Python Installation Process



Python Installation Process



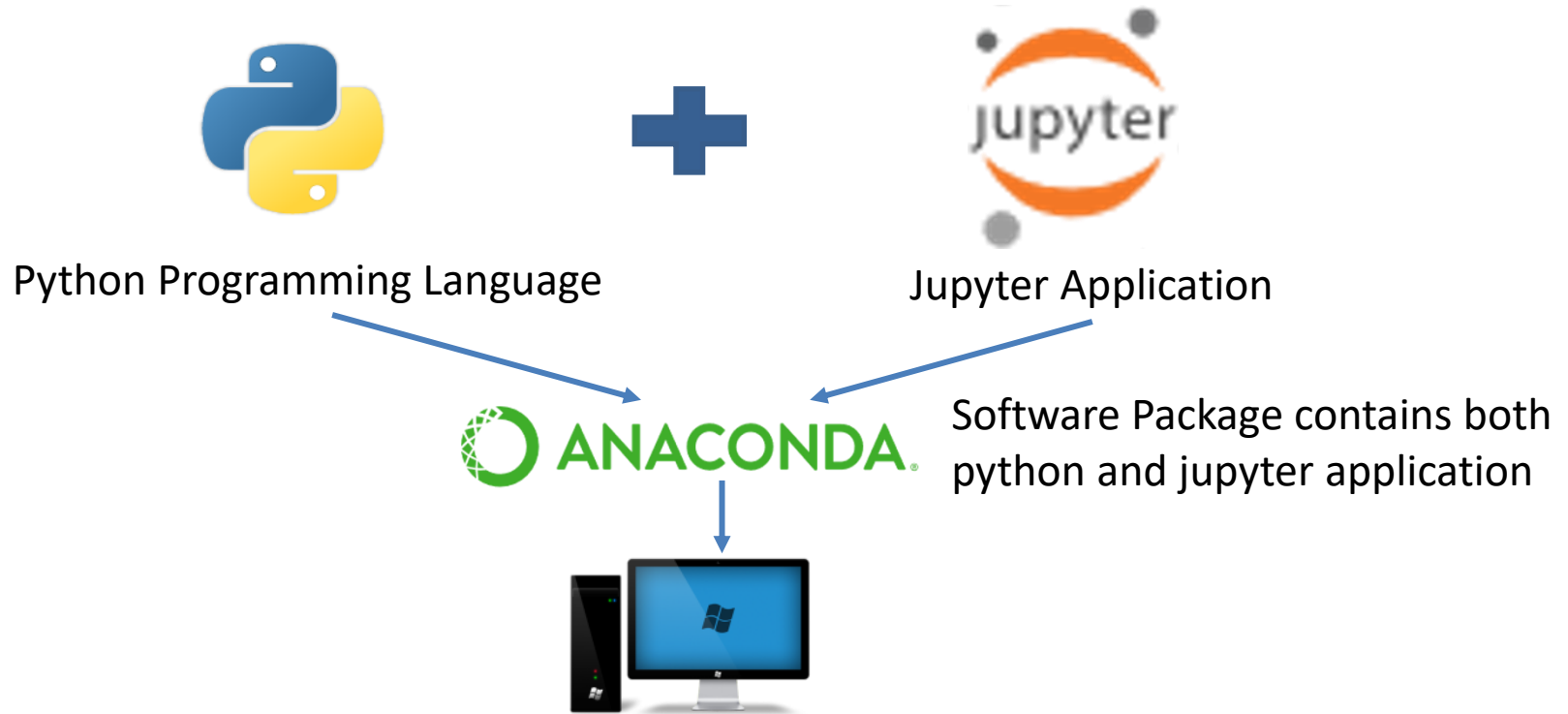
Why Jupyter Notebook?



Why?

- Edit code on web browser
- Easy in documentation
- Easy in demonstration
- User- friendly Interface









Python and Jupyter



Applications on base (root)

Channels

Refresh

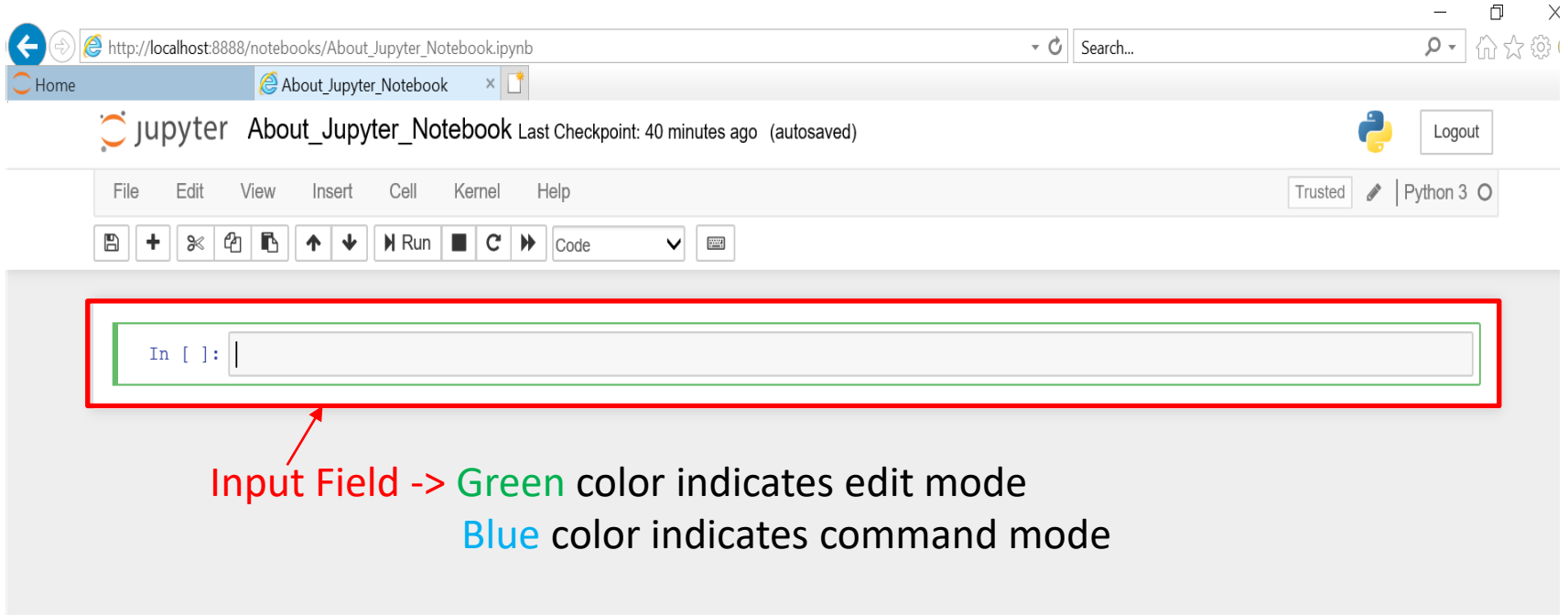
 <p>jupyterlab 0.31.4</p> <p>An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.</p> <p>Launch</p>	 <p>jupyter notebook 5.4.0</p> <p>Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.</p> <p>Launch</p>	 <p>qtconsole 4.3.1</p> <p>PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.</p> <p>Launch</p>	 <p>spyder 3.2.6</p> <p>Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features</p> <p>Launch</p>	 <p>glueviz 0.12.0</p> <p>Multidimensional data visualization across files. Explore relationships within and among related datasets.</p> <p>Install</p>	 <p>orange3 3.4.1</p> <p>Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.</p> <p>Install</p>
 <p>rstudio 1.1.383</p> <p>A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.</p> <p>Install</p>	 <p>vscode 1.36.1</p> <p>Streamlined code editor with support for development operations like debugging, task running and version control.</p> <p>Install</p>				

About Jupyter Notebook

The screenshot displays the Jupyter Notebook web interface in a browser. The address bar shows the URL `http://localhost:8888/notebooks/About_Jupyter_Notebook.ipynb`. The page title is "About_Jupyter_Notebook" with a subtitle "Last Checkpoint: a minute ago (unsaved changes)". The interface includes a top menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", and "Help". Below the menu is a toolbar with icons for saving, adding, deleting, and running cells. The main area shows a code cell with the prompt `In []:` followed by a large empty text input field. This input field is highlighted with a red rectangular border. A red arrow points from the text "Cell -> Access using Enter Key" below the cell to the input field. The bottom of the interface shows a status bar with "Trusted" and "Python 3" indicators.

Cell -> Access using Enter Key

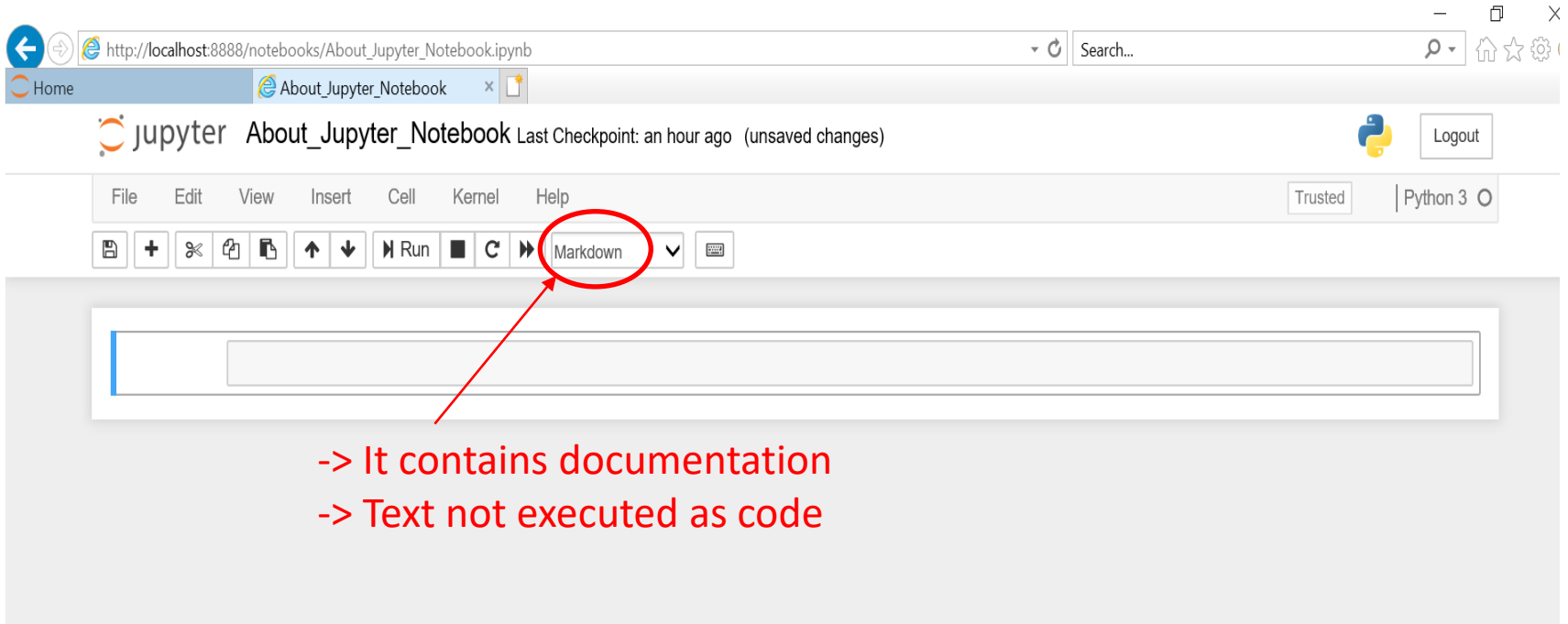
About Jupyter Notebook



The screenshot displays the Jupyter Notebook web interface. The browser address bar shows the URL `http://localhost:8888/notebooks/About_Jupyter_Notebook.ipynb`. The notebook title is "About_Jupyter_Notebook" with a status "Last Checkpoint: 40 minutes ago (autosaved)". The top menu bar includes "File", "Edit", "View", "Insert", "Cell", "Kernel", and "Help". The right side of the top bar shows a "Trusted" status, a pencil icon, and "Python 3". Below the menu bar is a toolbar with icons for saving, adding, deleting, and running cells. The main area contains a code cell with the prompt `In []:` followed by a cursor. The cell is highlighted with a red border, and a red arrow points to the input field. The input field is in edit mode, indicated by a green border. The text "Input Field -> Green color indicates edit mode" and "Blue color indicates command mode" is written below the cell.

Input Field -> Green color indicates edit mode
Blue color indicates command mode

About Jupyter Notebook



The screenshot displays the Jupyter Notebook web interface in a browser. The address bar shows the URL `http://localhost:8888/notebooks/About_Jupyter_Notebook.ipynb`. The page title is "About_Jupyter_Notebook" with a subtitle "Last Checkpoint: an hour ago (unsaved changes)". The top navigation bar includes "File", "Edit", "View", "Insert", "Cell", "Kernel", and "Help". Below this is a toolbar with icons for saving, adding, deleting, and running cells. The "Markdown" button is circled in red, and a red arrow points from it to the text below. The main content area is a large text input field.

-> It contains documentation
-> Text not executed as code

About Jupyter Notebook

- Command mode allow to edit notebook as whole
- To close edit mode (Press Escape key)
- Execution (Three ways)
 - Ctrl +Enter (Output field can not be modified)
 - Shift +Enter (Output field is modified)
 - Run button on Jupyter interface
- Comment line is written preceding with # symbol.

About Jupyter Notebook

- Important shortcut keys
 - A -> To create cell above
 - B -> To create cell below
 - D + D -> For deleting cell
 - M -> For markdown cell
 - Y -> For code cell

Fundamentals of Python

- Loading a simple delimited data file
- Counting how many rows and columns were loaded
- Determining which type of data was loaded
- Looking at different parts of the data by subsetting rows and columns

Pandas for Everyone

Python Data Analysis

Daniel Y. Chen

◆◆Addison-Wesley

Boston • Columbus • Indianapolis • New York • San Francisco • Amsterdam • Cape Town

Dubai • London • Madrid • Milan • Munich • Paris • Montreal • Toronto • Delhi • Mexico City

São Paulo • Sydney • Hong Kong • Seoul • Singapore • Taipei • Tokyo



Loading a simple delimited data file

```
In [1]: import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt
```

```
In [23]: df = pandas.read_csv('F:/2019-20/NPTEL/2 Introduction to Python/data/gapminder-FiveYearData.csv')
```

```
In [20]: df
```

[Data Source: www.github.com/jennybc/gapminder.](https://www.github.com/jennybc/gapminder)

Out[29]:

	country	year	pop	continent	lifeExp	gdpPercap
0	Afghanistan	1952	8425333.0	Asia	28.801	779.445314
1	Afghanistan	1957	9240934.0	Asia	30.332	820.853030
2	Afghanistan	1962	10267083.0	Asia	31.997	853.100710
3	Afghanistan	1967	11537966.0	Asia	34.020	836.197138
4	Afghanistan	1972	13079460.0	Asia	36.088	739.981106
5	Afghanistan	1977	14880372.0	Asia	38.438	786.113360
6	Afghanistan	1982	12881816.0	Asia	39.854	978.011439
7	Afghanistan	1987	13867957.0	Asia	40.822	852.395945

- `head` method shows us only the first 5 rows

```
In [24]: print(df.head())
```

	country	year	pop	continent	lifeExp	gdpPercap
0	Afghanistan	1952	8425333.0	Asia	28.801	779.445314
1	Afghanistan	1957	9240934.0	Asia	30.332	820.853030
2	Afghanistan	1962	10267083.0	Asia	31.997	853.100710
3	Afghanistan	1967	11537966.0	Asia	34.020	836.197138
4	Afghanistan	1972	13079460.0	Asia	36.088	739.981106

Get the number of rows and columns

```
In [25]: print(df.shape)  
(1704, 6)
```

get column names

```
In [26]: print(df.columns)
```

```
Index(['country', 'year', 'pop', 'continent', 'lifeExp', 'gdpPercap'], dtype='object')
```

get the dtype of each column

```
In [27]: print(df.dtypes)
```

```
country      object  
year         int64  
pop          float64  
continent    object  
lifeExp      float64  
gdpPercap    float64  
dtype: object
```


Pandas Types Versus Python Types

Pandas Type	Python Type	Description
object	string	Most common data type
int64	int	Whole numbers
float64	float	Numbers with decimals
datetime64	datetime	datetime is found in the Python standard library (i.e., it is not loaded by default and needs to be imported)

get more information about data

```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 1704 entries, 0 to 1703  
Data columns (total 6 columns):  
country      1704 non-null object  
continent    1704 non-null object  
year         1704 non-null int64  
lifeExp      1704 non-null float64  
pop          1704 non-null int64  
gdpPercap    1704 non-null float64  
dtypes: float64(2), int64(2), object(2)  
memory usage: 80.0+ KB  
None
```

Looking at Columns, Rows, and Cells

- *# get the country column and save it to its own variable*

```
In [31]: country_df = df['country']
```

show the first 5 observations

```
In [32]: print(country_df.head())
```

```
0    Afghanistan
1    Afghanistan
2    Afghanistan
3    Afghanistan
4    Afghanistan
Name: country, dtype: object
```

show the last 5 observations

```
In [33]: print(country_df.tail())
```

```
1699    Zimbabwe  
1700    Zimbabwe  
1701    Zimbabwe  
1702    Zimbabwe  
1703    Zimbabwe  
Name: country, dtype: object
```

Looking at country, continent, and year

```
In [34]: subset = df[['country', 'continent', 'year']]  
print(subset.head())
```

	country	continent	year
0	Afghanistan	Asia	1952
1	Afghanistan	Asia	1957
2	Afghanistan	Asia	1962
3	Afghanistan	Asia	1967
4	Afghanistan	Asia	1972

```
In [35]: print(subset.tail())
```

	country	continent	year
1699	Zimbabwe	Africa	1987
1700	Zimbabwe	Africa	1992
1701	Zimbabwe	Africa	1997
1702	Zimbabwe	Africa	2002
1703	Zimbabwe	Africa	2007