



python
Advance Level

Done By: Eng. Mohammed Marwan Shahin
Technical Training Specialist
Technology Academy

Training Outlines



- Anaconda Installation
- File Handling
- Web Scraping Using re module & BeautifulSoup
- Exception Handling
- Regular Expressions
- Sending Mails
- Excel automation
- Multi-threading
- Lambda Function
- Communicating with external databases (Sqlite)
- Create GUI using Tkinter

Python - Advance Level

Lesson 01 : Anaconda Distribution



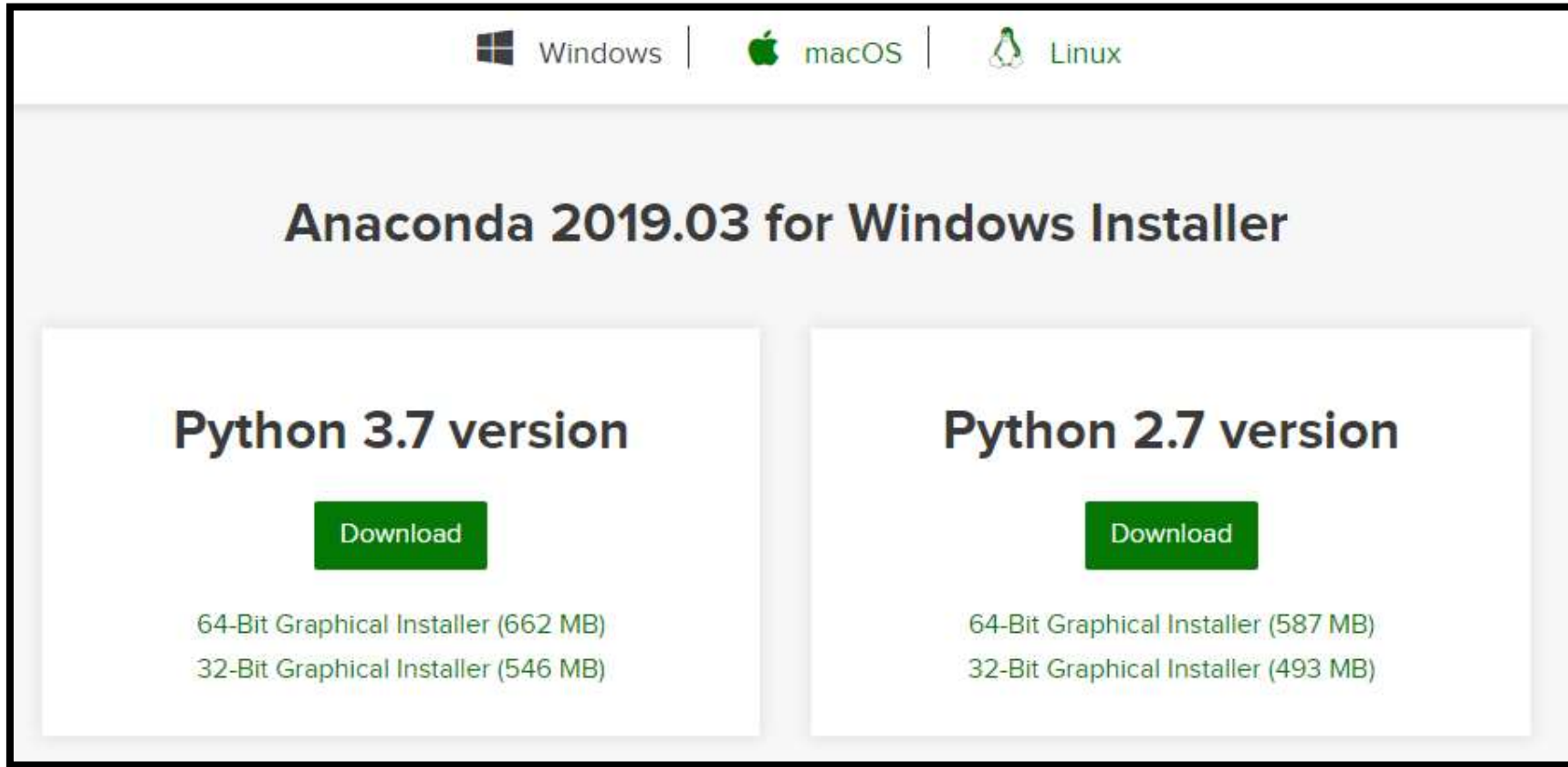
Anaconda Distribution



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:

1. Quickly download **1,500+ Python/R data science packages**
2. Manage libraries, dependencies, and environments with **Conda**
3. Develop and train machine learning and deep learning models with **scikit-learn, TensorFlow, and Theano**
4. Analyze data with scalability and performance with **NumPy& pandas.**
5. Visualize results with **Matplotlib, Bokeh, Datashader, and Holoviews**

Anaconda Installation

A screenshot of the Anaconda 2019.03 for Windows Installer download page. The page has a dark blue header with the Python logo. Below the header, there are three tabs: Windows, macOS, and Linux. The main heading is "Anaconda 2019.03 for Windows Installer". There are two main sections: "Python 3.7 version" and "Python 2.7 version". Each section has a green "Download" button and two links: "64-Bit Graphical Installer" and "32-Bit Graphical Installer".

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 \(493 MB\)](#)

Or (<https://docs.anaconda.com/anaconda/install/windows/>)

Anaconda Installation



[Download the Anaconda installer.](#)

Double click the installer to launch.

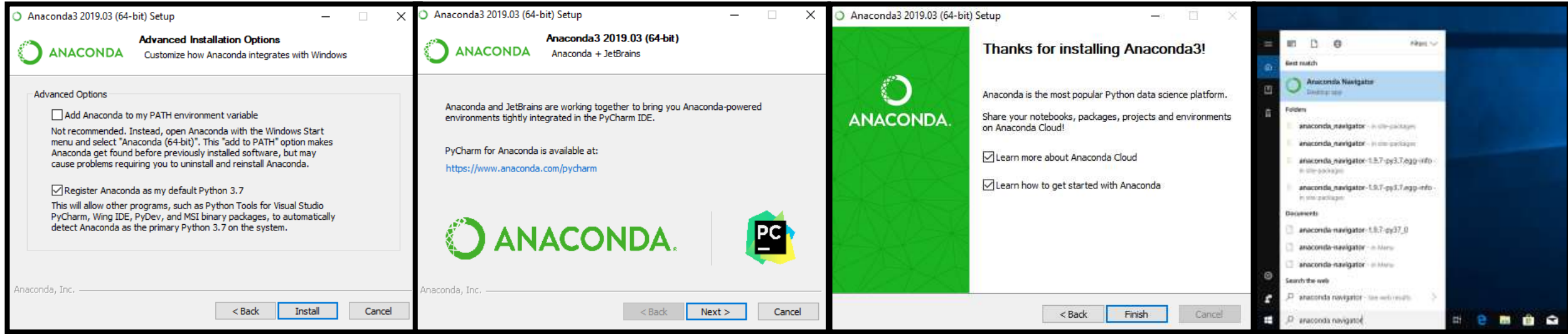
Click Next.

Read the licensing terms and click “I Agree”.

Select an install for “Just Me”

Select a destination folder to install Anaconda and click the Next button. !

Anaconda Installation

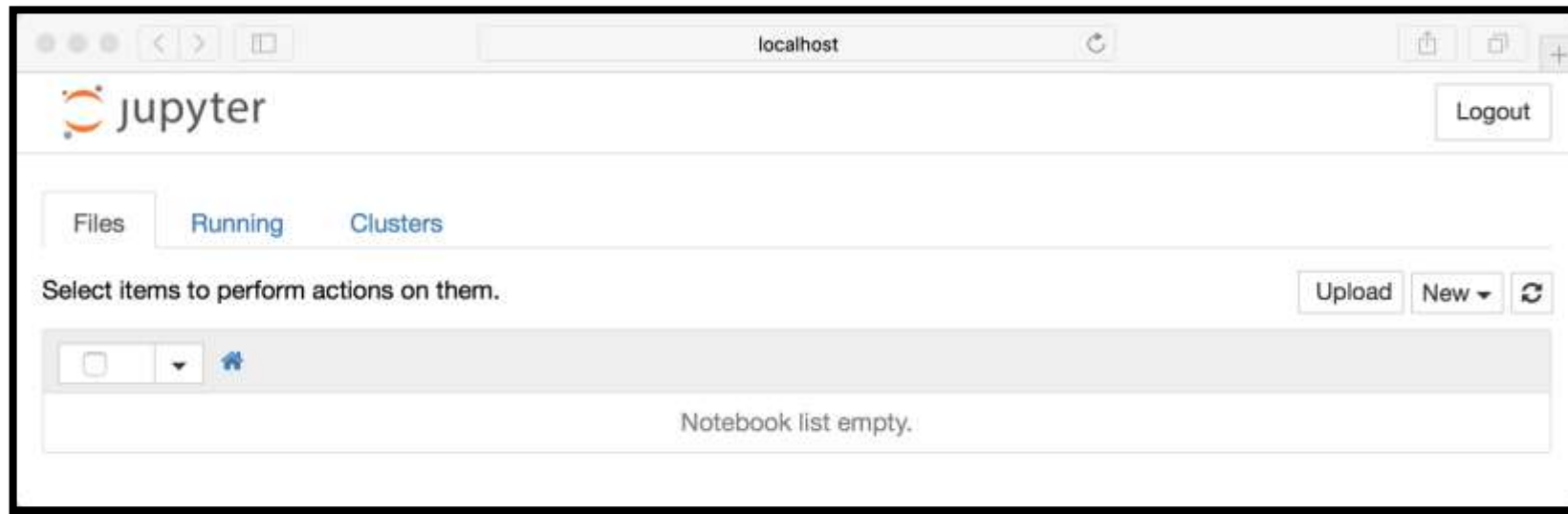


Jupyter Notebook



The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text.

The name, Jupyter, comes from the core supported programming languages that it supports: **Julia**, **Python**, and **R**.





Installing Python Packages

pip:

If you use pip, you can install it with

```
pip install Package name
```

How to use pip from the Jupyter Notebook:

If you're using the Jupyter notebook and want to install a package with pip, you similarly might be inclined to run pip directly in the shell:

```
In [3]: # DON'T DO THIS
!pip install numpy

Requirement already satisfied: numpy in /Users/jakevdp/anaconda/envs/python3.6
```

Python - Advance Level

Lesson 02 : File Handlings



FILE IO



What is a file?

File is a named location on disk to store related information. It is used to permanently store data in a non-volatile memory (e.g. hard disk).

Since, random access memory (RAM) is volatile which loses its data when computer is turned off, we use files for future use of the data.

When we want to read from or write to a file we need to open it first. When we are done, it needs to be closed, so that resources that are tied with the file are freed.

Hence, in Python, a file operation takes place in the following order.

Open a file

Read or write (perform operation)

Close the file

Reading Text Files



One way to read or write a file in Python is to use the built-in **open** function. The **open** function provides a **File object** that contains the methods and attributes you need in order to **read, save, and manipulate the file**.

File object
file = **open**(*File Path* *File name* *Mode*)
"/resources/data/Example1.txt", "r")

In this chapter we only cover two modes:

- ****r**** Read mode for reading files
- ****w**** Write mode for writing files

FILE IO modes



'r'

Open a file for reading. (default)

'w'

Open a file for writing. Creates a new file if it does not exist or truncates the file if it exists.

'a'

Open for appending at the end of the file without truncating it. Creates a new file if it does not exist.

'+'

Open a file for updating (reading and writing)

Reading Text Files



For the next example, we will use the text file **Example1.txt**. The file is shown in figure 2:

We read the file:

```
>>>example1="/resources/data/Example1.txt"  
>>>file1 = open(example1,"r")
```

We can view the attributes of the file.

The name of the file:

```
>>>file1.name  
>>>file1.mode
```

We can read the file and assign it to a variable :

```
FileContent=file1.read()  
FileContent
```

This is line 1
This is line 2
This is line 3



How to close a file Using Python?

When we are done with operations to the file, we need to properly close the file. Closing a file will free up the resources that were tied with the file and is done using Python **close()** method. Python has a garbage collector to clean up unreferenced objects but, we must not rely on it to close the file

```
f = open("test.txt")  
# perform file operations  
f.close()
```

The best way to do this is using the **with** statement. This ensures that the file is closed when the block inside with is exited. We don't need to explicitly call the **close()** method. It is done internally.

```
with open("test.txt") as f:  
    # perform file operations
```

FILE IO methods



We can see that, the `read()` method returns newline as `'\n'`. Once the end of file is reached, we get empty string on further reading.

- We can change our current file cursor (position) using the `seek()` method. Similarly, the `tell()` method returns our current position (in number of bytes).

```
f.tell() # get the current file position
```

```
f.seek(0) # change our current file cursor (position)
```

- we can use `readline()` method to read individual lines of a file. This method reads a file till the newline, including the newline character.

```
f.readline()
```

- The `readlines()` method returns a list of remaining lines of the entire file. All these reading method return empty values when end of file (EOF) is reached.

```
f.readlines()
```




Reading Text Files (Example)

```
>>> file1 = open("testfile.txt", "r")
>>> file1.read()
'This is a Python File. \n\nPython is fantastic language. \n\nMac OS is a fantastic OS. \n'
>>> file1.read()
''
>>> file1.tell()
84
>>> file1.seek(0,0)
0
>>> file1.read()
'This is a Python File. \n\nPython is fantastic language. \n\nMac OS is a fantastic OS. \n'
>>> file1.seek(0,0)
0
>>> file1.read(21)
'This is a Python File'
>>> file1.read(21)
'. \n\nPython is fantast'
>>>
```

Try the followings:



```
readMe2 = open('exampleFile.txt', 'r').readlines()  
print(readMe2)
```

```
>>> ===== RESTART =====  
>>>  
Some text  
even more text  
even more text  
even more text  
>>> ===== RESTART =====  
>>>  
Some text  
even more text  
even more text  
even more text  
['Some text', 'even more text', 'even more text', 'even  
>>>
```



Write Text Files

We can open a file object using the method `** write() **` to save the text file to a list. To write the mode, argument must be set to write **w***. *Let's write a file *Example2.txt* with the line: "This is line A"

We Write the file:

```
>>>a=open("Marwan.txt","w+")
>>>a.write("Etisalat\n")
```

To close the file

```
>>>a.close()
```

```
with open("test.txt",'w') as f:
    f.write("my first file\n")
    f.write("This file\n\n")
    f.write("contains three lines\n")
```

"This program will create a new file named 'test.txt' if it does not exist. If it does exist, it is overwritten. We must include the newline characters ourselves to distinguish different lines."



Write Text Files (Example)

```
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct 5 2014, 20:15:28)
[GCC 4.2.1 (Apple Inc. build 5577)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> f = open("testfile.txt", "w")
>>> f.read()

>>> f.write("I have entered some text into this file \n let's see if this works")
65
>>> f.close()

>>> f = open("testfile.txt", "r")
>>> f.read()
"I have entered some text into this file \n let's see if this works"
>>> I
```



Append Text Files

```
>>> f = open("testfile.txt", "a")
>>> f.write("\n This is my appending text to my testfile")
42
>>> f.close()
>>> f = open("testfile.txt", "r")
>>> f.read()
"I have entered some text into this file \n let's see if this works\n This is my appending text to my testf
ile"
>>> f.close()
\
>>> f = open("testfile.txt", "a+")
>>> f.write("\n This is some gibberish ")
25
>>> f.read()
"
>>> f.seek(0,0)
0
>>> f.read()
"I have entered some text into this file \n let's see if this works\n This is my appending text to my testf
ile\n This is some gibberish "
>>> |
```



Quiz
1

Which of the following command is used to open a file “c:\temp.txt” in read-mode only?

- ☐ infile = open(“c:\temp.txt”, “r”)
- ☐ infile = open(“c:\\temp.txt”, “r”)
- ☐ infile = open(file = “c:\temp.txt”, “r+”)
- ☐ infile = open(file = “c:\\temp.txt”, “r+”)



Quiz
2

What does the readlines() method returns?

- ☐ str
- ☐ a list of lines
- ☐ list of single characters
- ☐ list of integers



Quiz
3

Which of the following command is used to open a file “c:\temp.txt” in append-mode?

- ☐ outfile = open(“c:/temp.txt”, “a”)
- ☐ outfile = open(“c:\\temp.txt”, “rw”)
- ☐ outfile = open(“c:\temp.txt”, “w+”)
- ☐ outfile = open(“c:\\temp.txt”, “r+”)
- ☐ outfile = open(“c:\\temp.txt”, “a”)



Exercise 1



Write a Python program (using function) to read an entire text file.

- Content of text.txt

What is Python language? Python is a widely used high-level, general-purpose, interpreted, dynamic programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than possible in languages such as C++ or Java. Python supports multiple programming paradigms, including object-oriented, imperative and functional program.

Exercise 1



Write a Python program (using function) to read an entire text file.

- Content of text.txt

What is Python language? Python is a widely used high-level, general-purpose, interpreted, dynamic programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than possible in languages such as C++ or Java. Python supports multiple programming paradigms, including object-oriented, imperative and functional program.

```
def file_read(fname):  
    txt = open(fname)  
    print(txt.read())  
  
file_read('test.txt')
```

Exercise 2



Write a Python program using function to write/append text to a file and display the text.

Exercise 2



Write a Python program to write/append text to a file and display the text.

```
def file_read(fname):  
    with open(fname, "w") as myfile:  
        myfile.write("Python Exercises\n")  
        myfile.write("Java Exercises")  
    txt = open(fname)  
    print(txt.read())  
  
file_read('abc.txt')
```

Exercise 3



Write a Python program to read a file line by line and store it into a list.

Exercise 3



Write a Python program to read a file line by line and store it into a list.

```
def file_read(fname):  
    with open(fname) as f:  
        #Content_list is the list that contains the read lines.  
        content_list = f.readlines()  
        print(content_list)  
  
file_read(\'test.txt\')
```

Python - Advance Level

Lesson 3 : Web Scrapping



Web Scraping



Web Scraping is a technique employed to extract large amount of data from websites whereby the data is extracted and saved to a local file in your computer or to a database.



Webpages



Web Scraping



Use Cases



E-commerce
portals



Market Research

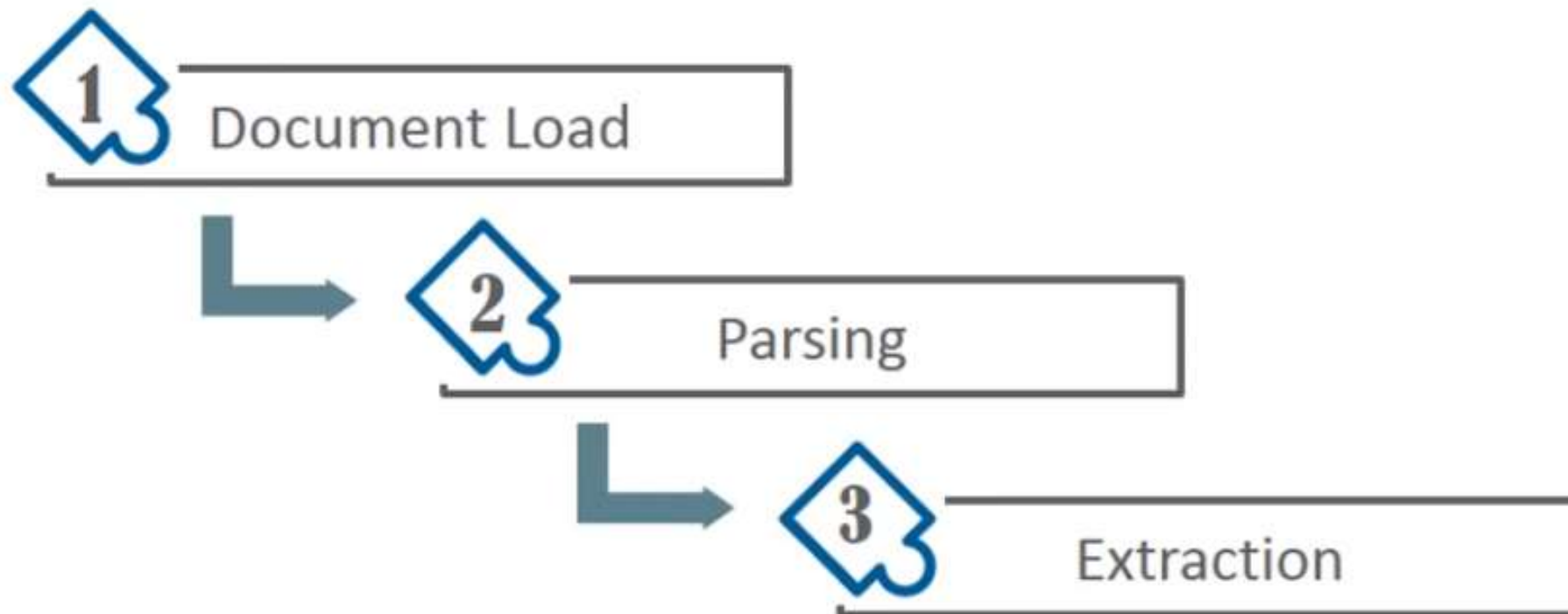


Social
Websites



Travel Websites

Web Scrapping



Web Scraping Consideration



It's important to read and understand the legal information and terms and conditions mentioned in the website.



Different Python Packages



- 1 Pattern
- 2 Scrapy
- 3 Mechanize

- 4 BeautifulSoup
- 5 Requests