PYTHON QUICK REFRESHER

Introduction to Python

What is Python?

- Python is one of the most popular programming languages, created by **Guido van Rossum** and released in 1991.
- Python is an **interpreted** programming language as opposed to a compiled programming language.

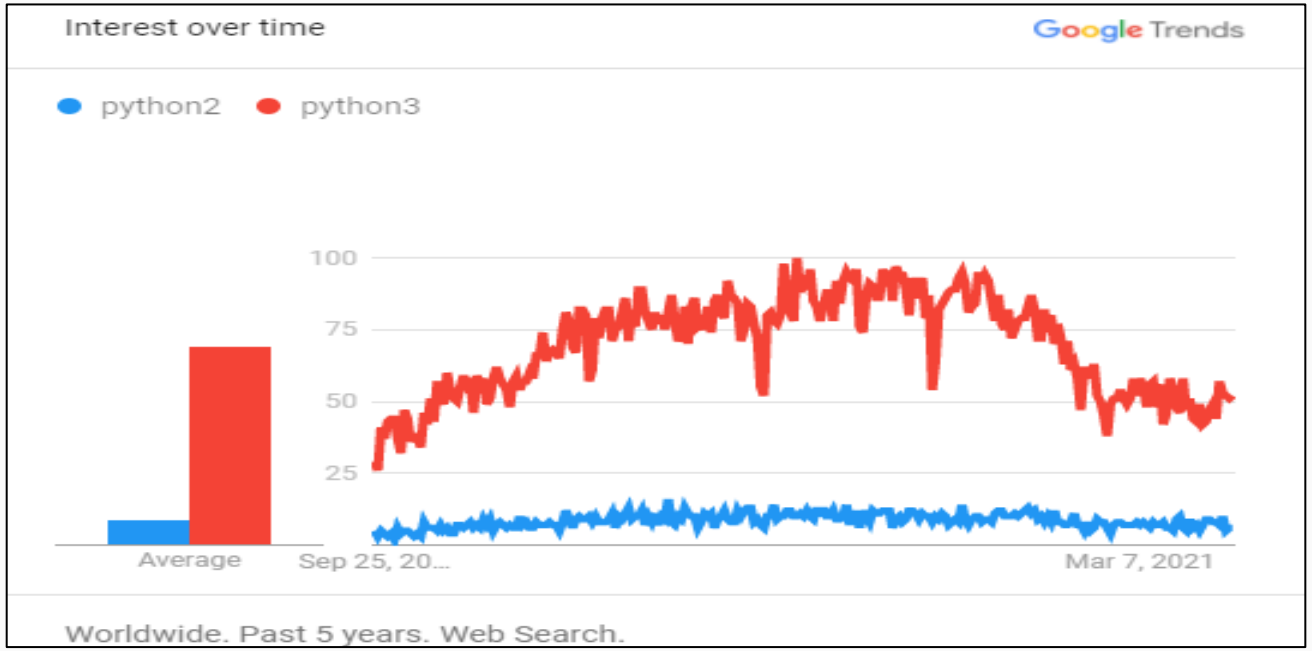
Why python?

- Python is a beginner-friendly language due to its simplified syntax which is close to natural language.
- It has been designed to write clear, logical code for small and large-scale projects.
- It is used extensively in the industry for software development, data science, artificial intelligence, and machine learning.

Setting Up Python

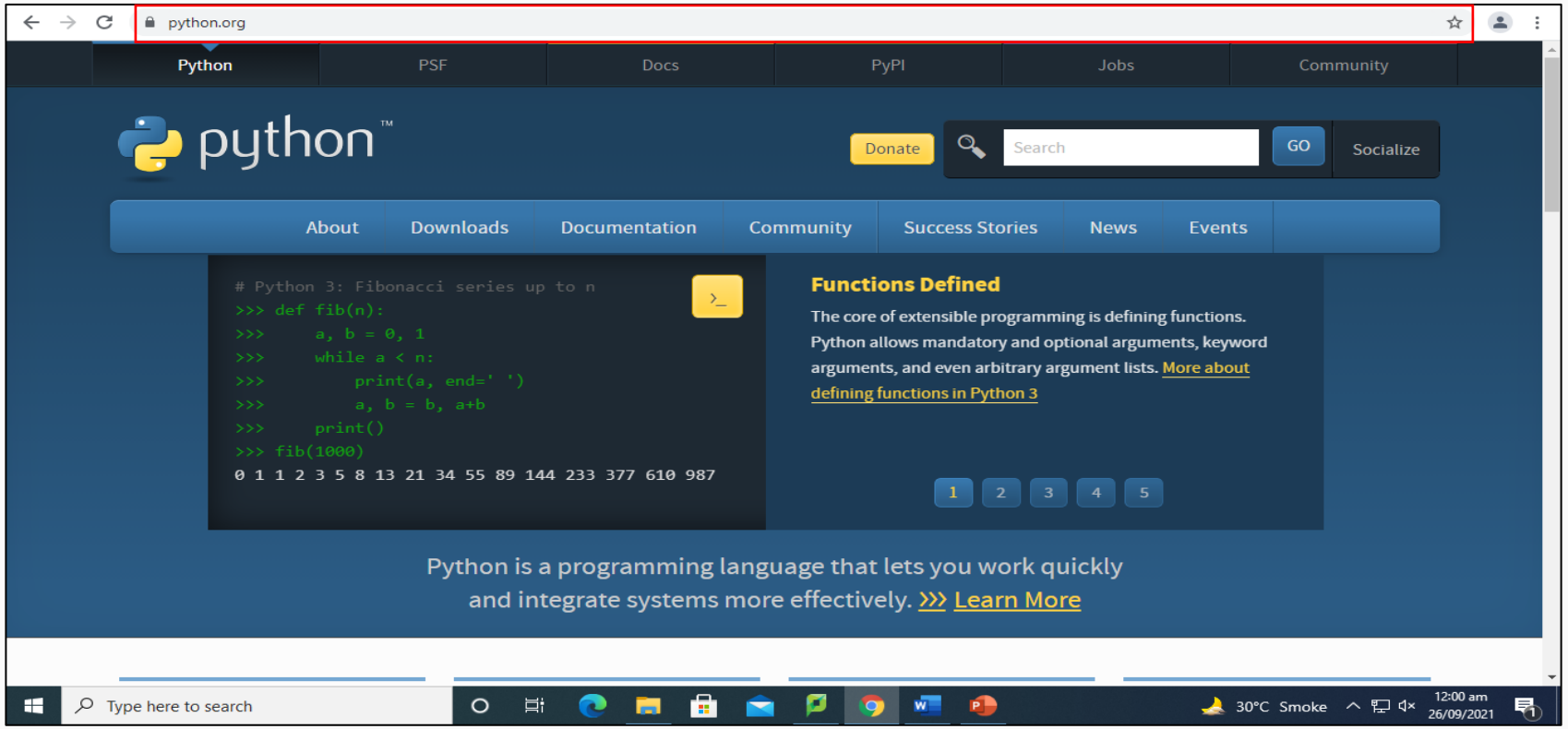
Python3 vs Python2

- Python2 is an outdated version of python. Python3 is the current version and is in-demand.
- In this course, we will be learning Python3 .



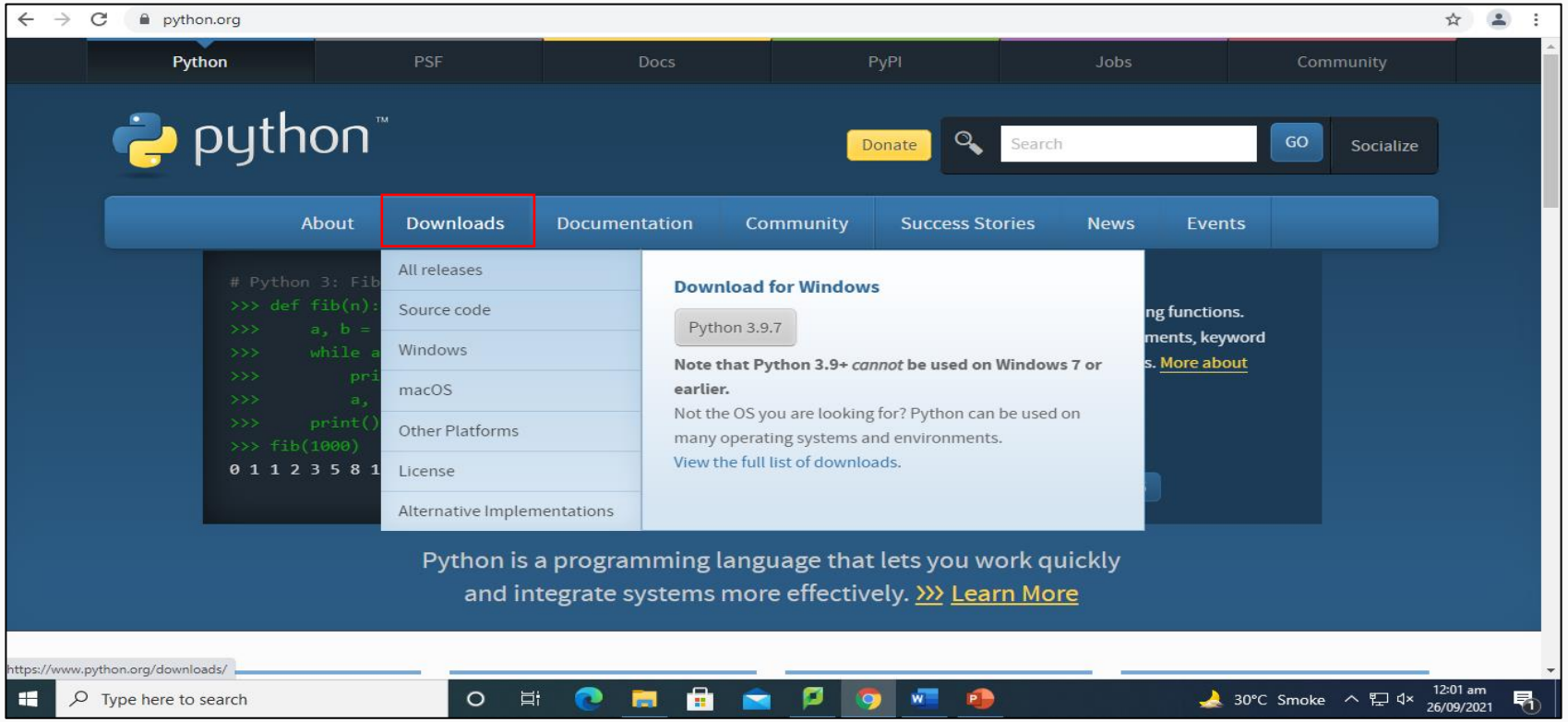
Setting Up Python – Windows (1/8)

- Visit: https://www.python.org/



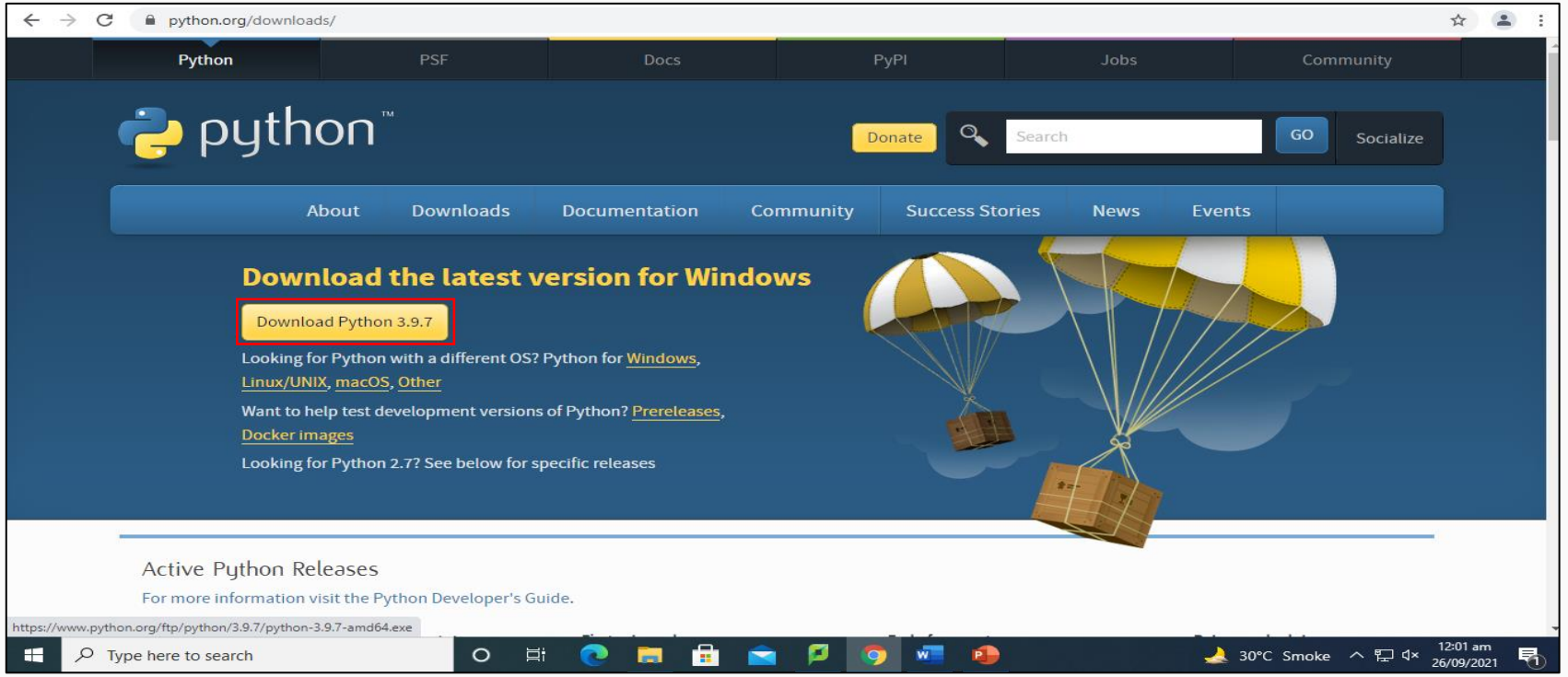
Setting Up Python – Windows (2/8)

- Click on Downloads.



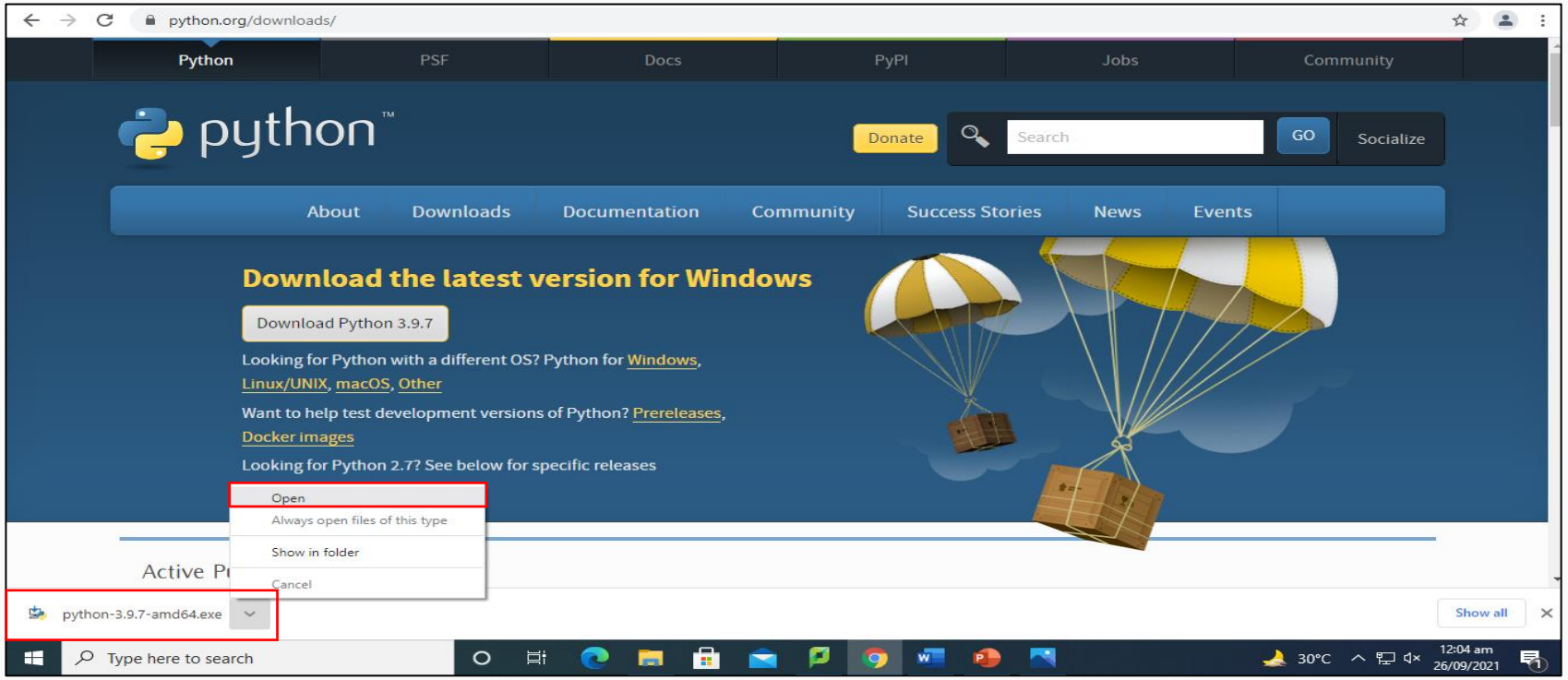
Setting Up Python – Windows (3/8)

- This will show you the latest version of Python3 for your operating system. Download it.



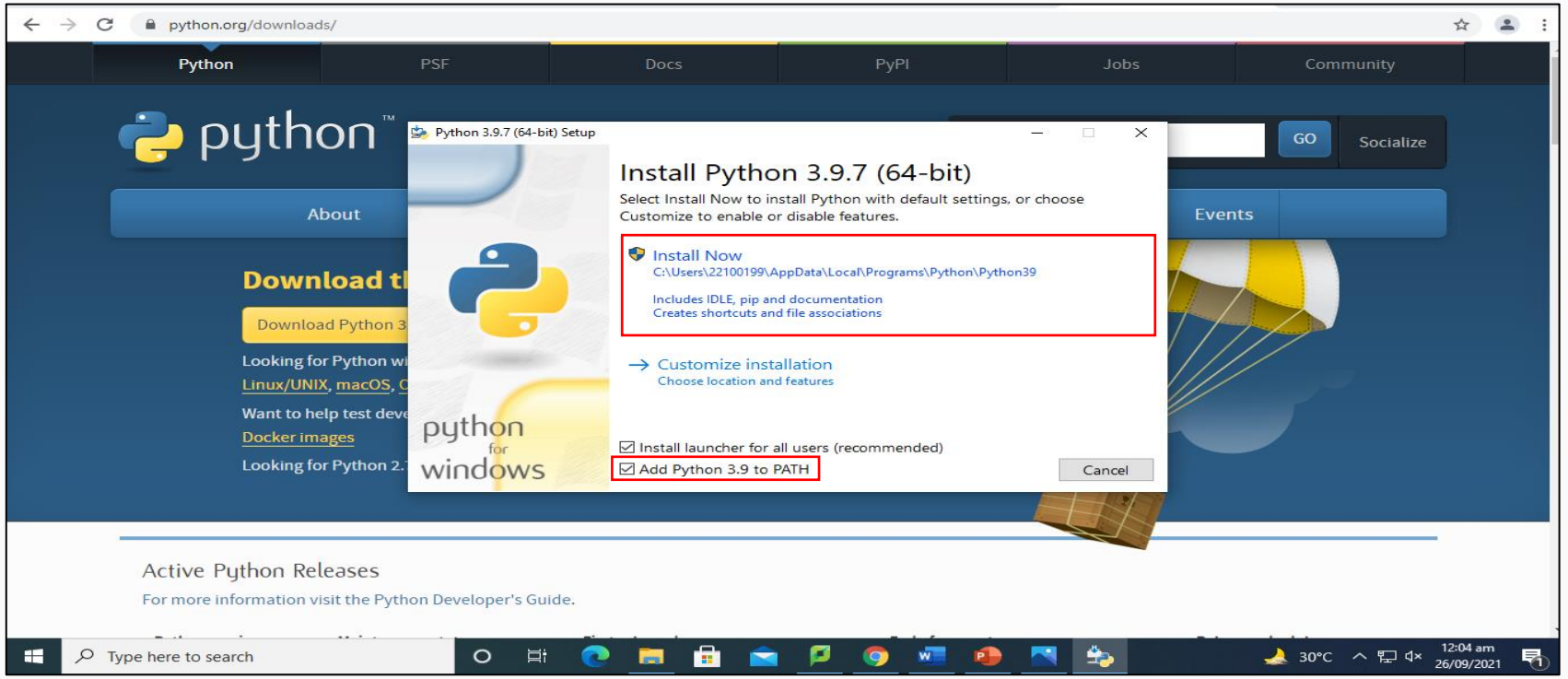
Setting Up Python – Windows (4/8)

- Once downloaded, click Open.



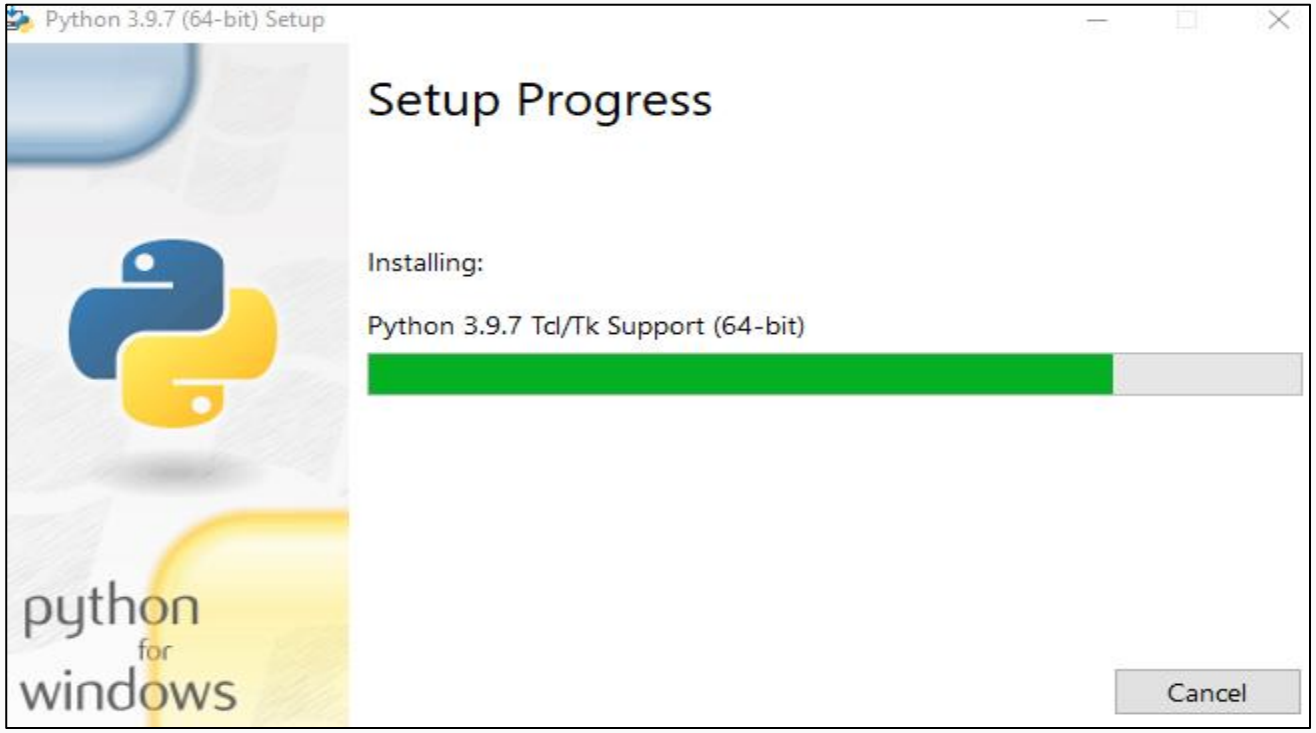
Setting Up Python – Windows (5/8)

- Installation wizard will open up. Make sure that ‘Add Python 3.9 to Path’ is checked. Click on Install Now.



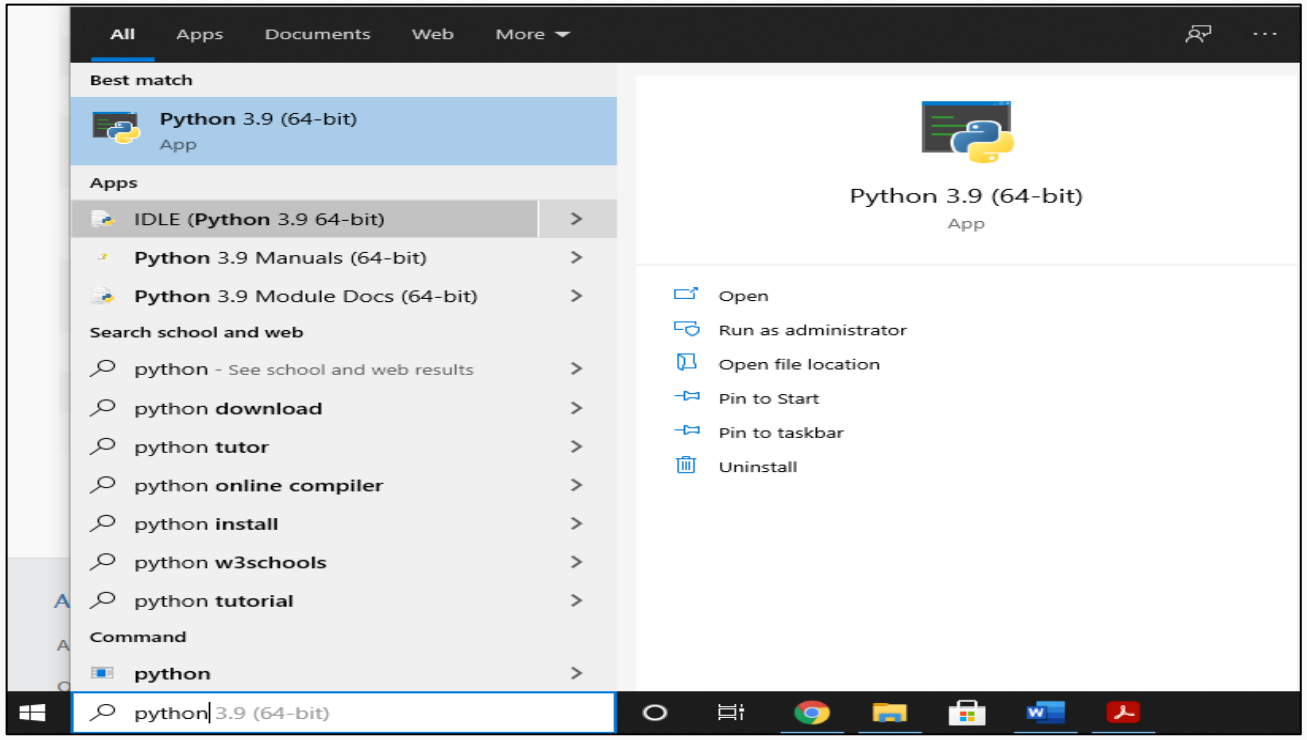
Setting Up Python – Windows (6/8)

- It will take sometime to install.



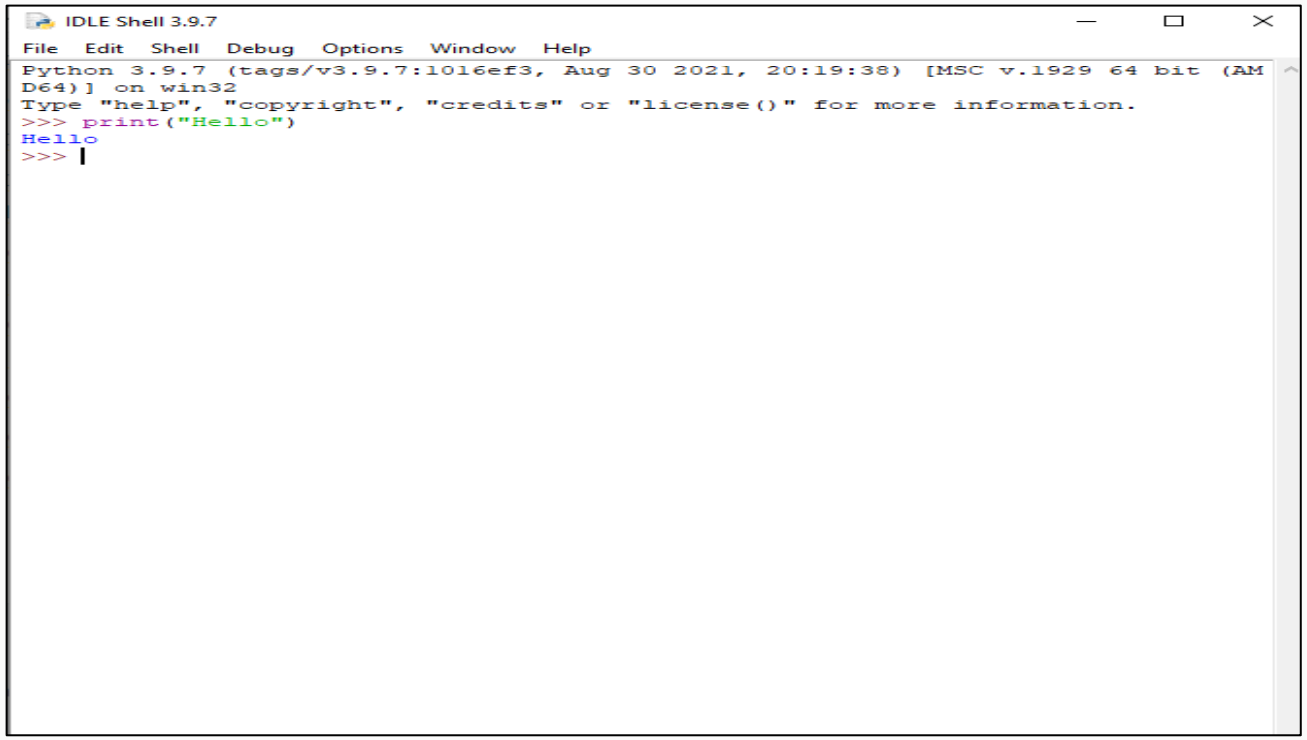
Setting Up Python – Windows (7/8)

- Once installed, type python in your search bar and click on ‘IDLE (Python 3.9 64-bit)’.



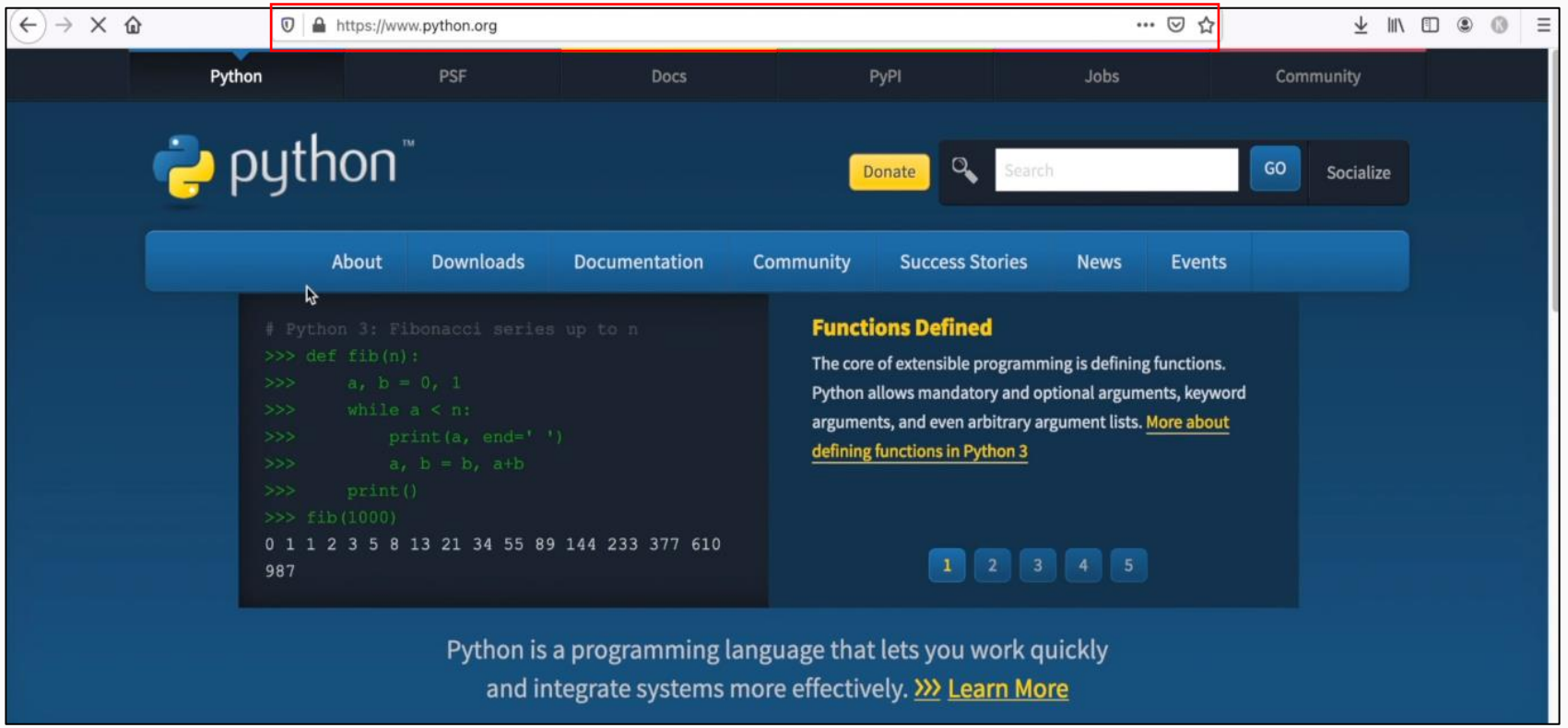
Setting Up Python – Windows (8/8)

- This will open up Python IDLE. Type print(“Hello”) and press Enter to execute.



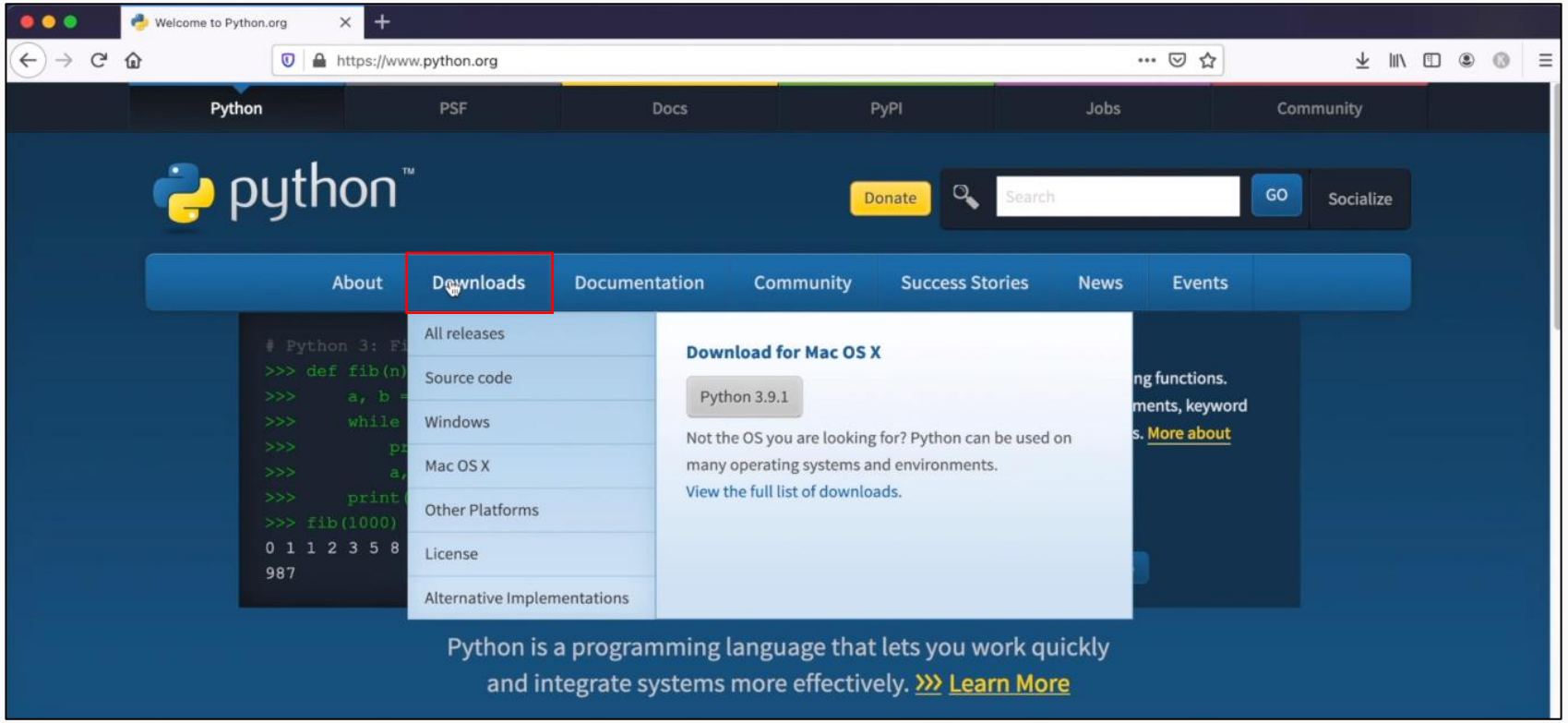
Setting Up Python – Mac (1/13)

- Visit: https://www.python.org/



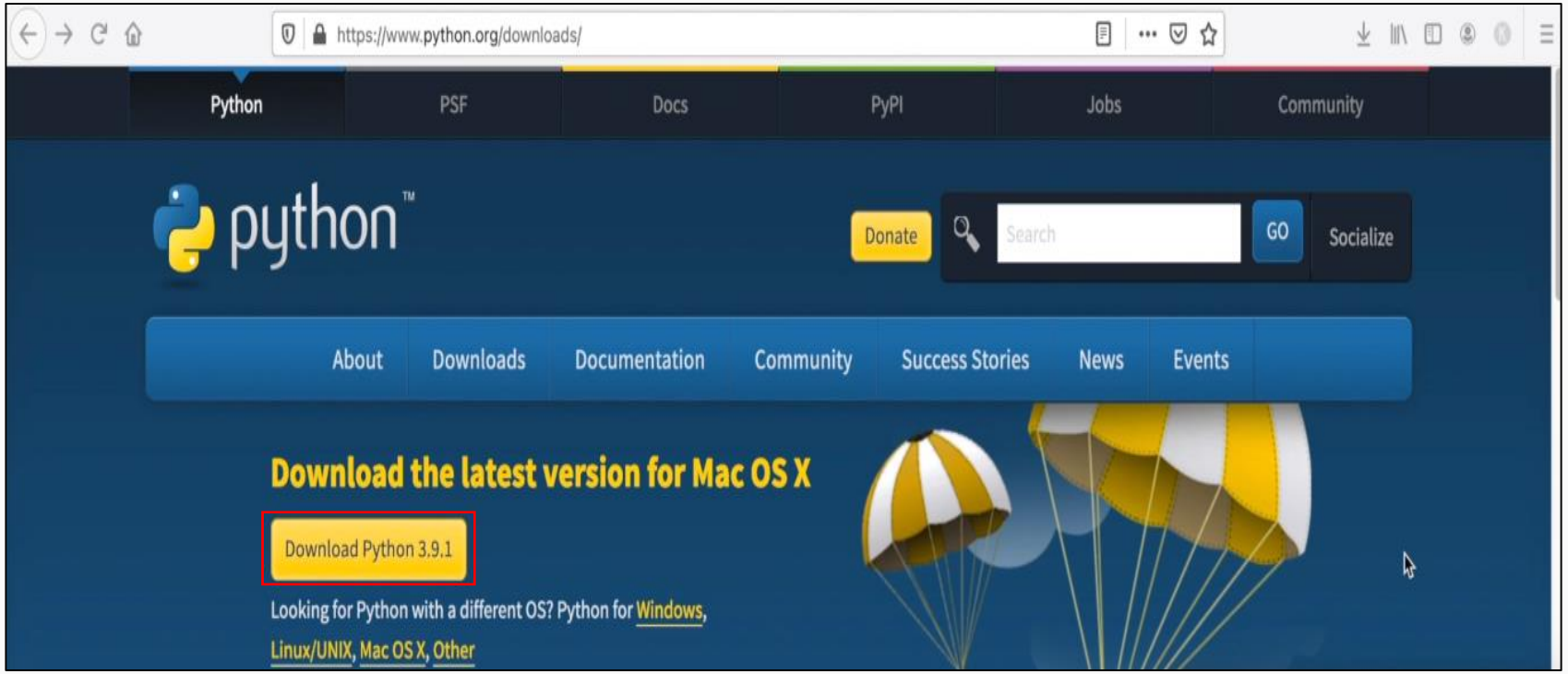
Setting Up Python – Mac (2/13)

- Click on Downloads.



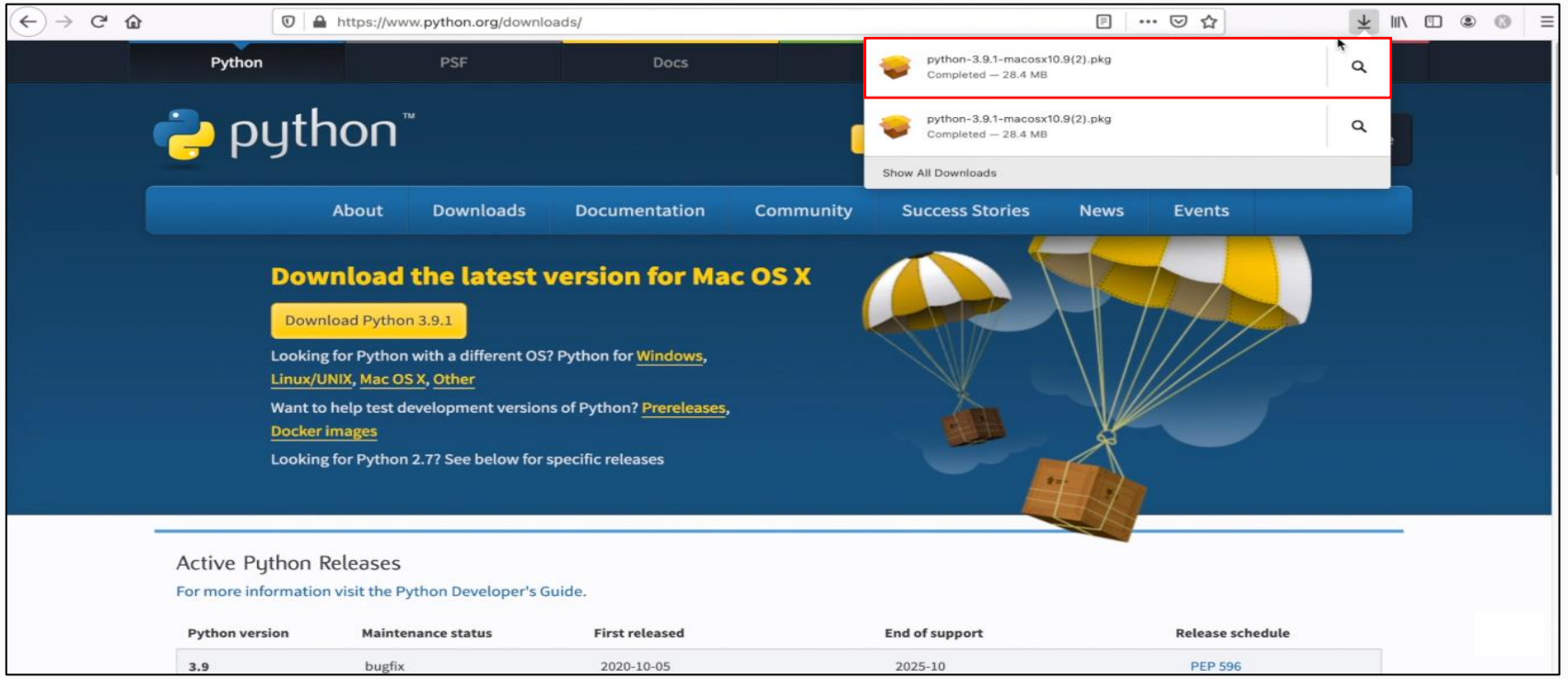
Setting Up Python – Mac (3/13)

- This will show you the latest version of Python3 for your operating system. Download it.



Setting Up Python – Mac (4/13)

- Once downloaded, click on the file.



Setting Up Python – Mac (5/13)

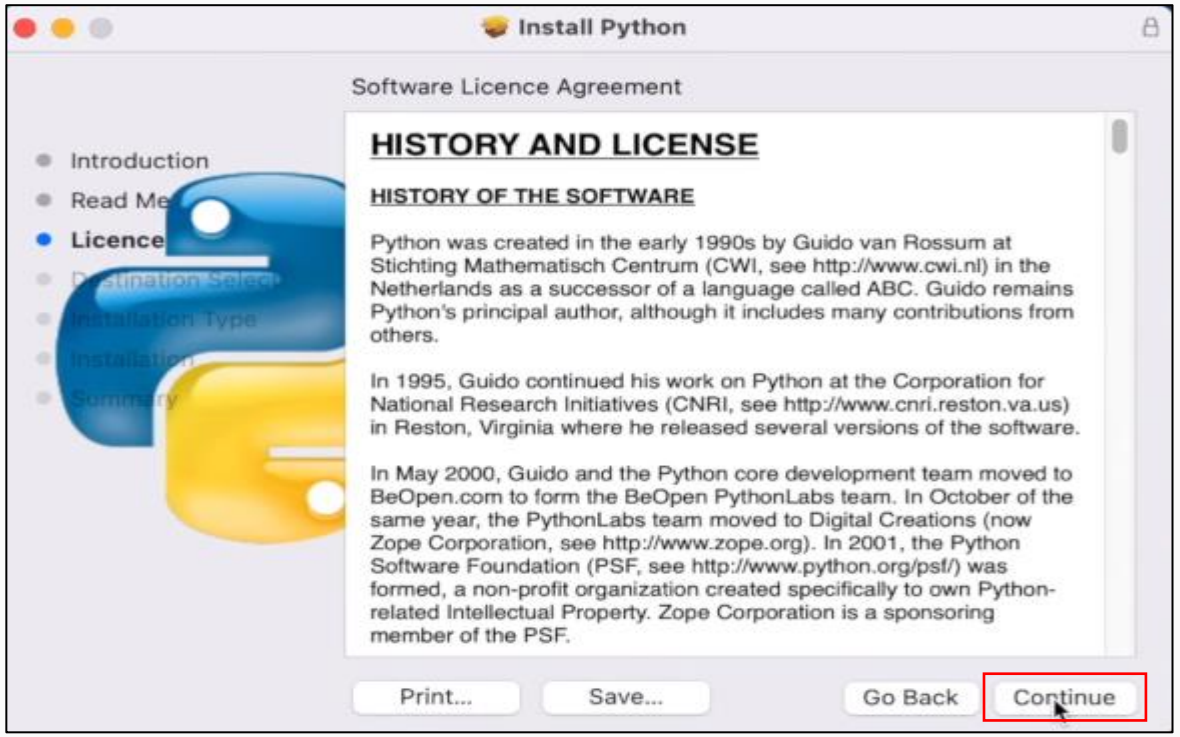
- Click continue.

Setting Up Python – Mac (6/13)

- Click continue.

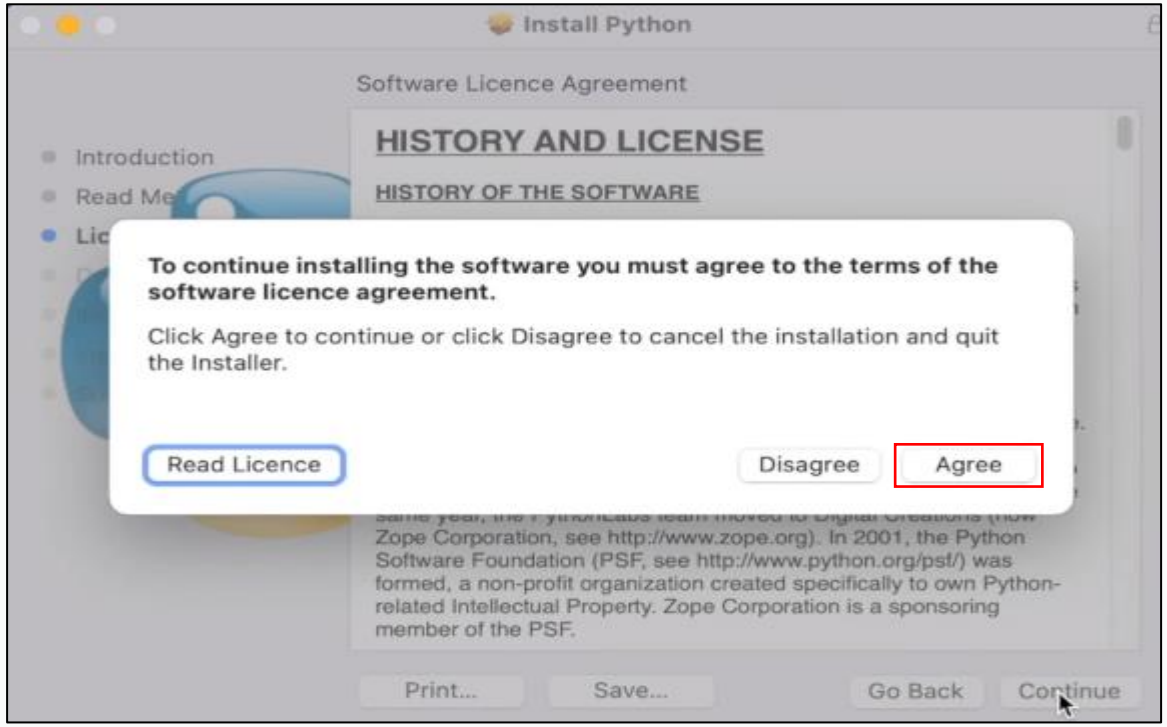
Setting Up Python – Mac (7/13)

- Click continue.



Setting Up Python – Mac (8/13)

- Read and accept the license agreement.



Setting Up Python – Mac (9/13)

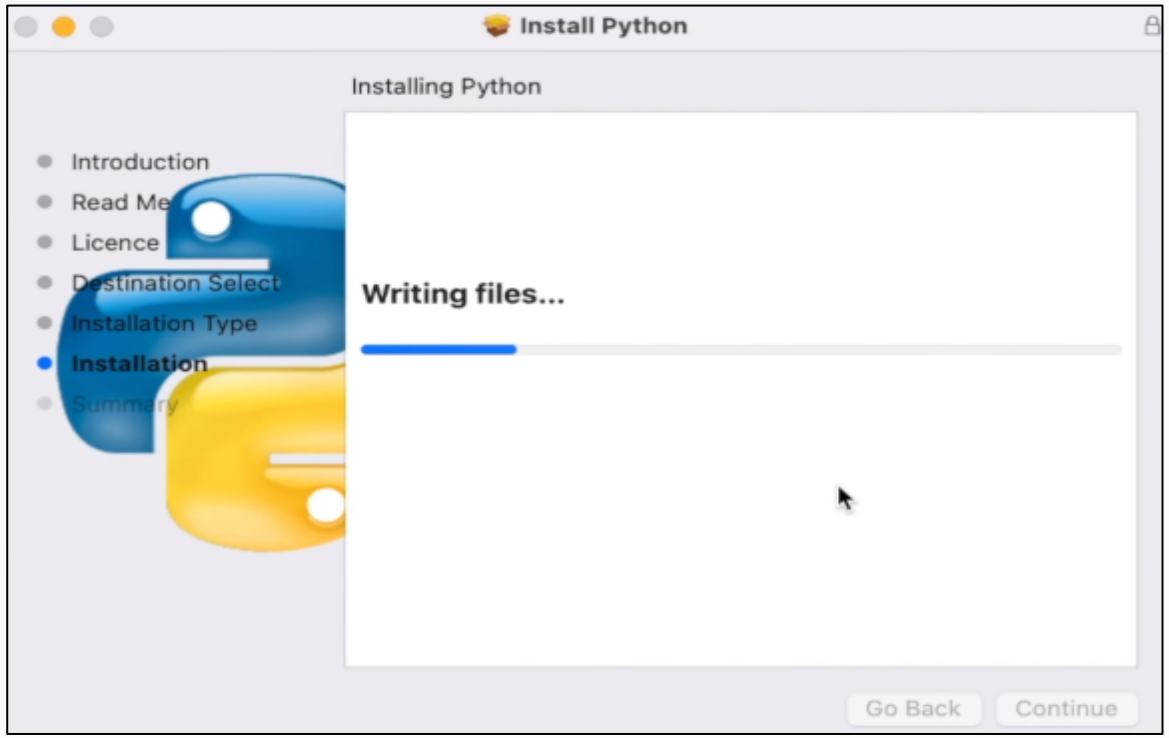
- Click continue.

Setting Up Python – Mac (10/13)

- Click install. If prompted, enter your password.

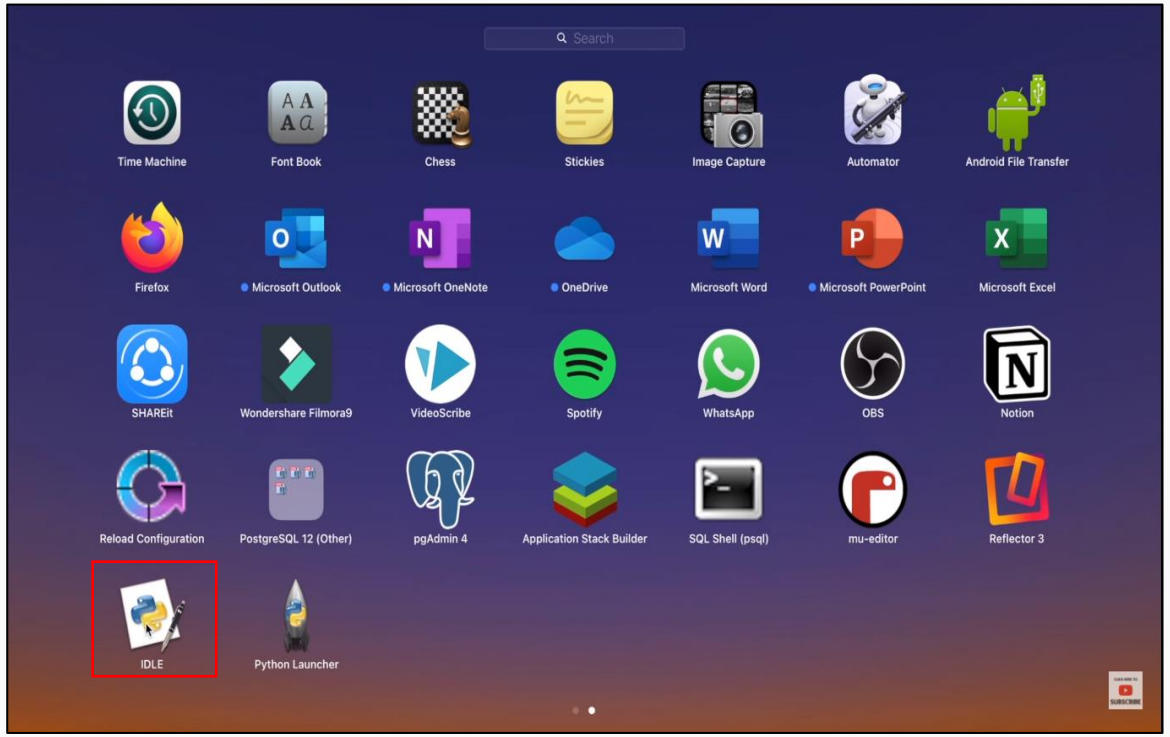
Setting Up Python – Mac (11/13)

- Wait for the installation to finish.



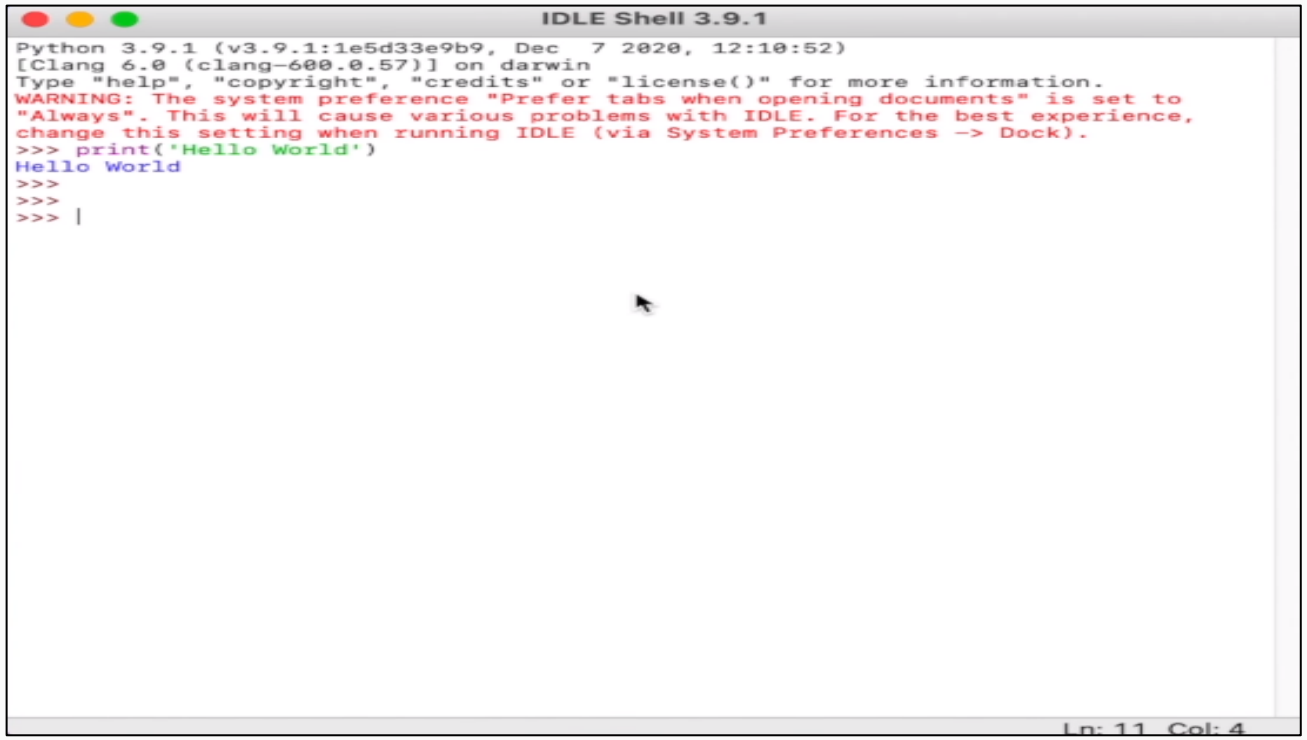
Setting Up Python – Mac (12/13)

- Once installed, search for IDLE in your spotlight and click on it.



Setting Up Python – Mac (13/13)

- This will open up Python IDLE. Type print(“Hello World”) and press Enter to execute.



Jupyter Notebook

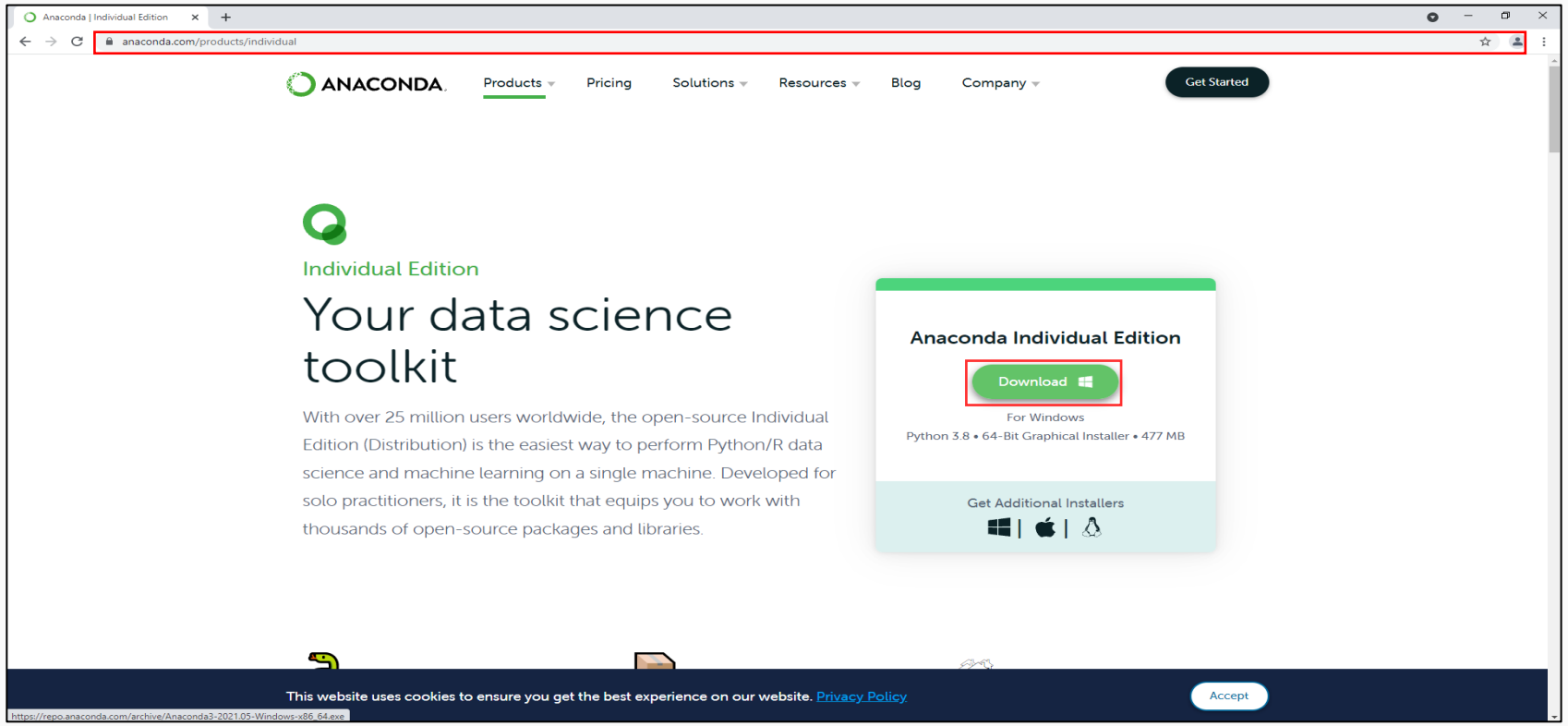
- Jupyter Notebook is a web application that allows you to create documents that may contain code, visualizations, and narrative text.
- It is very popular and is used extensively in data science, artificial intelligence, and machine learning.
- A notebook has the .ipynb extension

Alternatives to Jupyter Notebook

- Google Colaboratory is an alternative to Jupyter Notebook and does not require downloading anything.

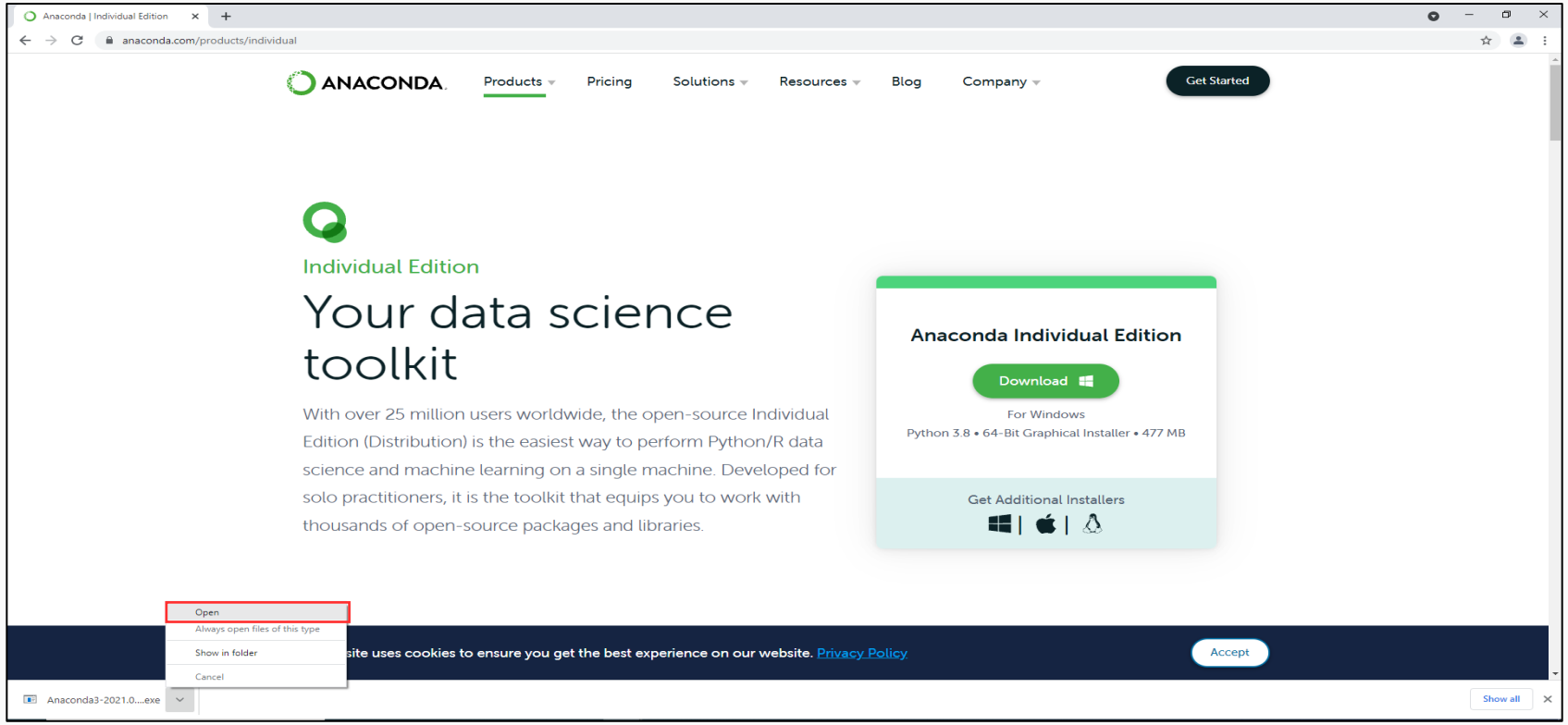
Anaconda for Windows (1/13)

- Visit: https://www.anaconda.com/products/individual
- Click on the Download button.



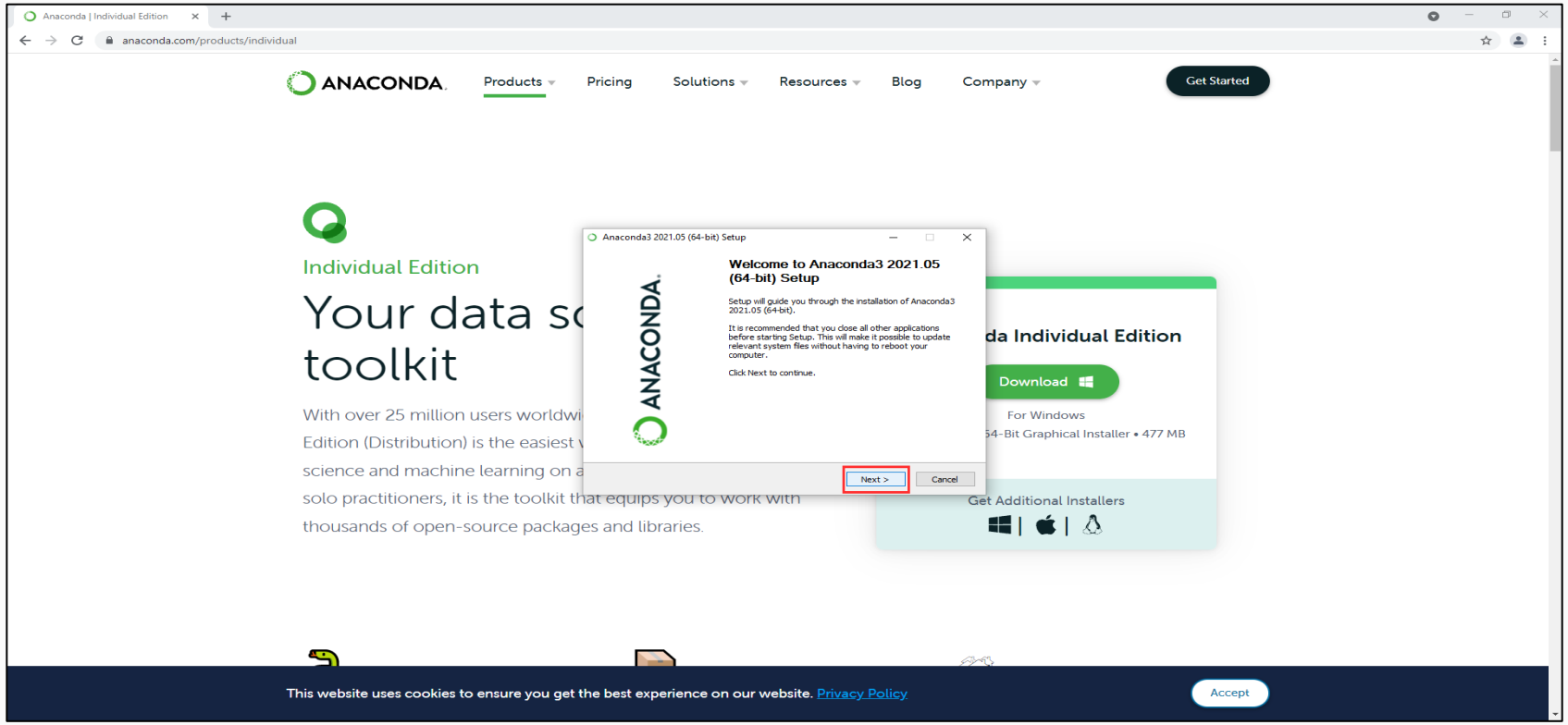
Anaconda for Windows (2/13)

- Once Downloaded, click on Open.



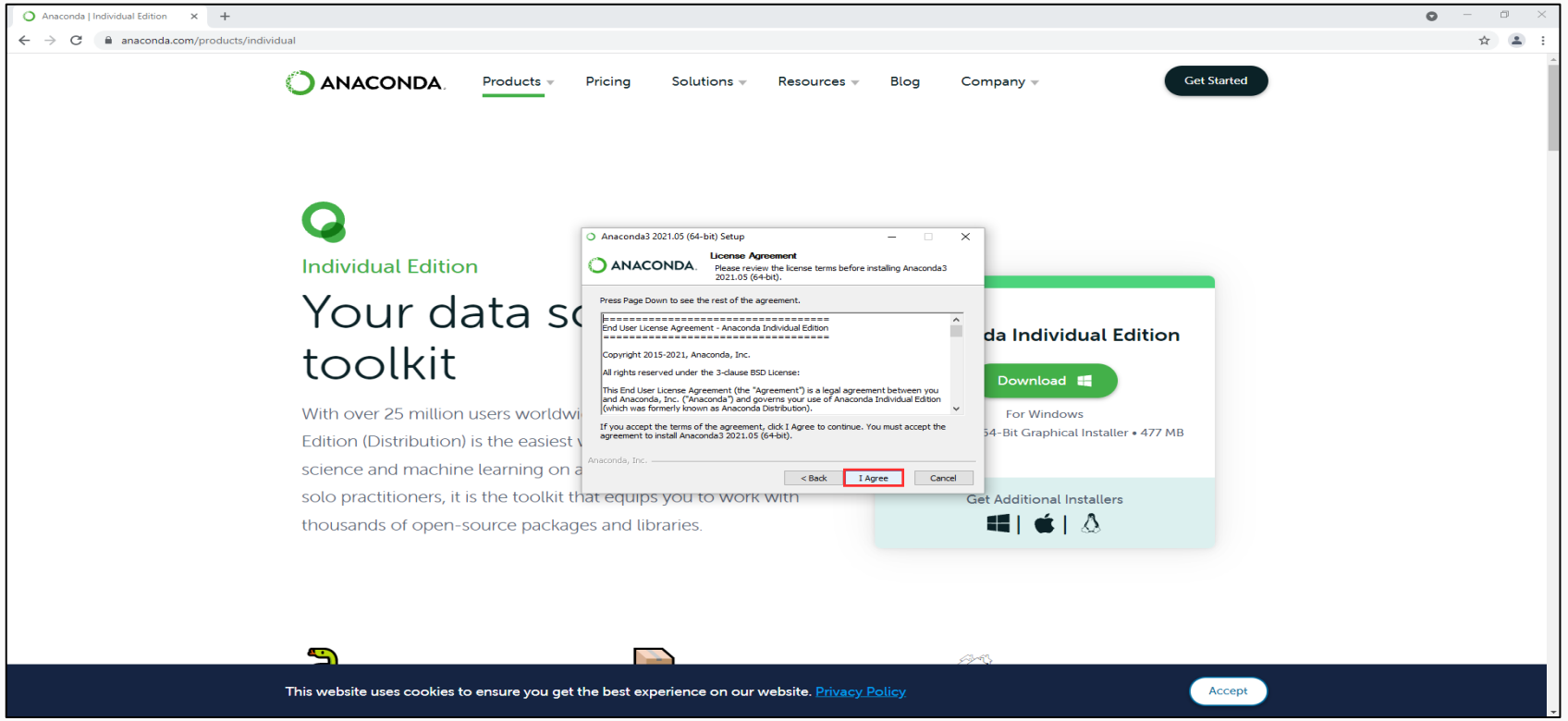
Anaconda for Windows (3/13)

- An installation Wizard will open up.
- Click on Next.



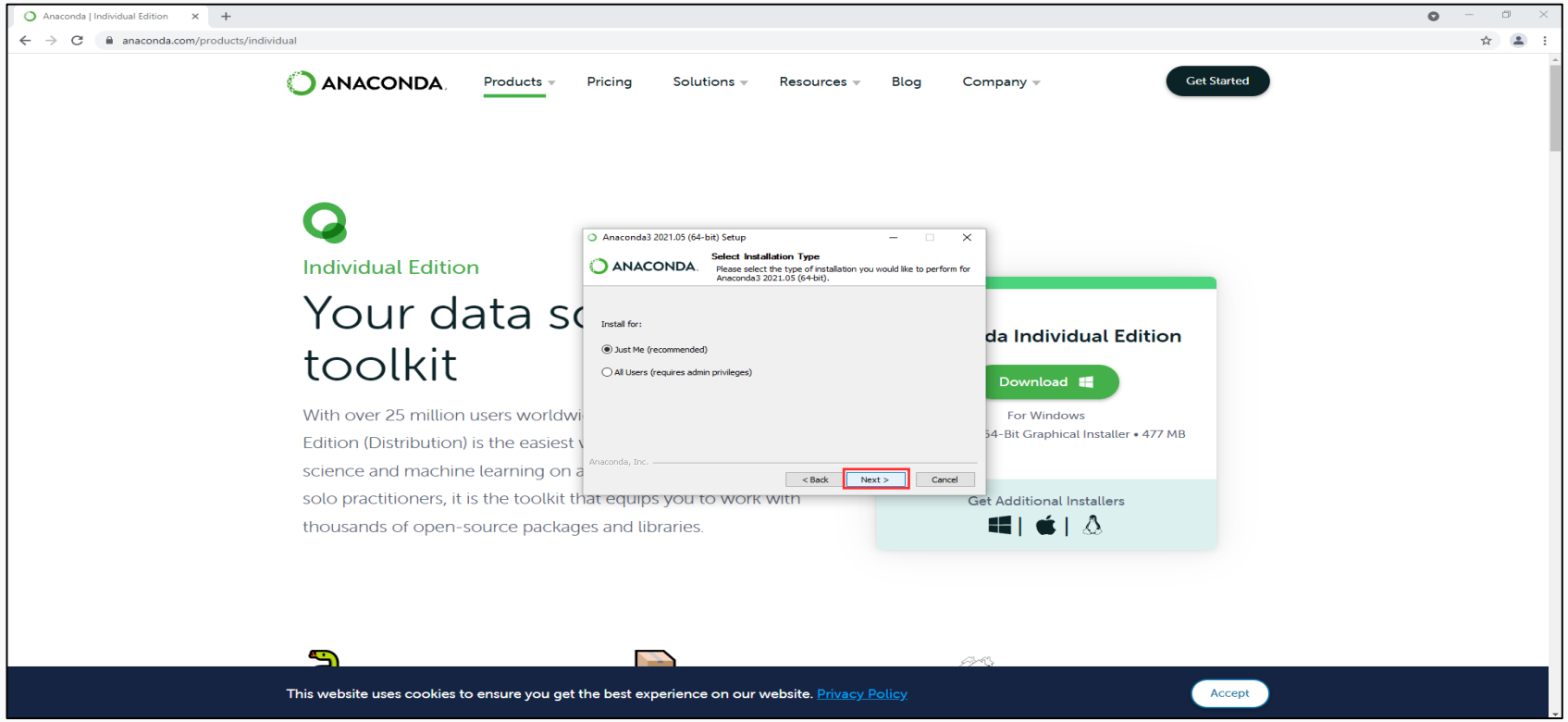
Anaconda for Windows (4/13)

- Read the License Agreement and click I Agree.



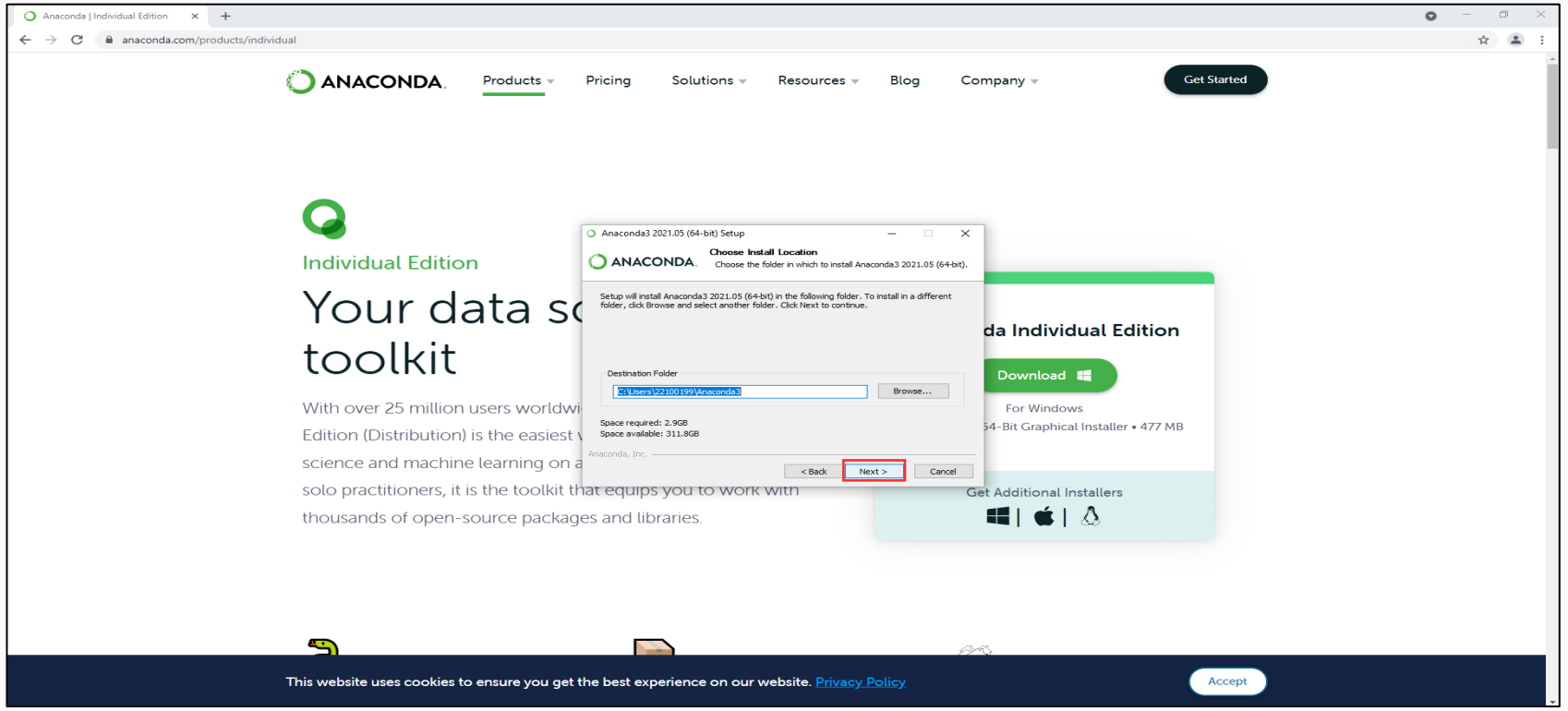
Anaconda for Windows (5/13)

- Click Next.



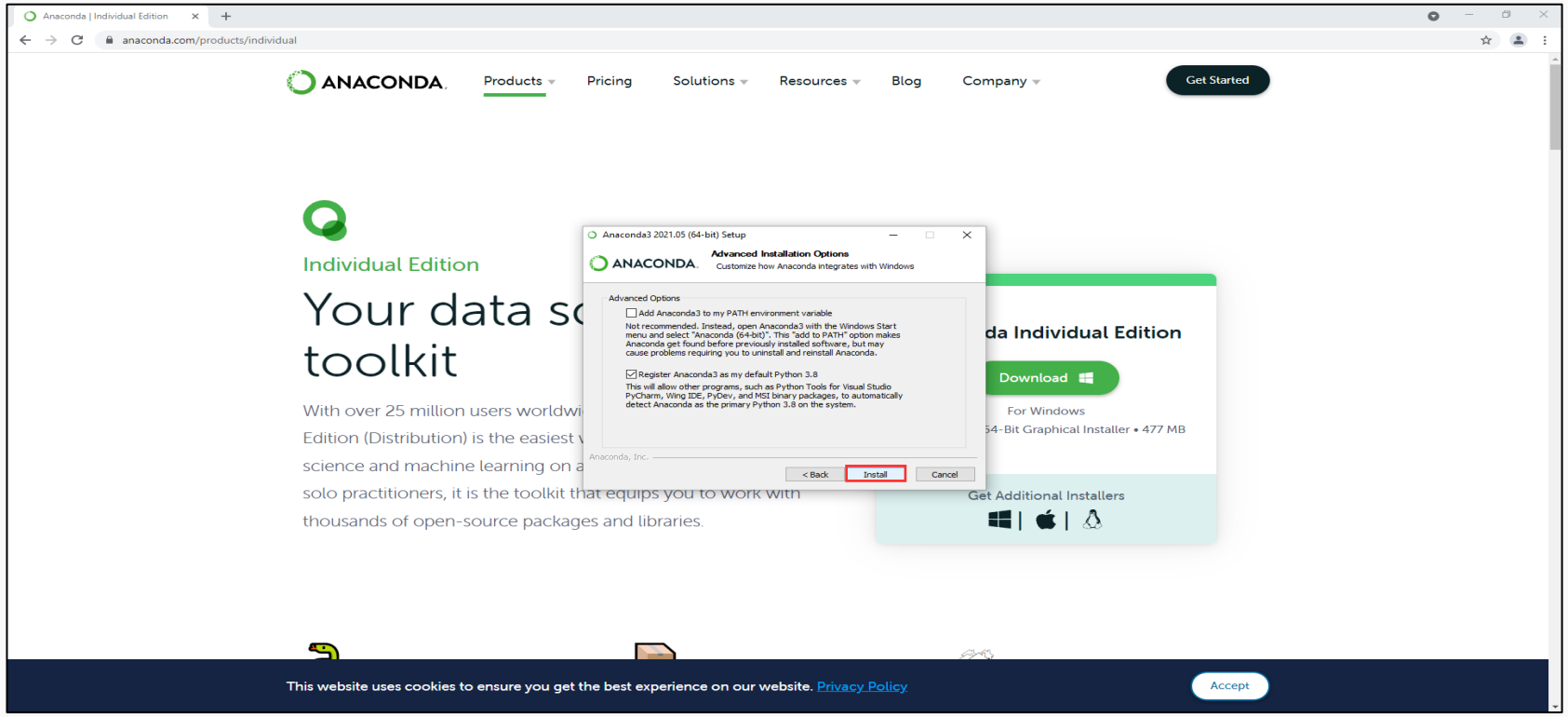
Anaconda for Windows (6/13)

- Choose the destination and click Next.



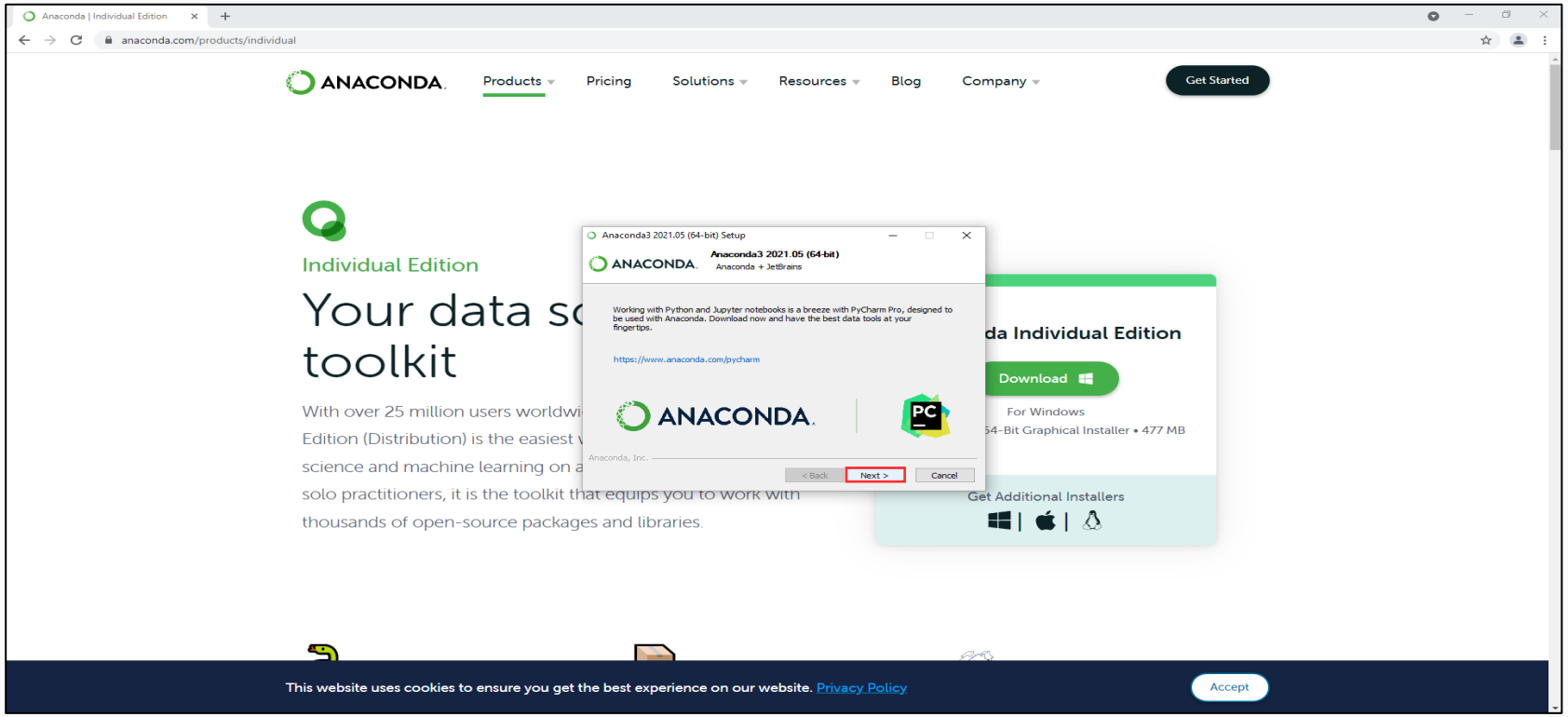
Anaconda for Windows (7/13)

- Click Install.



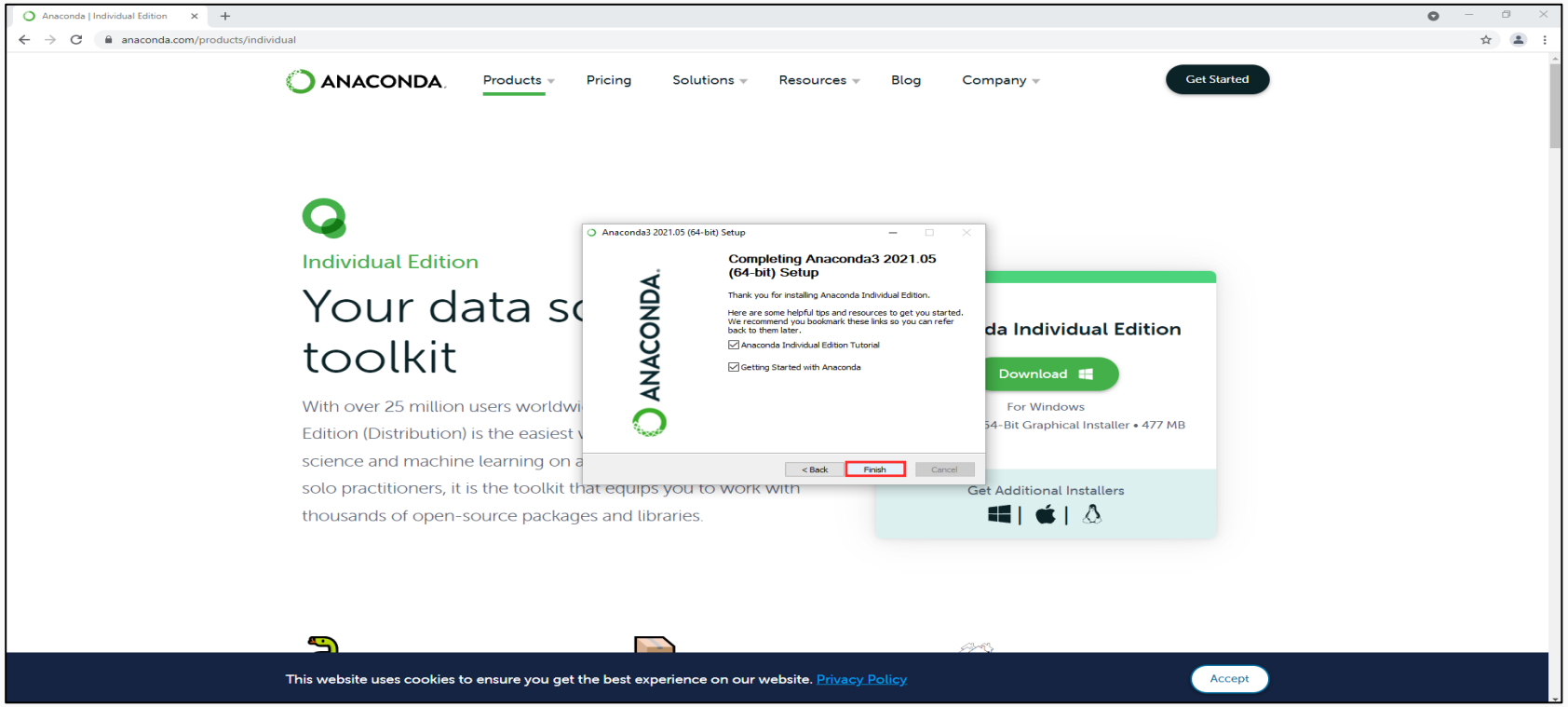
Anaconda for Windows (8/13)

- Wait for the installation to finish and click Next.



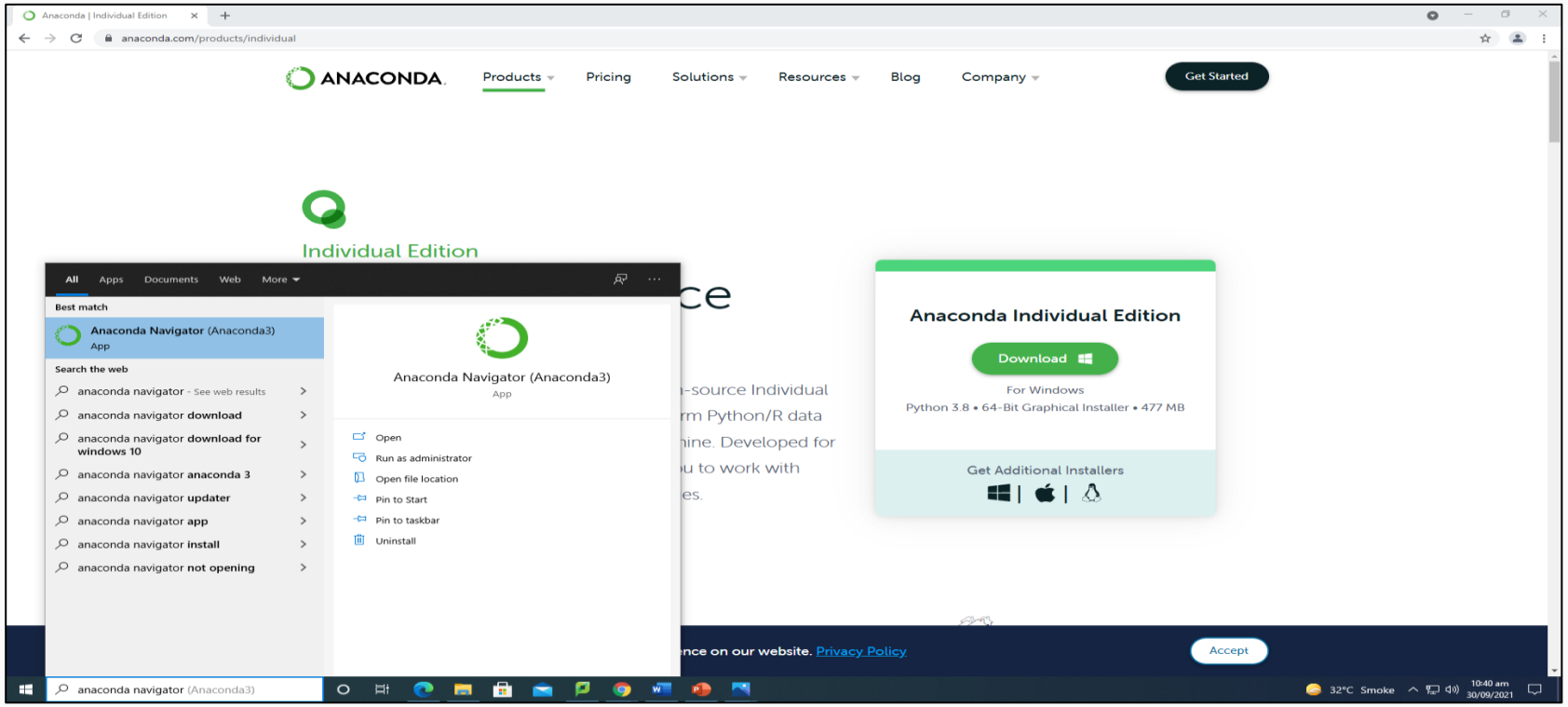
Anaconda for Windows (9/13)

- Click Finish.



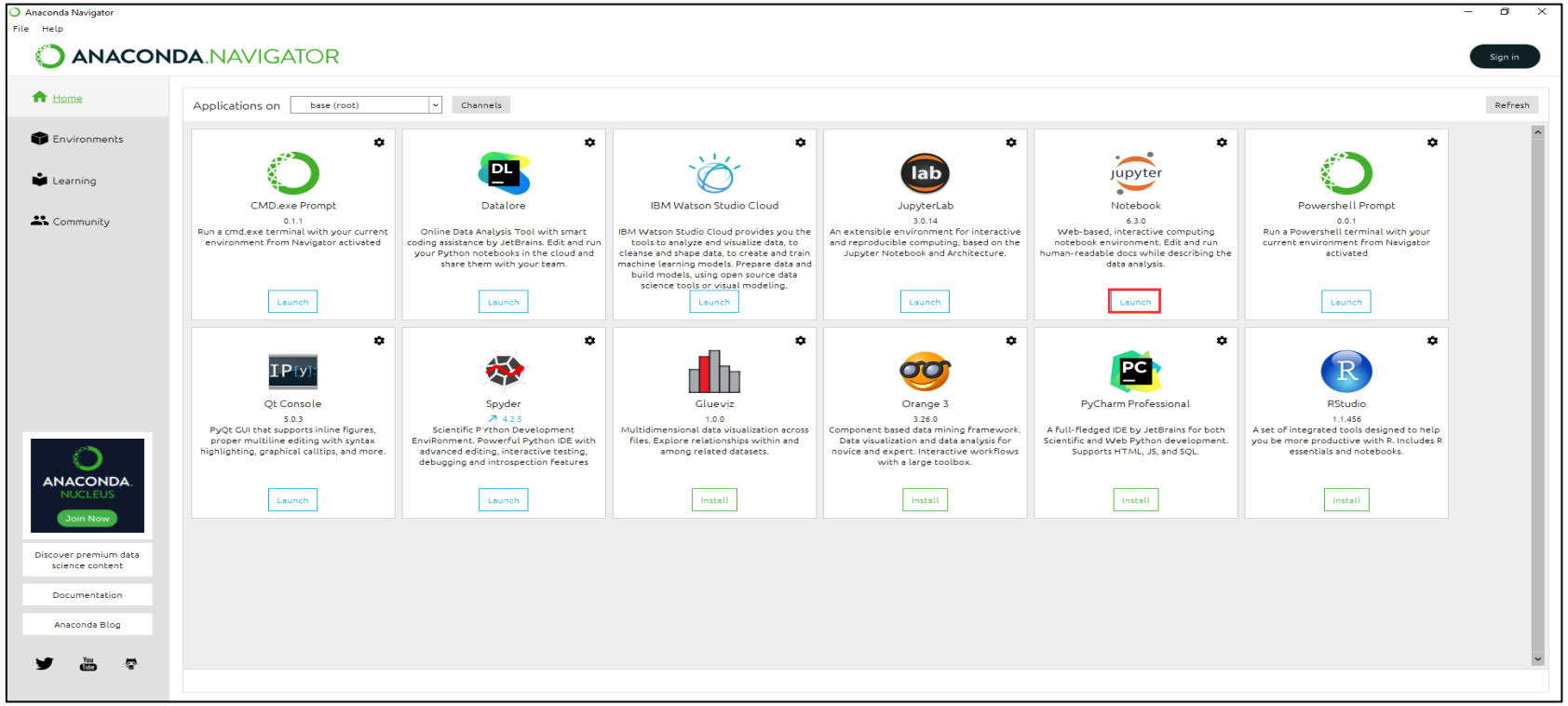
Anaconda for Windows (10/13)

- To launch Jupyter Notebook, search for anaconda navigator and open it.



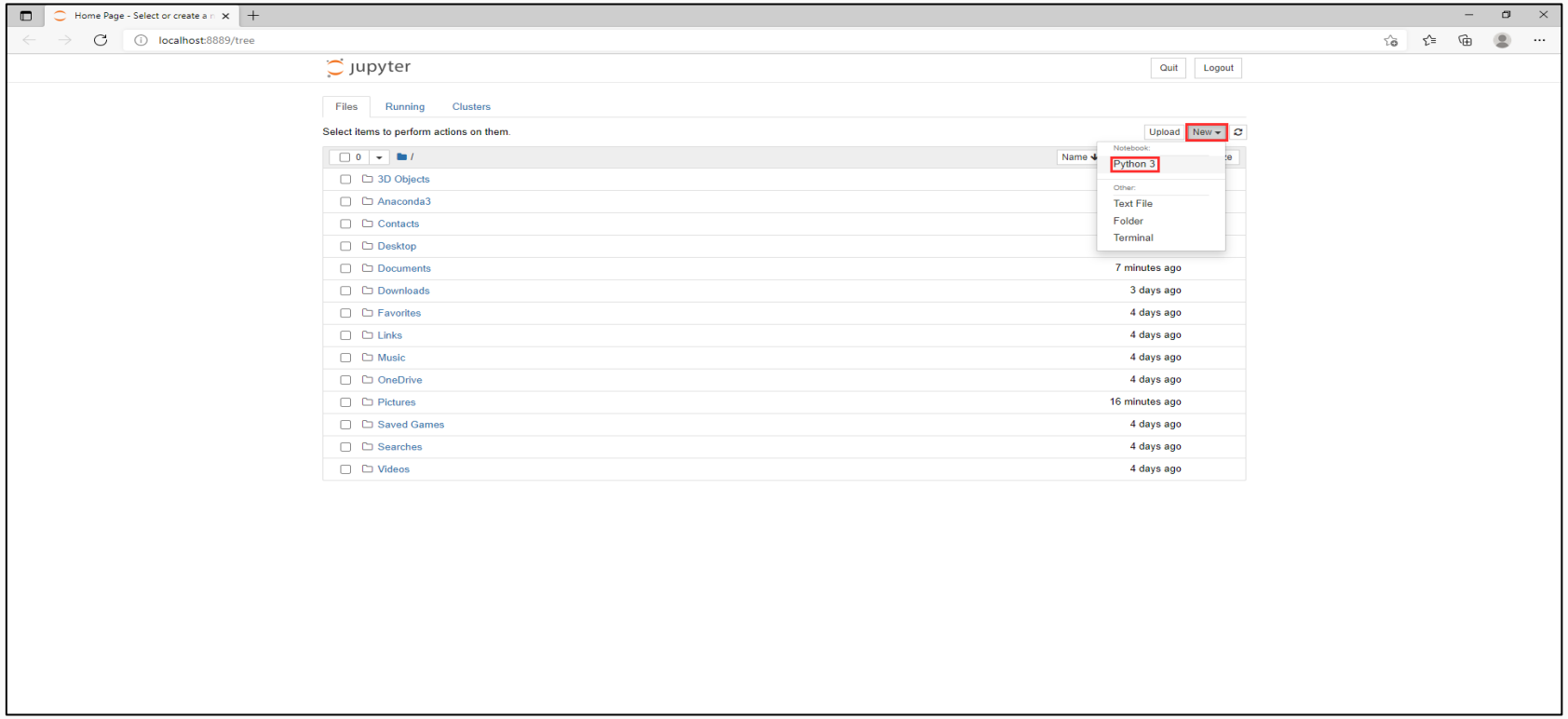
Anaconda for Windows (11/13)

- Following screen will open up.
- Launch Jupyter Notebook from here.



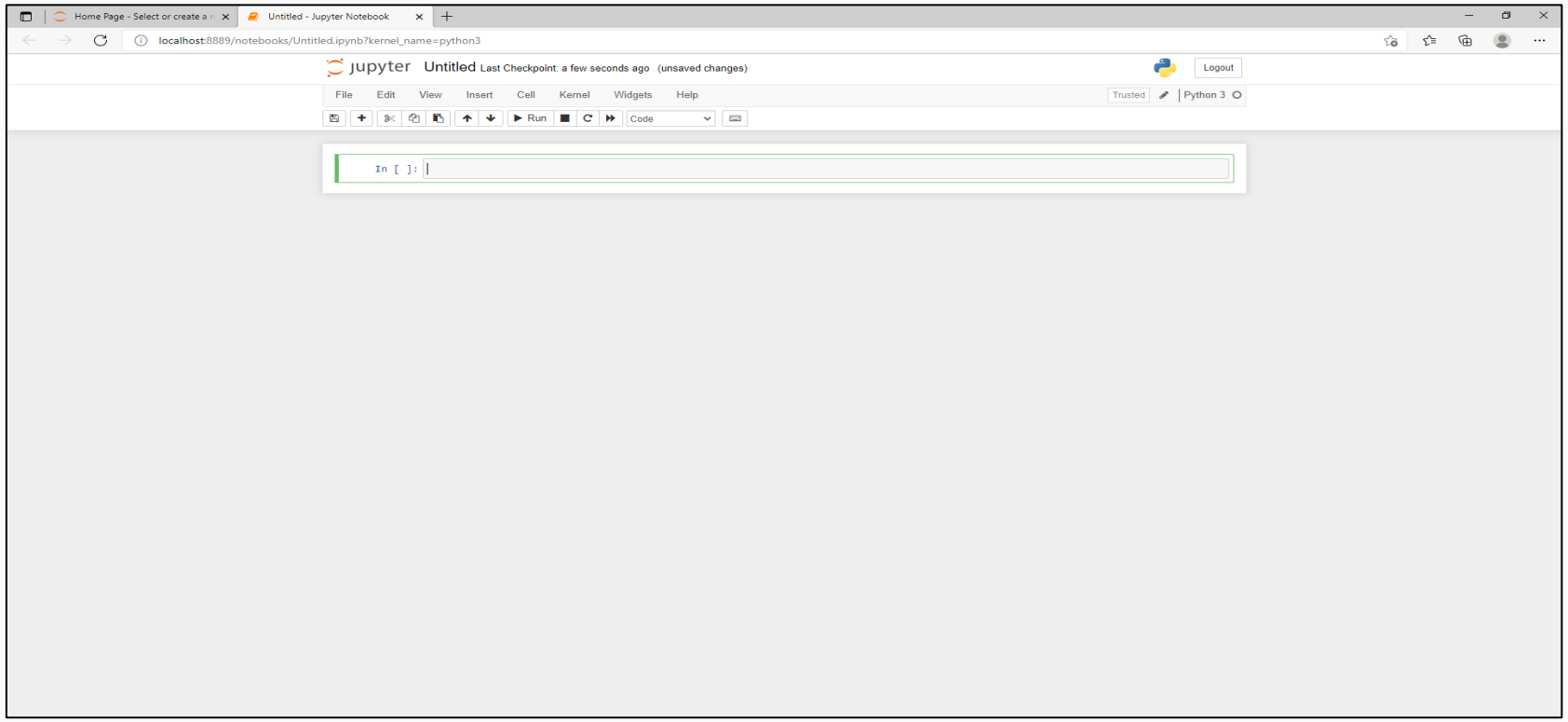
Anaconda for Windows (12/13)

- This will launch Jupyter Notebook in your default browser.
- Click on New -> Python3 to create a new Notebook.



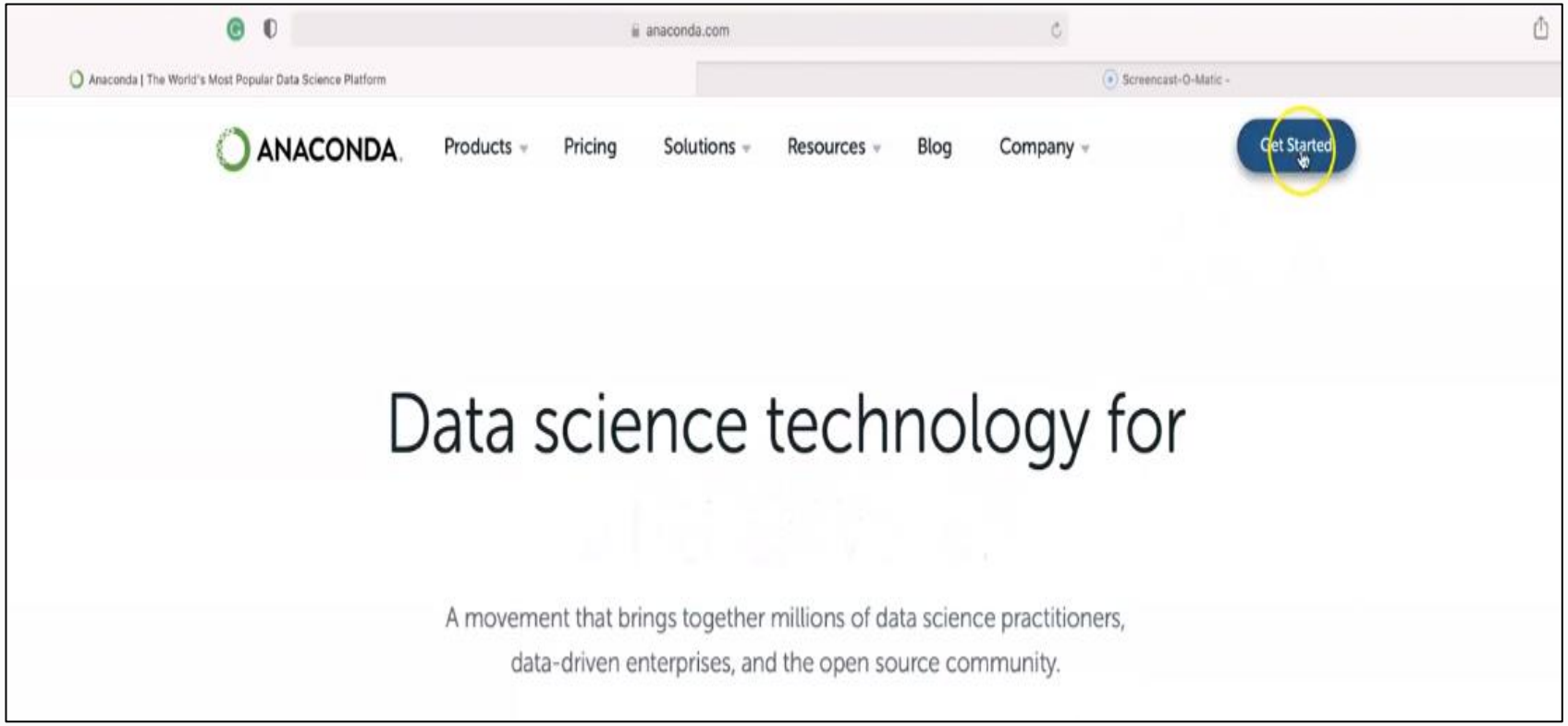
Anaconda for Windows (13/13)

- This is how a Notebook looks like.



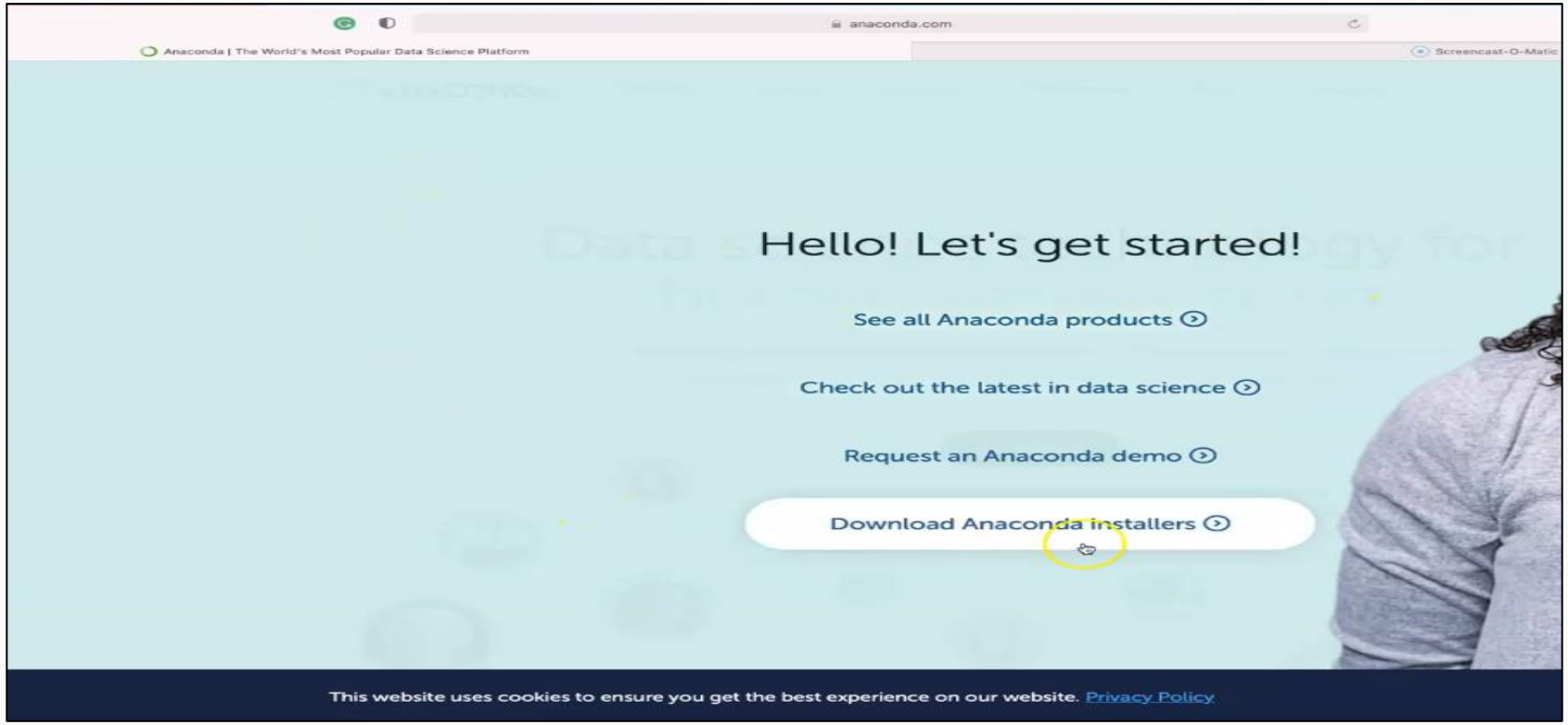
Anaconda for Mac (1/14)

- Visit: https://www.anaconda.com/
- Click on Get Started.



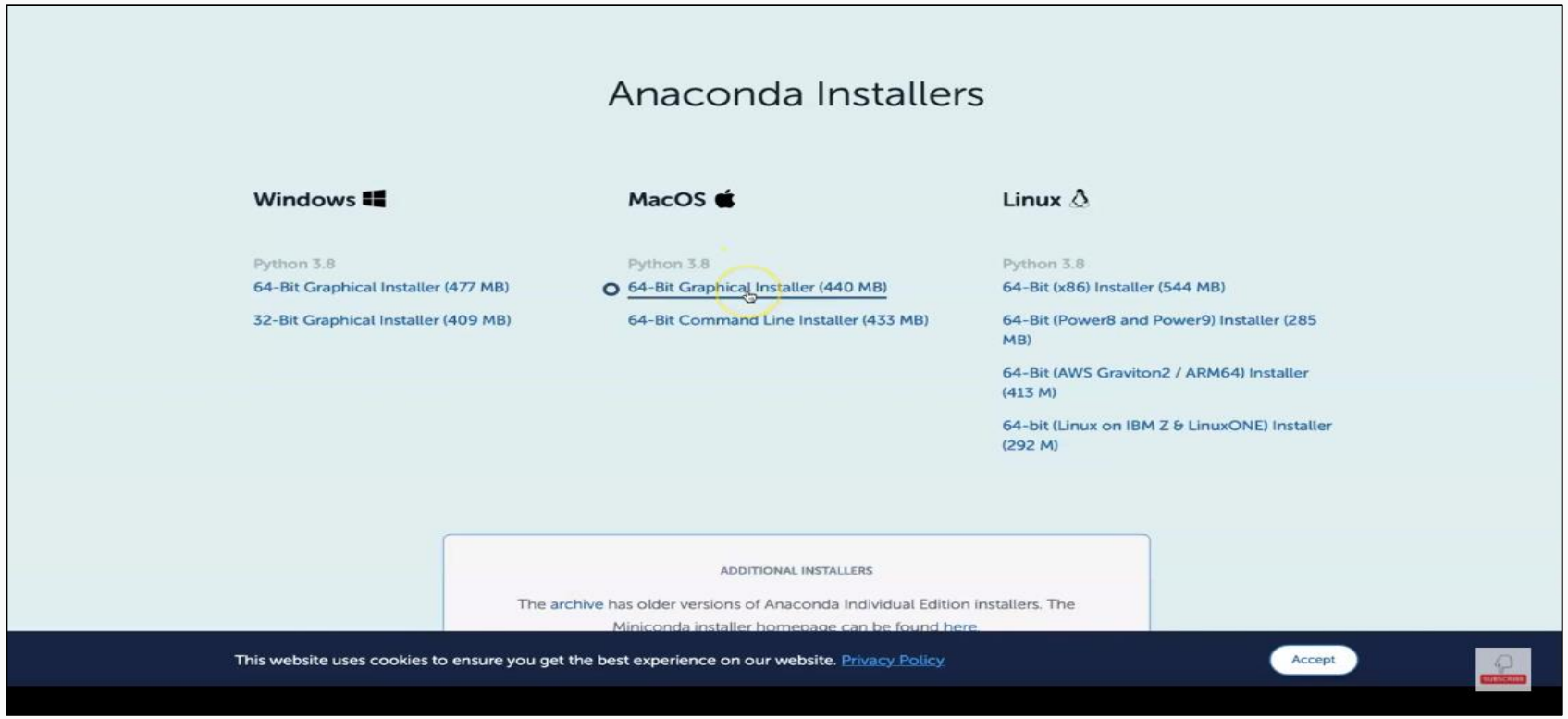
Anaconda for Mac (2/14)

- Click Download Anaconda Installers.



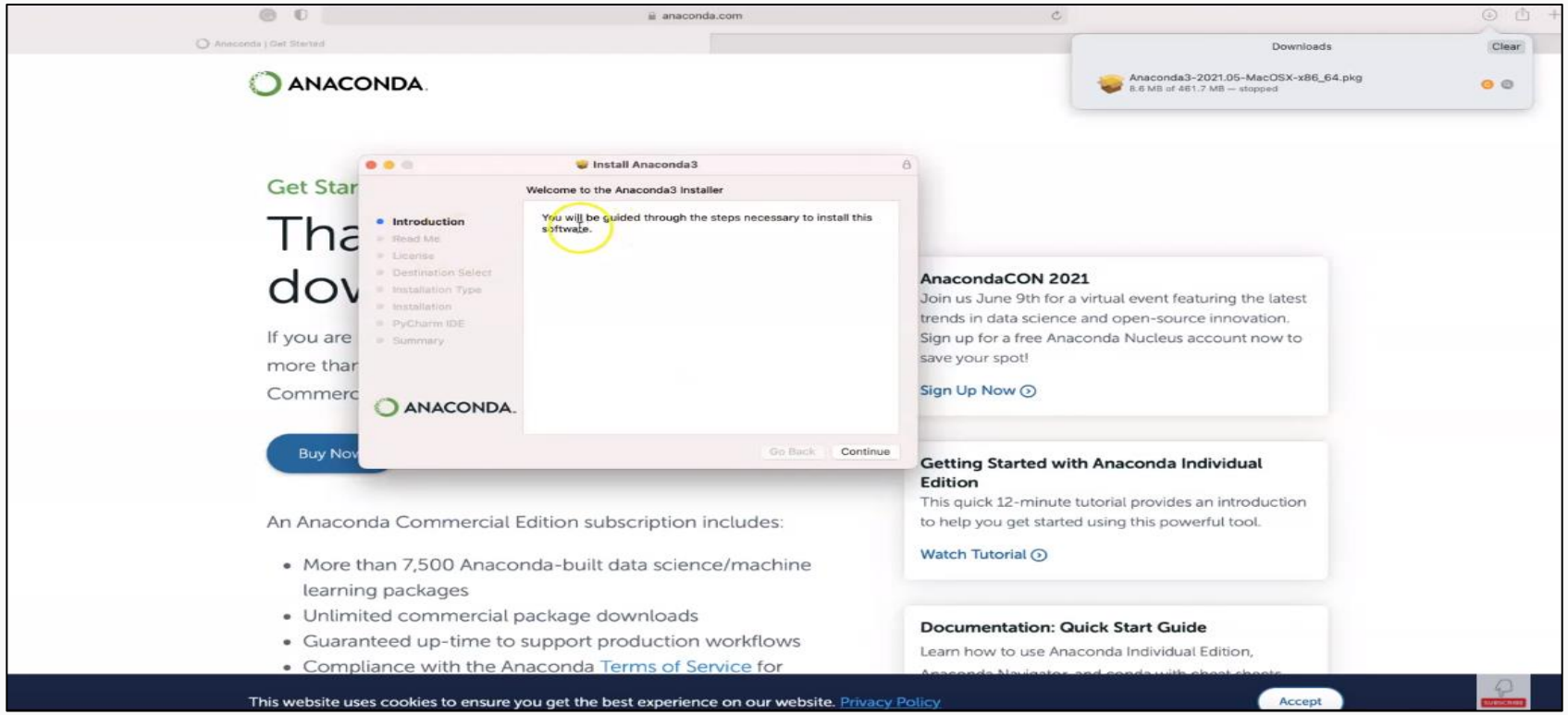
Anaconda for Mac (3/14)

- Download the graphical installer for Mac.



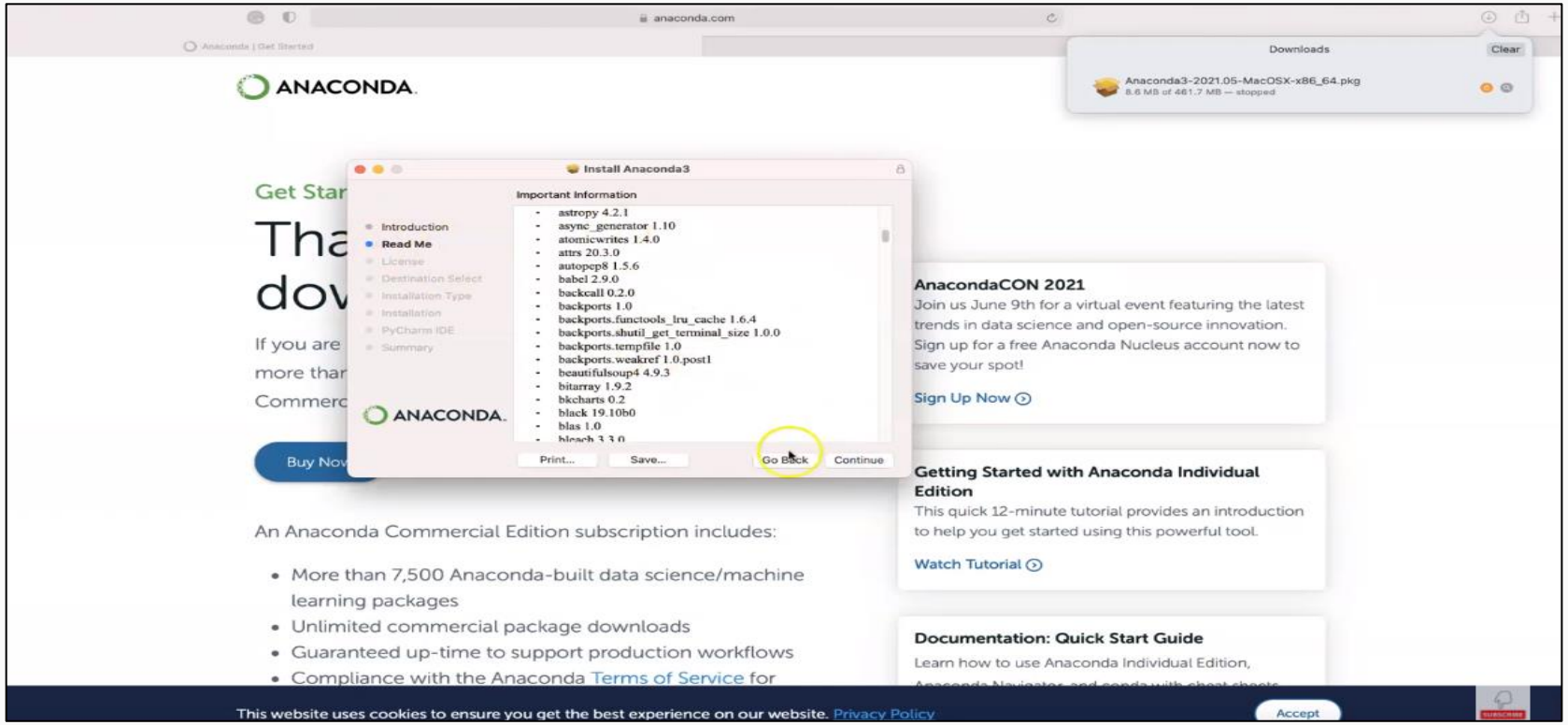
Anaconda for Mac (4/14)

- Once download, launch the installation wizard by double clicking on the file.
- Click Continue.



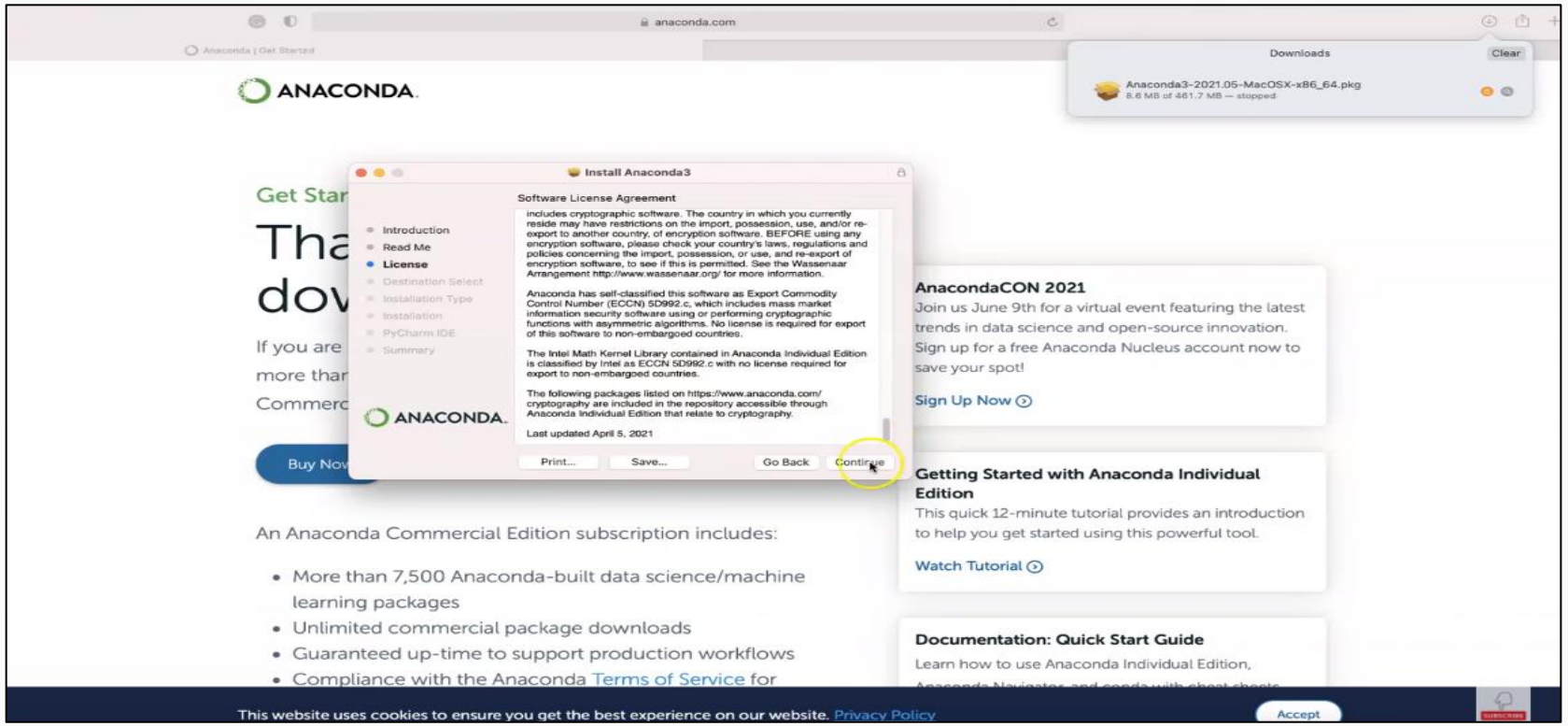
Anaconda for Mac (5/14)

- Read the Read Me file and click continue.



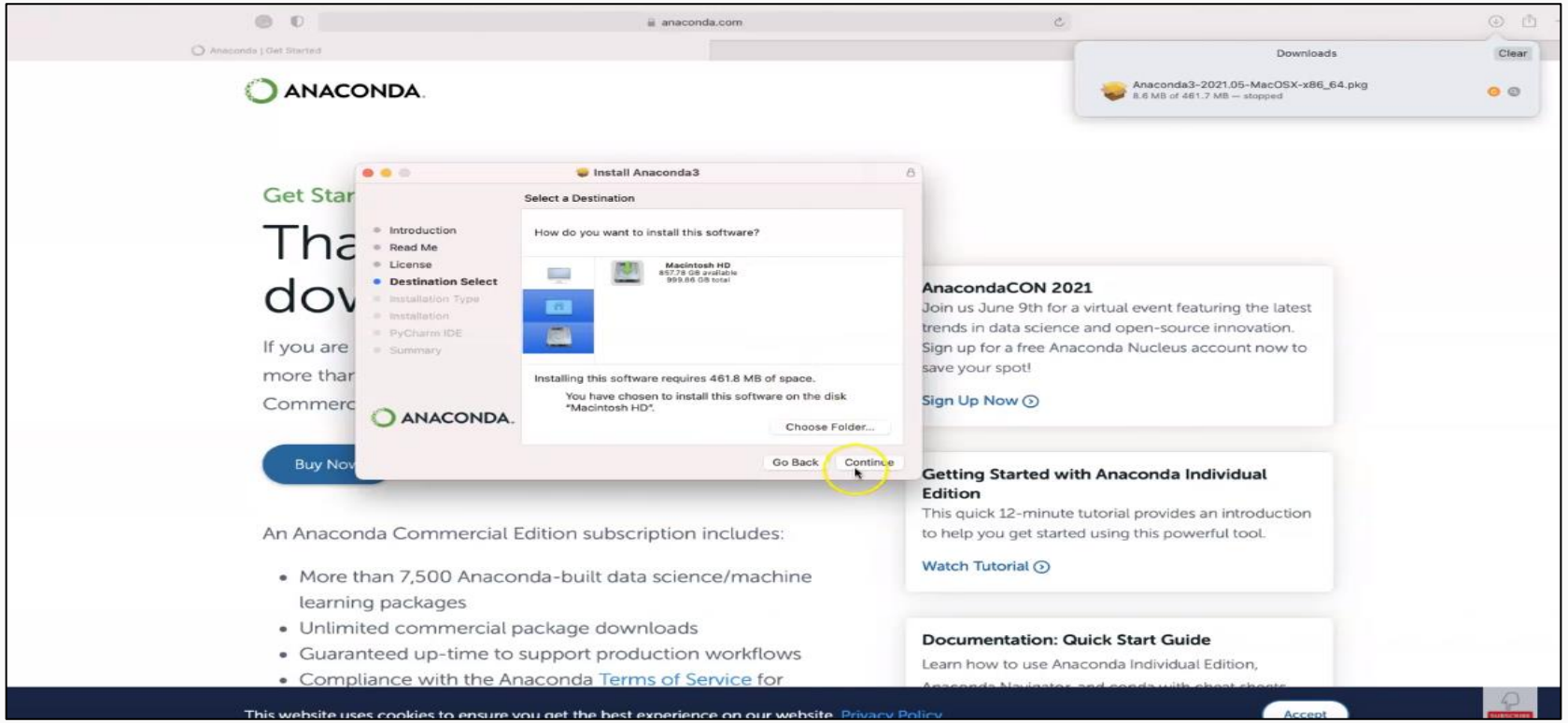
Anaconda for Mac (6/14)

- Read the License Agreement and click continue.



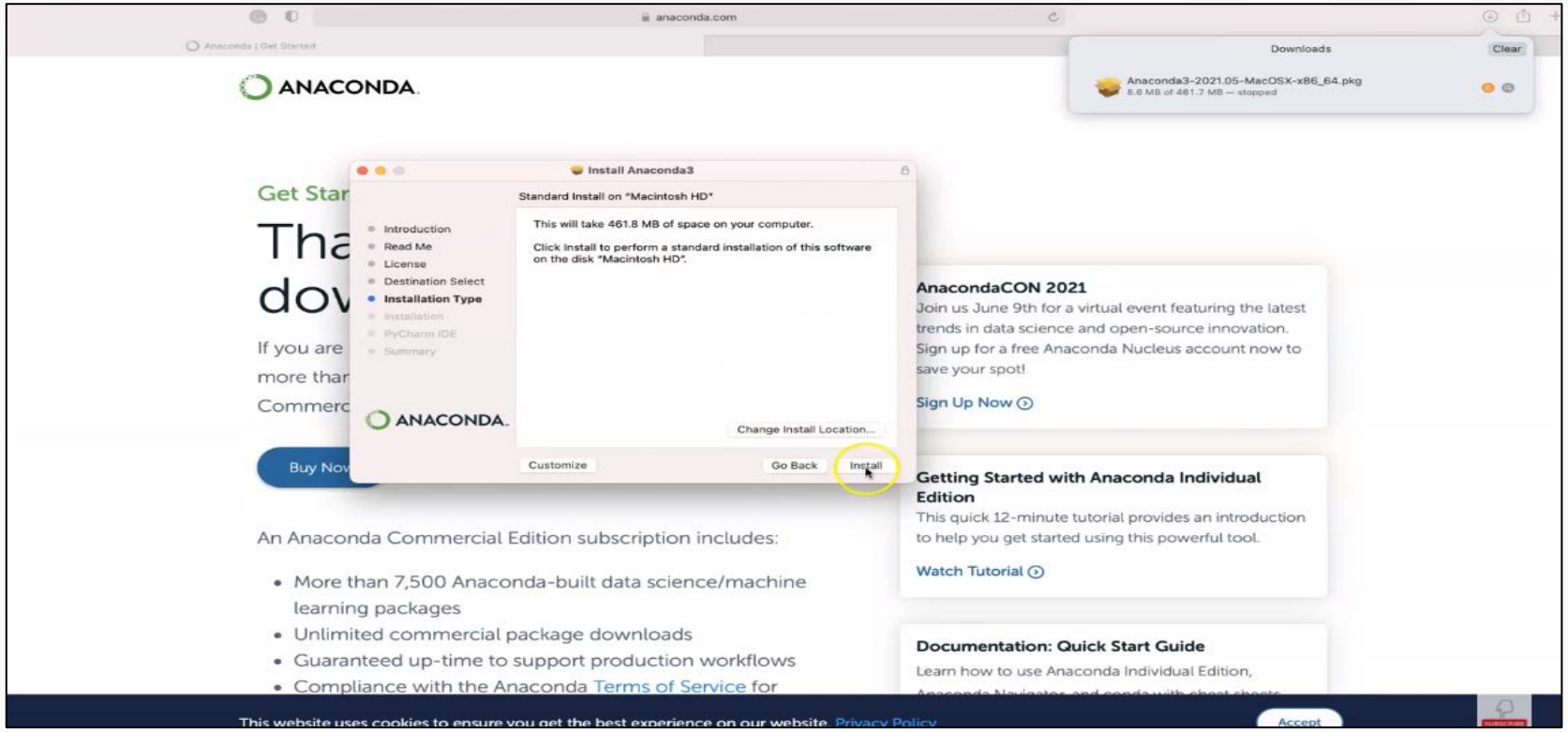
Anaconda for Mac (7/14)

- Select the destination and click Continue.



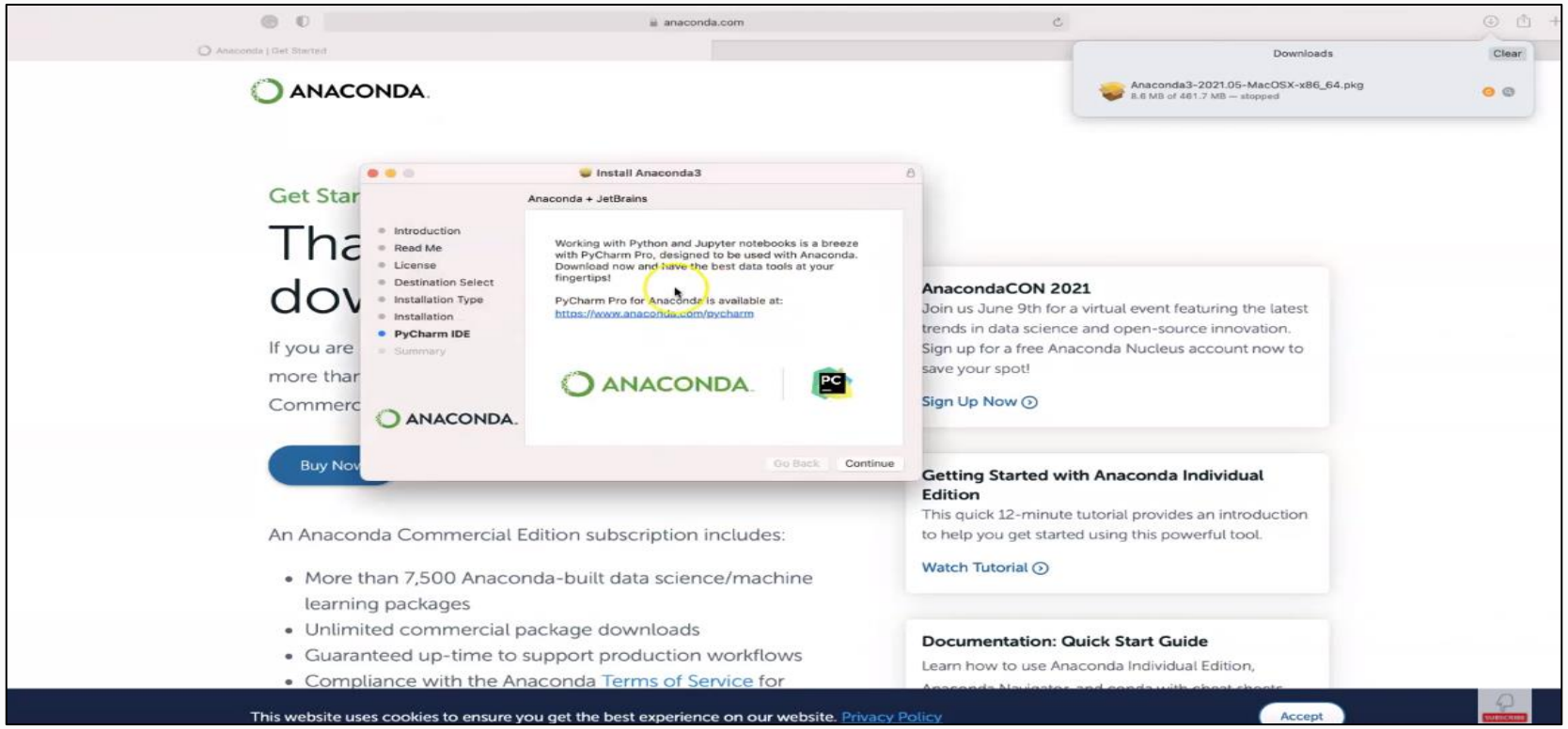
Anaconda for Mac (8/14)

- Click Install.



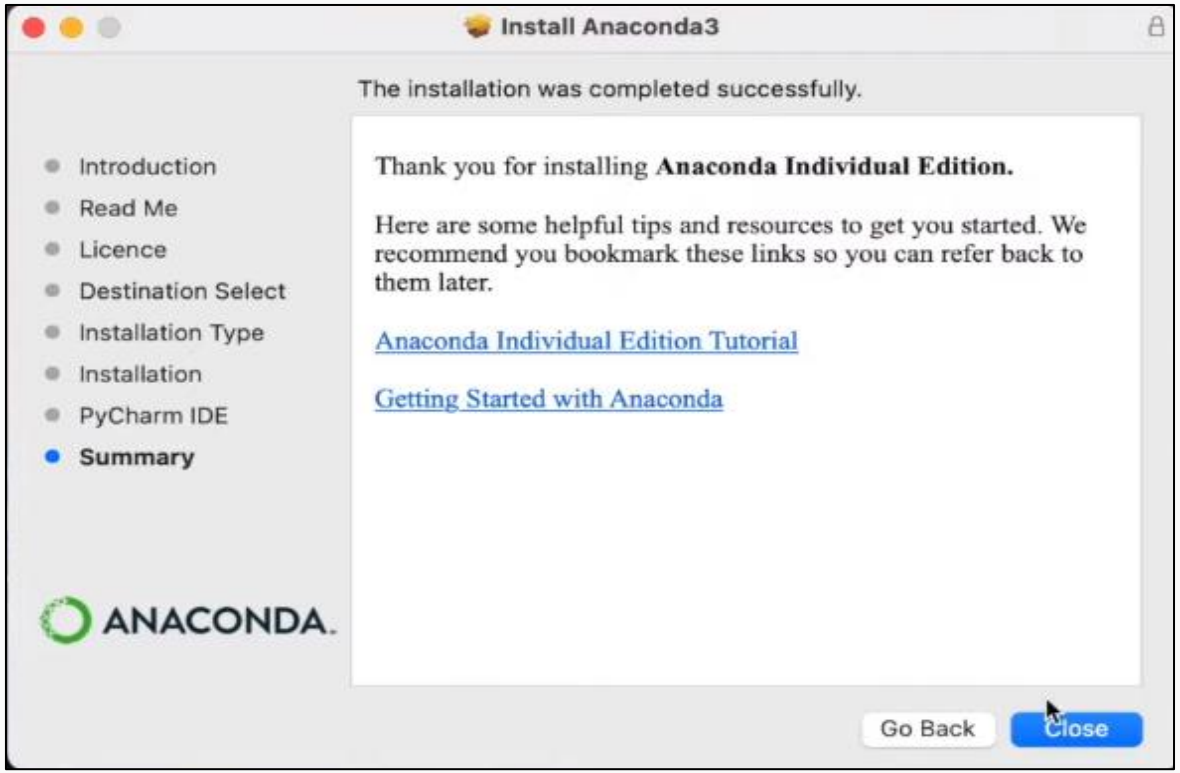
Anaconda for Mac (9/14)

- Click Continue.



Anaconda for Mac (10/14)

- Once the installation finishes, click Close.

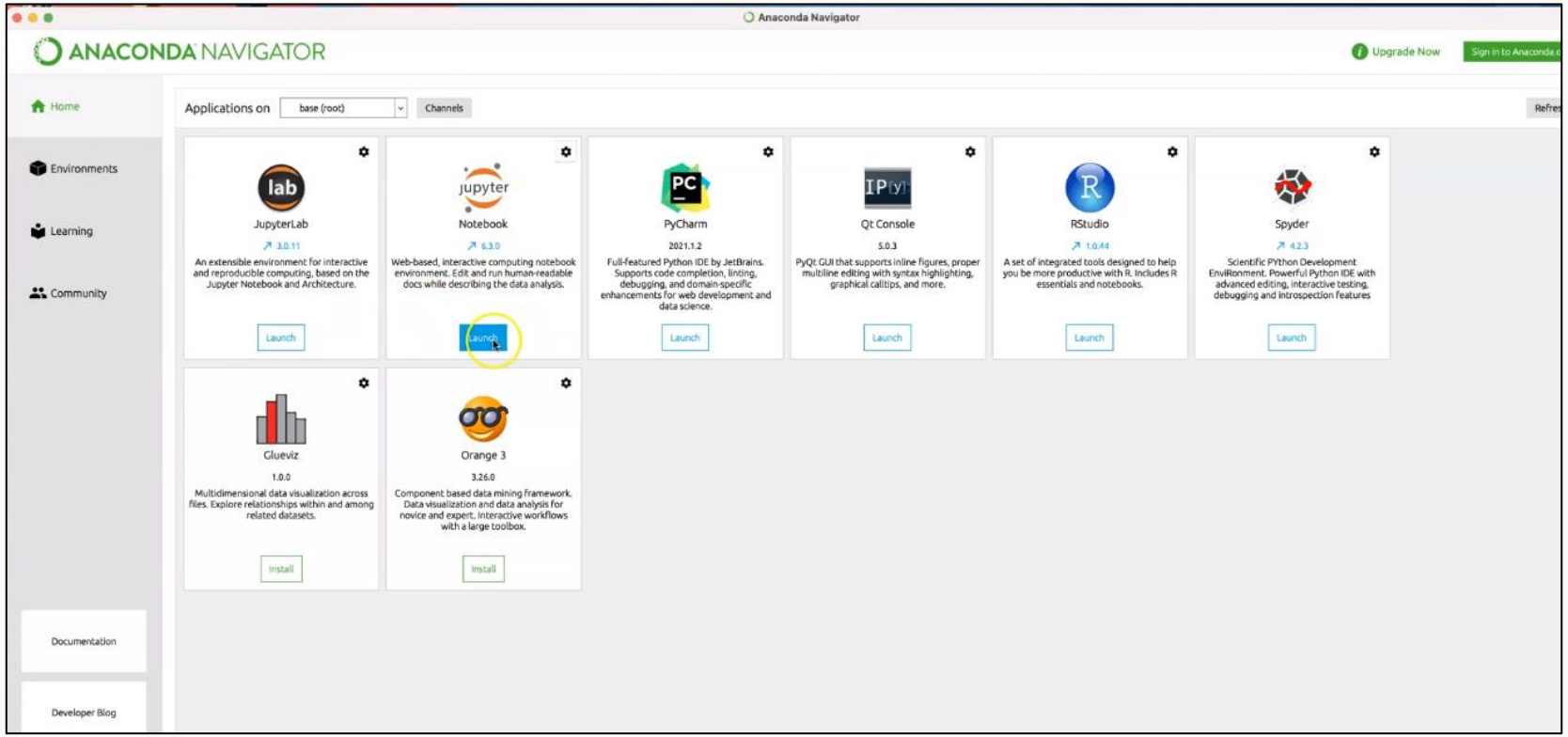


Anaconda for Mac (11/14)

- Open your spotlight and click on Anaconda Navigator to launch it.

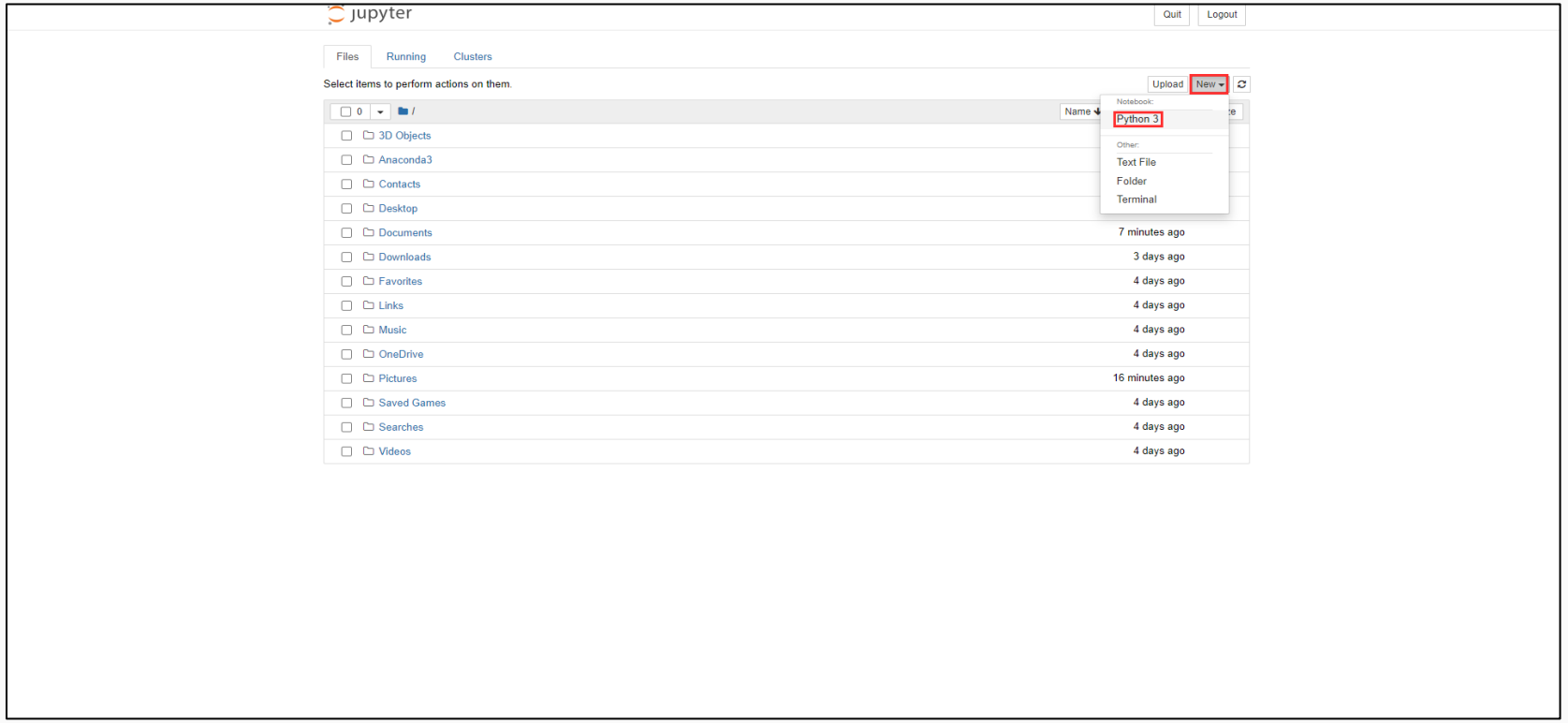
Anaconda for Mac (12/14)

- Launch Jupyter Notebook. It will take sometime to launch.



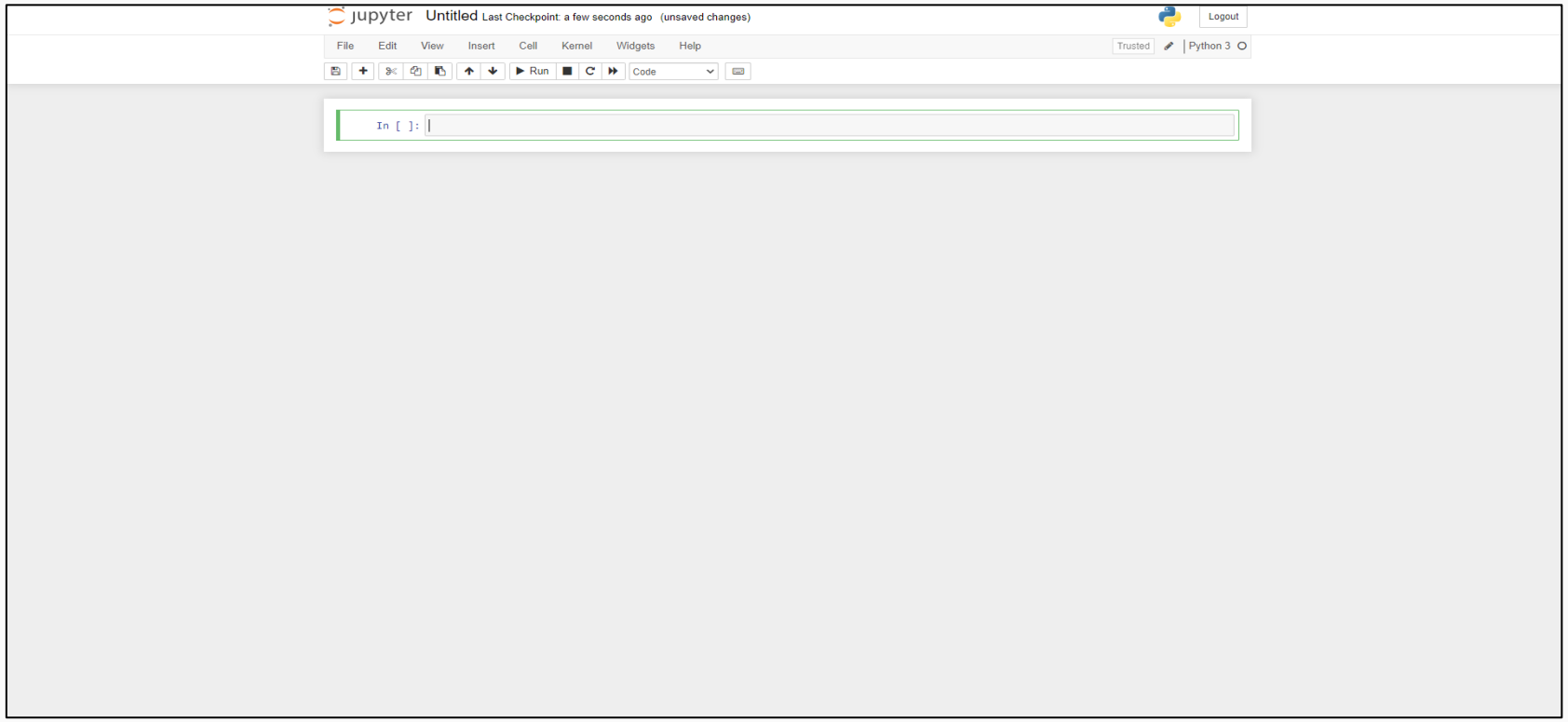
Anaconda for Mac (13/14)

- This will launch Jupyter Notebook in your default browser.
- Click on New -> Python3 to create a new Notebook.



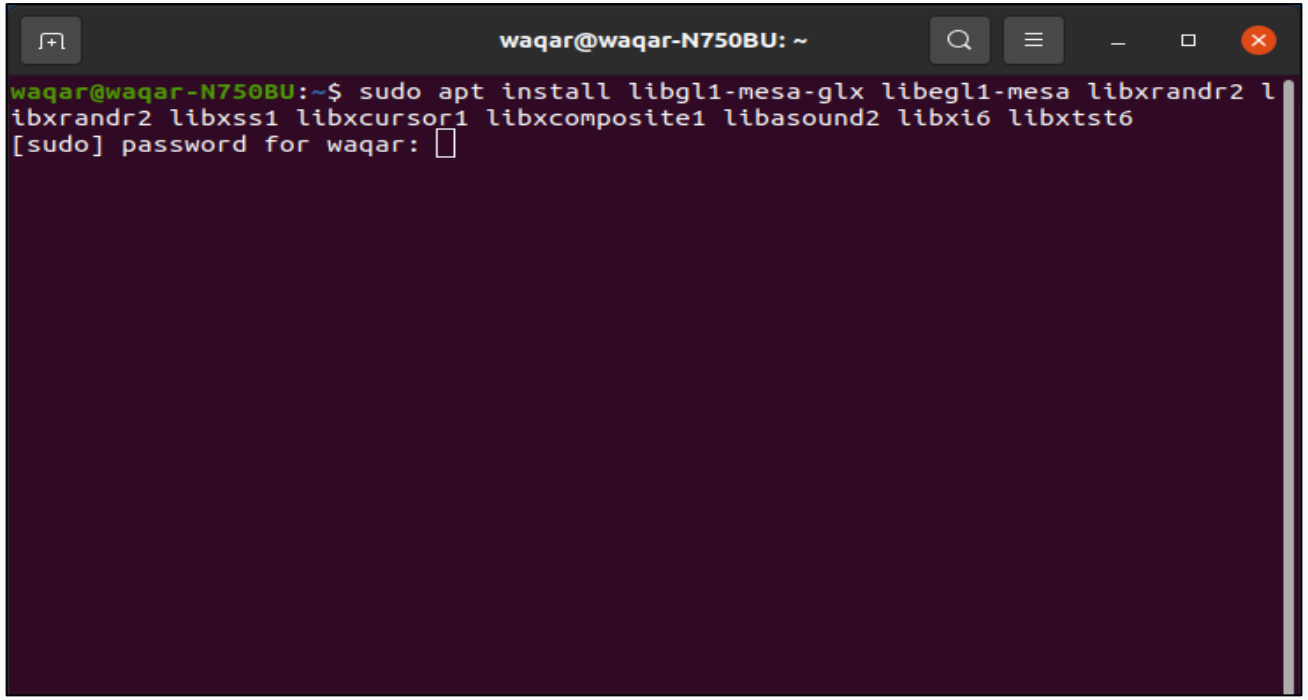
Anaconda for Mac (14/14)

- This is how a Notebook looks like.



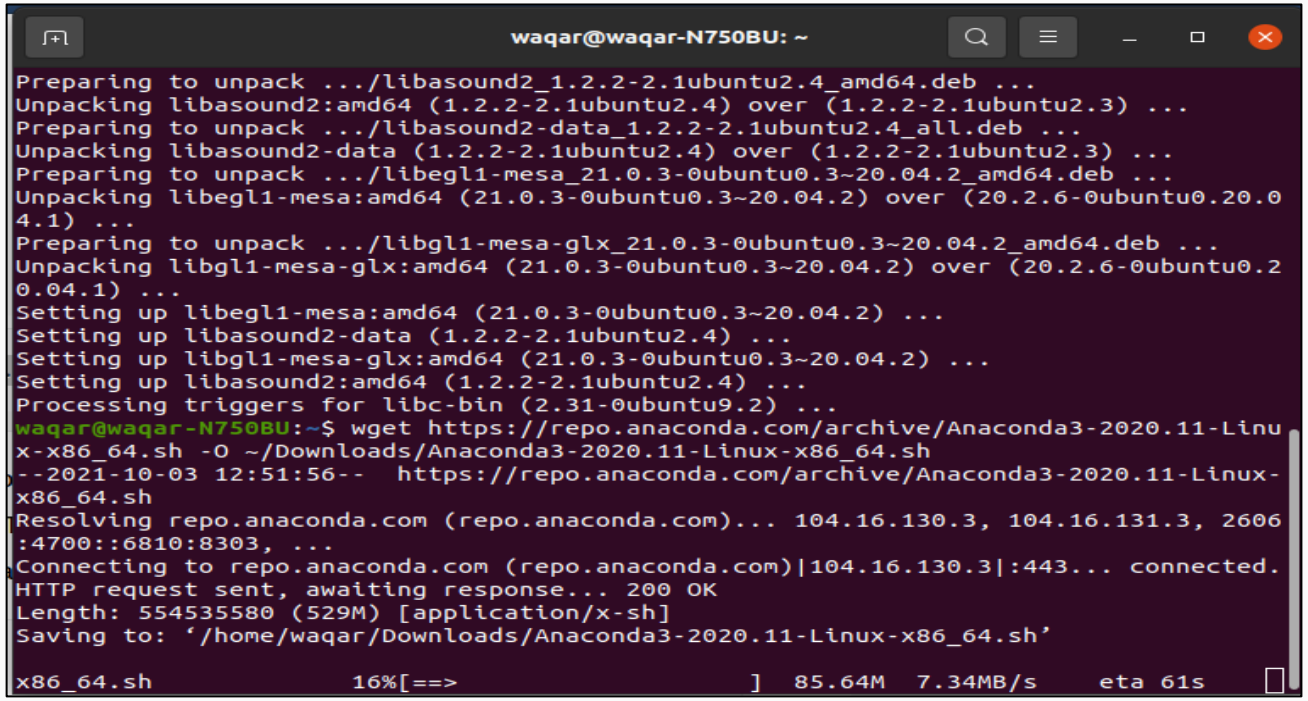
Anaconda for Ubuntu (1/11)

- Open your terminal, paste the following command, and hit enter
  sudo apt install libgl1-mesa-glx libegl1-mesa libxrandr2 libxrandr2 libxss1 libxcursor1 libxcomposite1 libasound2 libxi6 libxtst6
- Type your password and hit Enter.



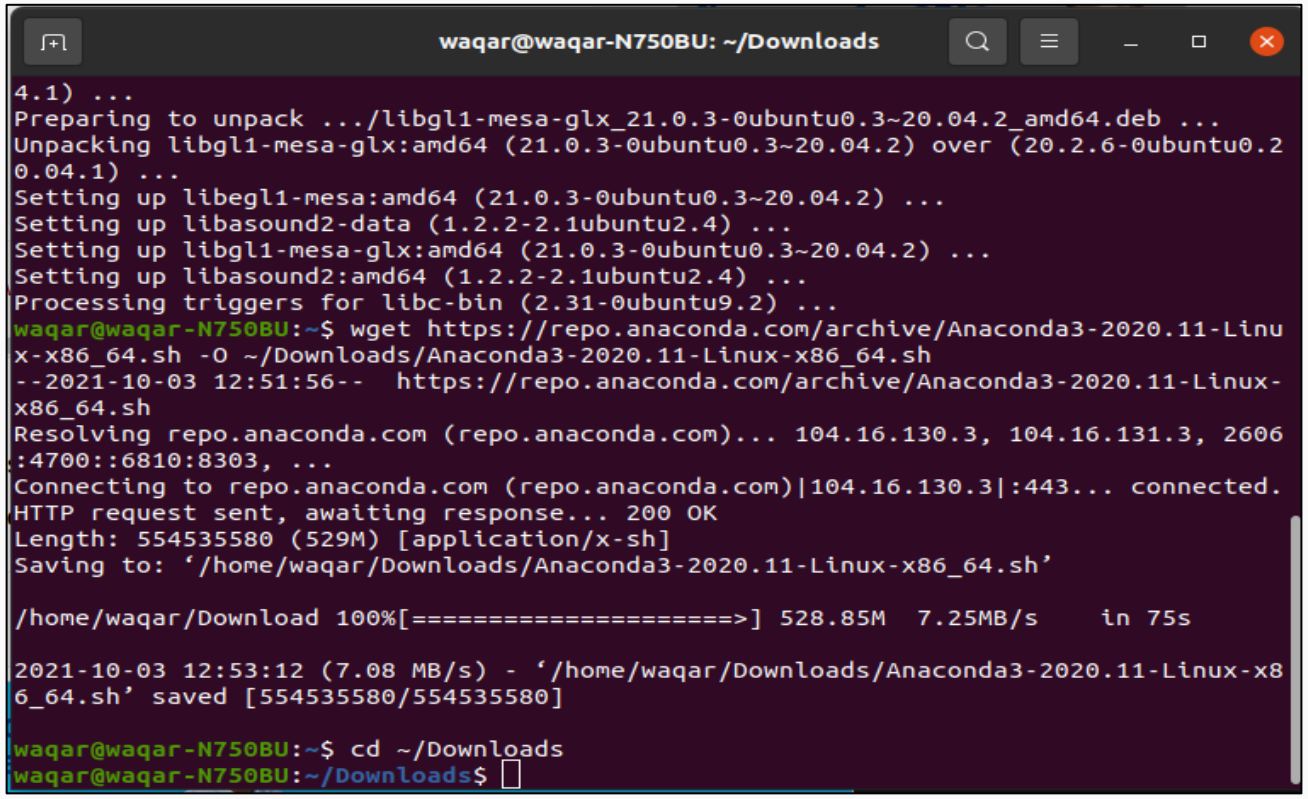
Anaconda for Ubuntu (2/11)

- Next give the following command and hit enter
  wget https://repo.anaconda.com/archive/Anaconda3-2020.11-Linux-x86_64.sh -O ~/Downloads/Anaconda3-2020.11-Linux-x86_64.sh



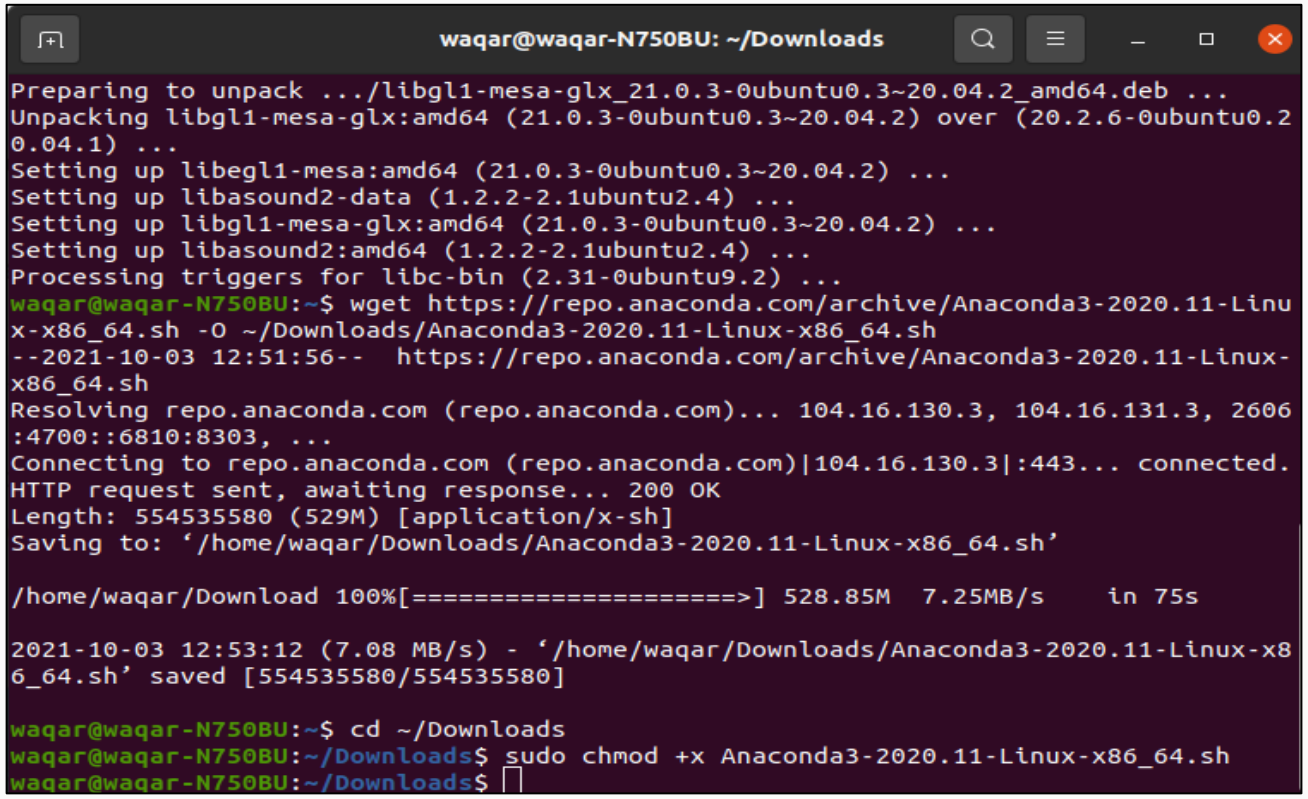
Anaconda for Ubuntu (3/11)

- Next give the following command and hit enter
  cd ~/Downloads



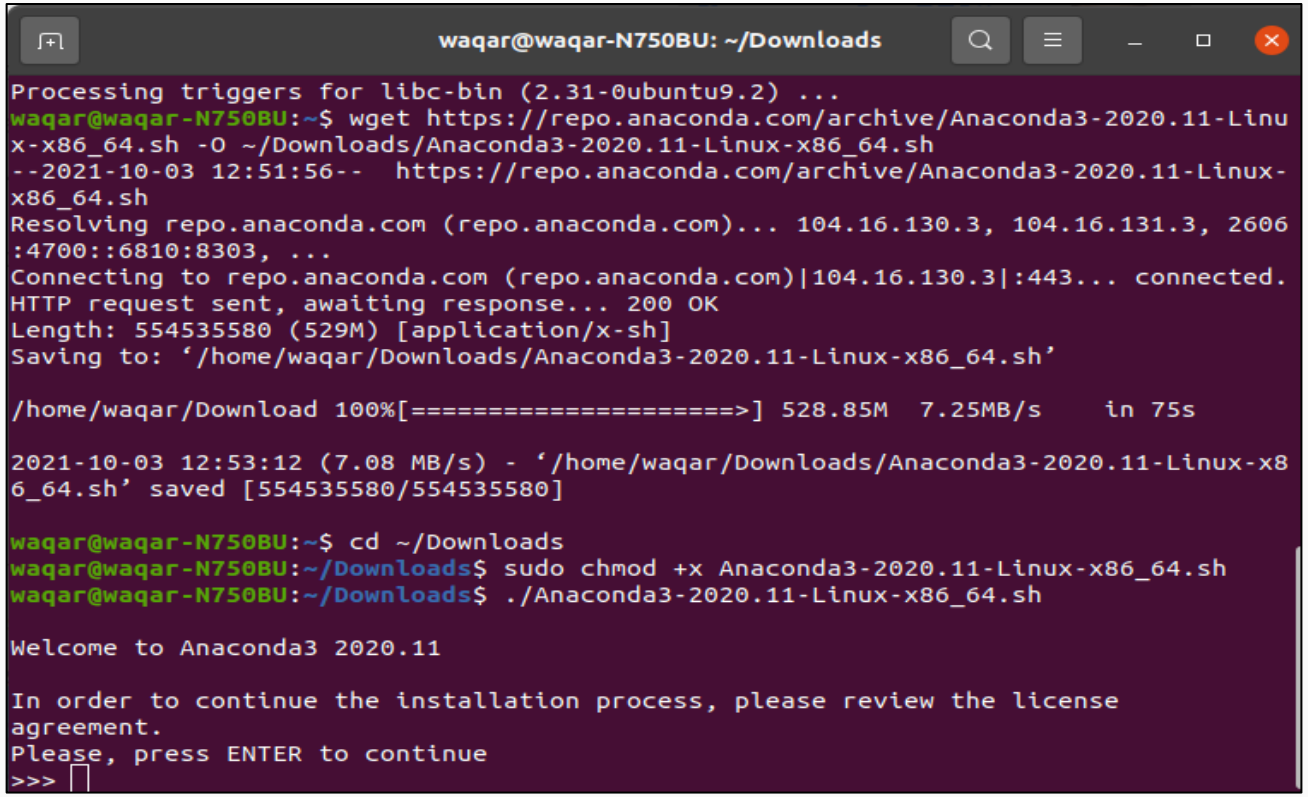
Anaconda for Ubuntu (4/11)

- Next give the following command and hit enter
  sudo chmod +x Anaconda3-2020.11-Linux-x86_64.sh



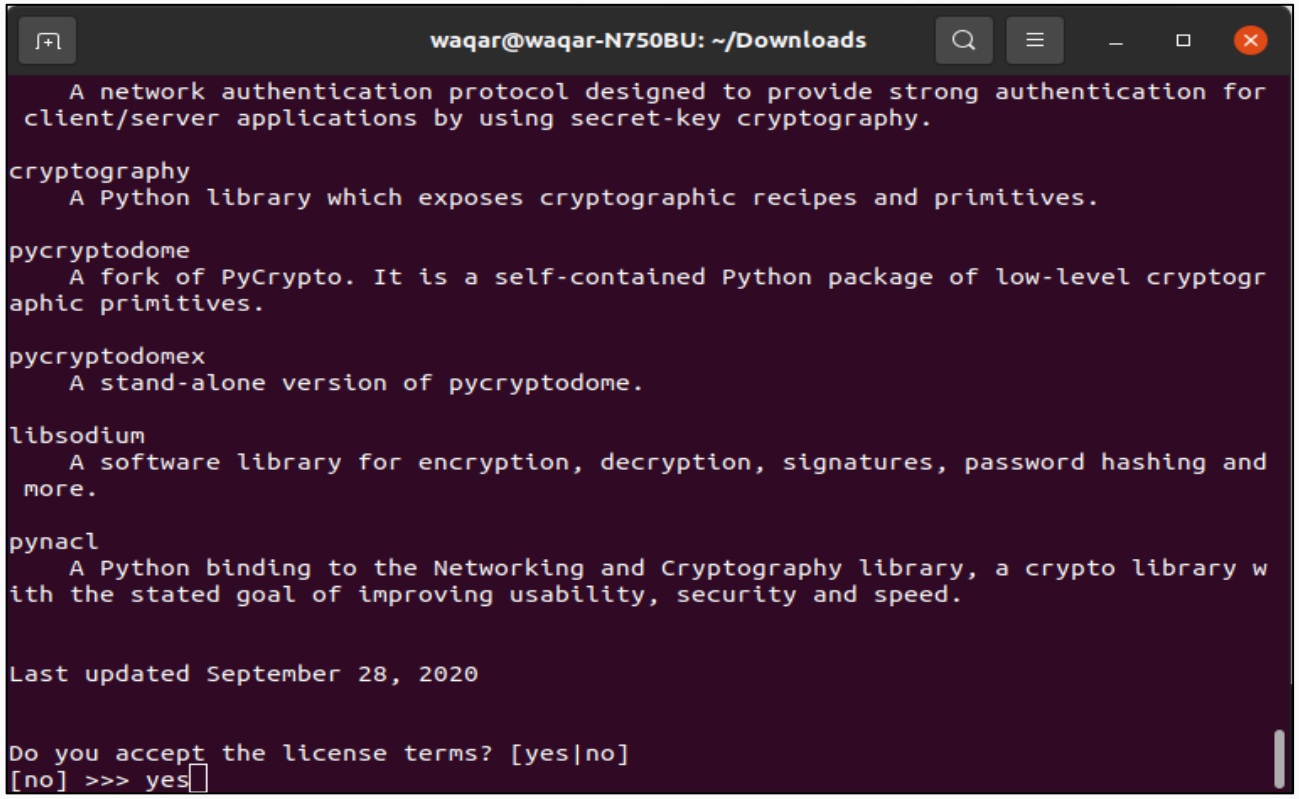
Anaconda for Ubuntu (5/11)

- Next give the following command and hit enter
  ./Anaconda3-2020.11-Linux-x86_64.sh
- When prompted, press enter to continue.



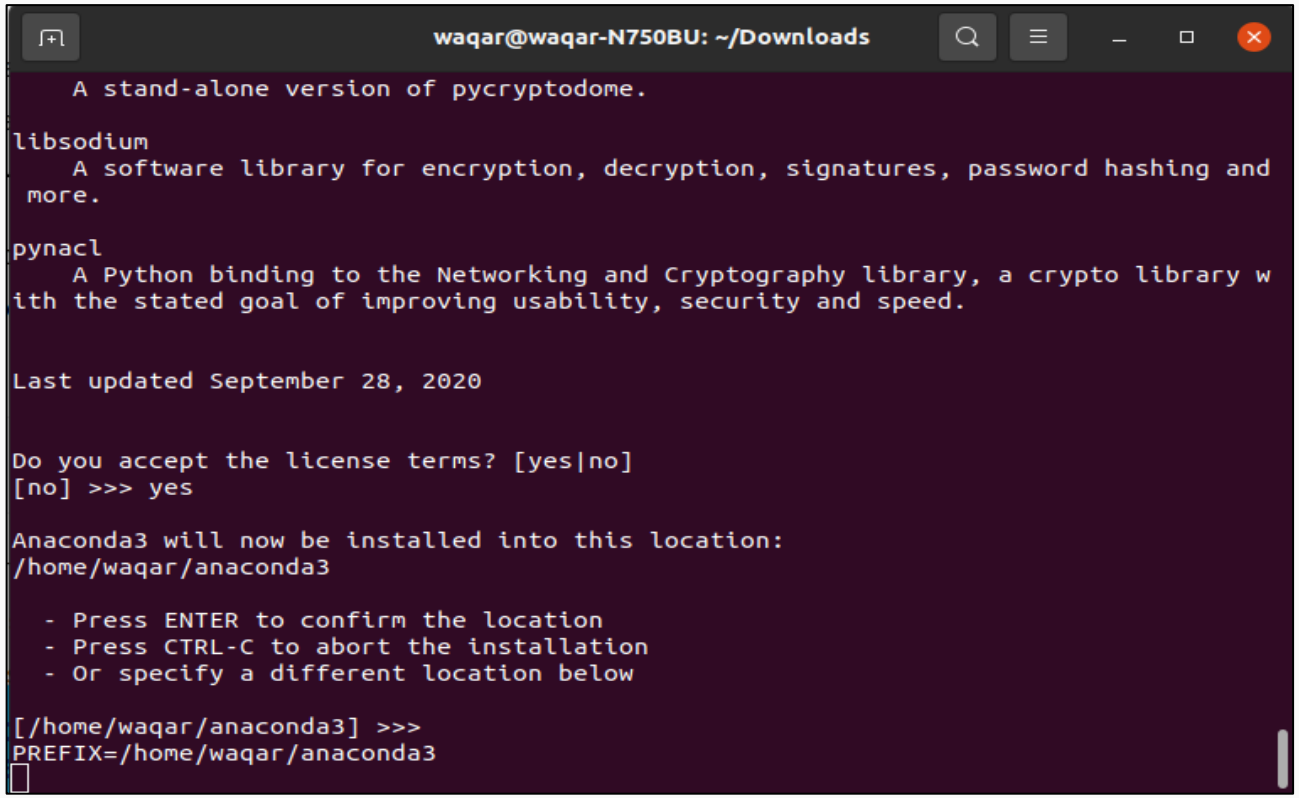
Anaconda for Ubuntu (6/11)

- Press the Enter key to scroll through the License Agreement.
- After reading the Agreement, type yes in the prompt and hit enter.



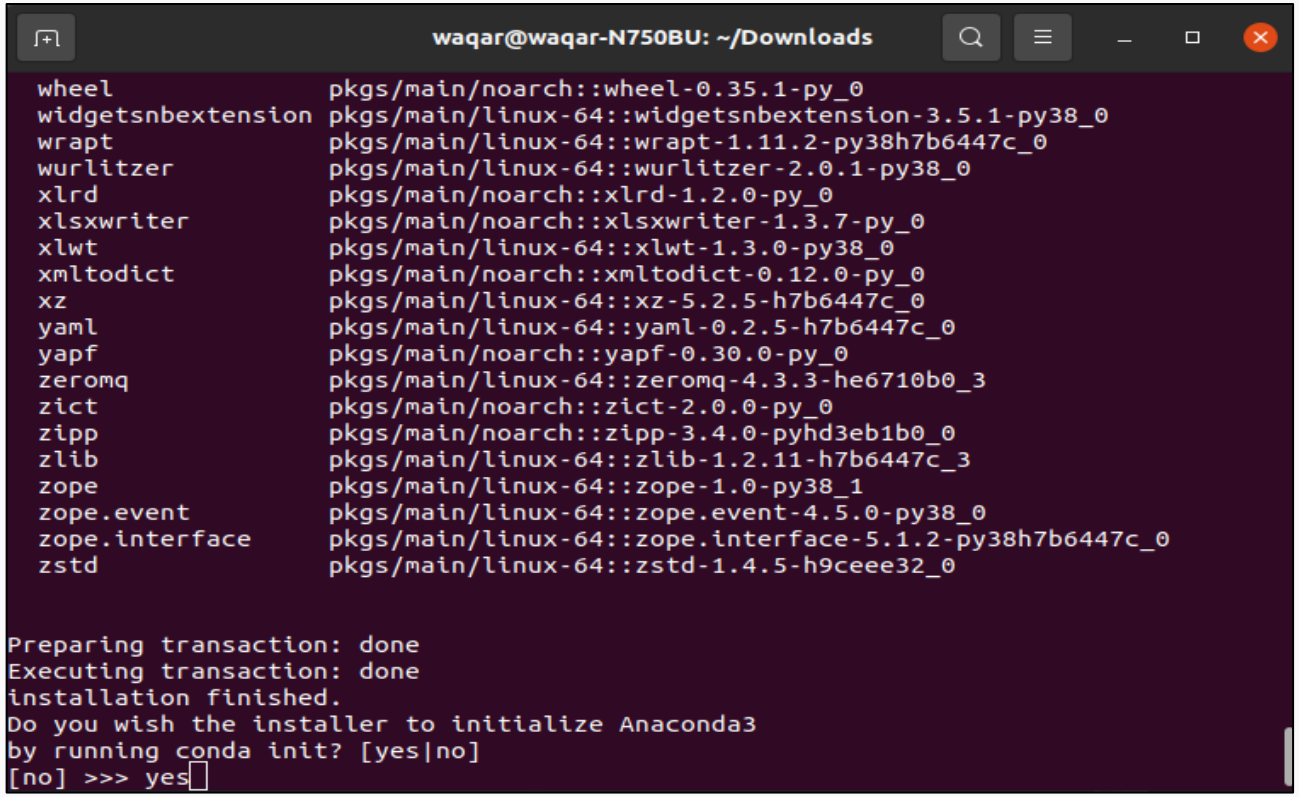
Anaconda for Ubuntu (7/11)

- Anaconda installer will ask you where do you want to install Anaconda. We suggest pressing Enter key to install it in the home directory.
- After sometime, Anaconda will be installed on your machine.



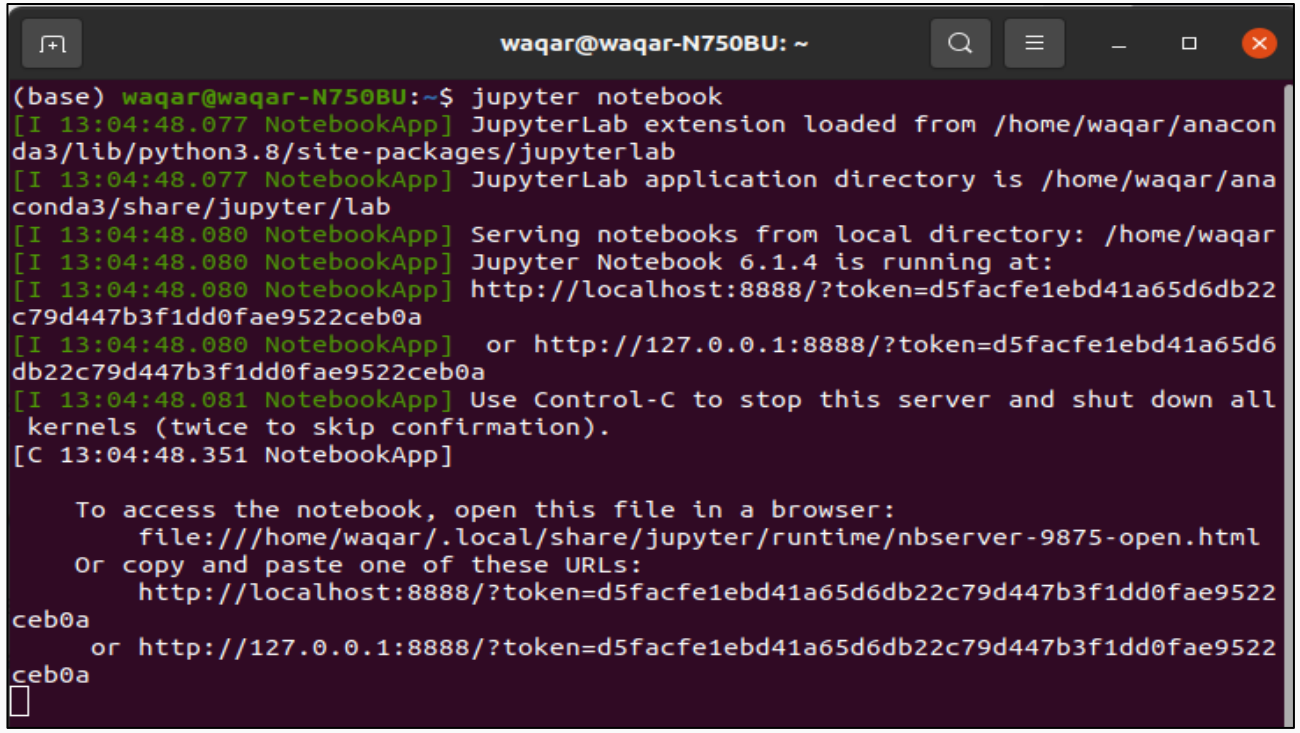
Anaconda for Ubuntu (8/11)

- Once installation is finished, you will be asked “Do you wish the installer to initialize Anaconda3 by running conda init?” type “yes” and hit Enter. Setup will finish the installation process.



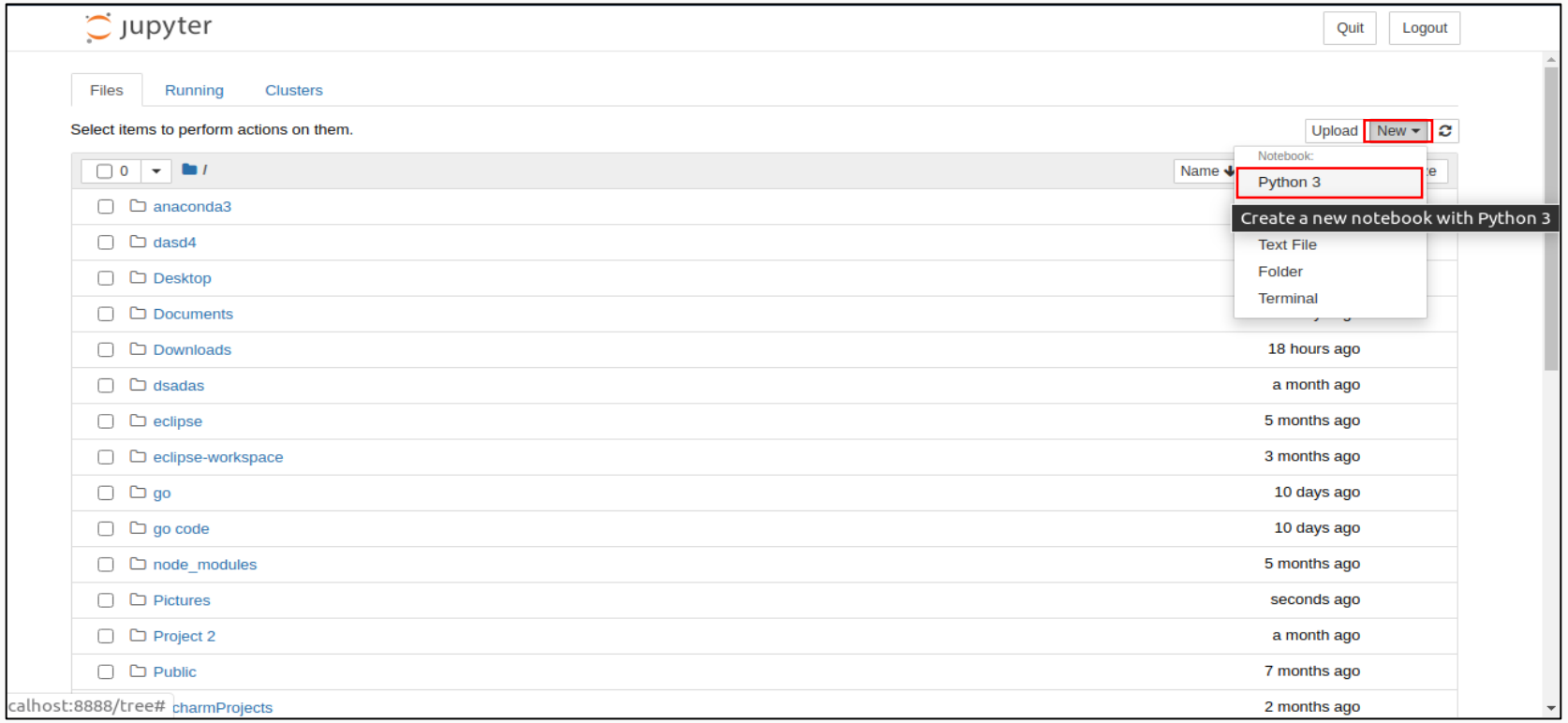
Anaconda for Ubuntu (9/11)

- To launch Jupyter Notebook, restart your terminal.
- You will see (base) written before your home directory name.
- Type “jupyter notebook” without the quotation marks and hit enter.
- After a few minutes, Jupyter Notebook will run on your default browser.



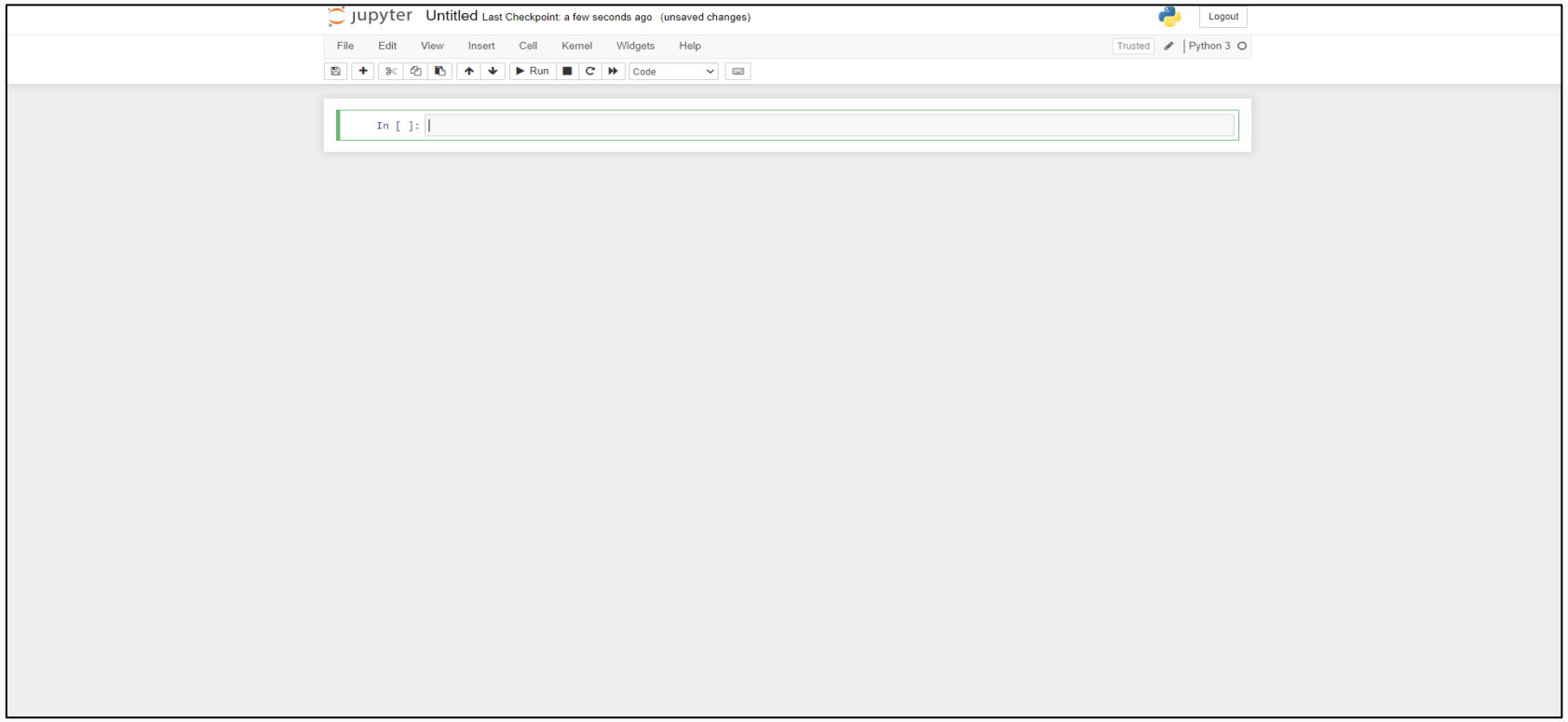
Anaconda for Mac (10/11)

- Click on New -> Python3 to create a new Notebook.



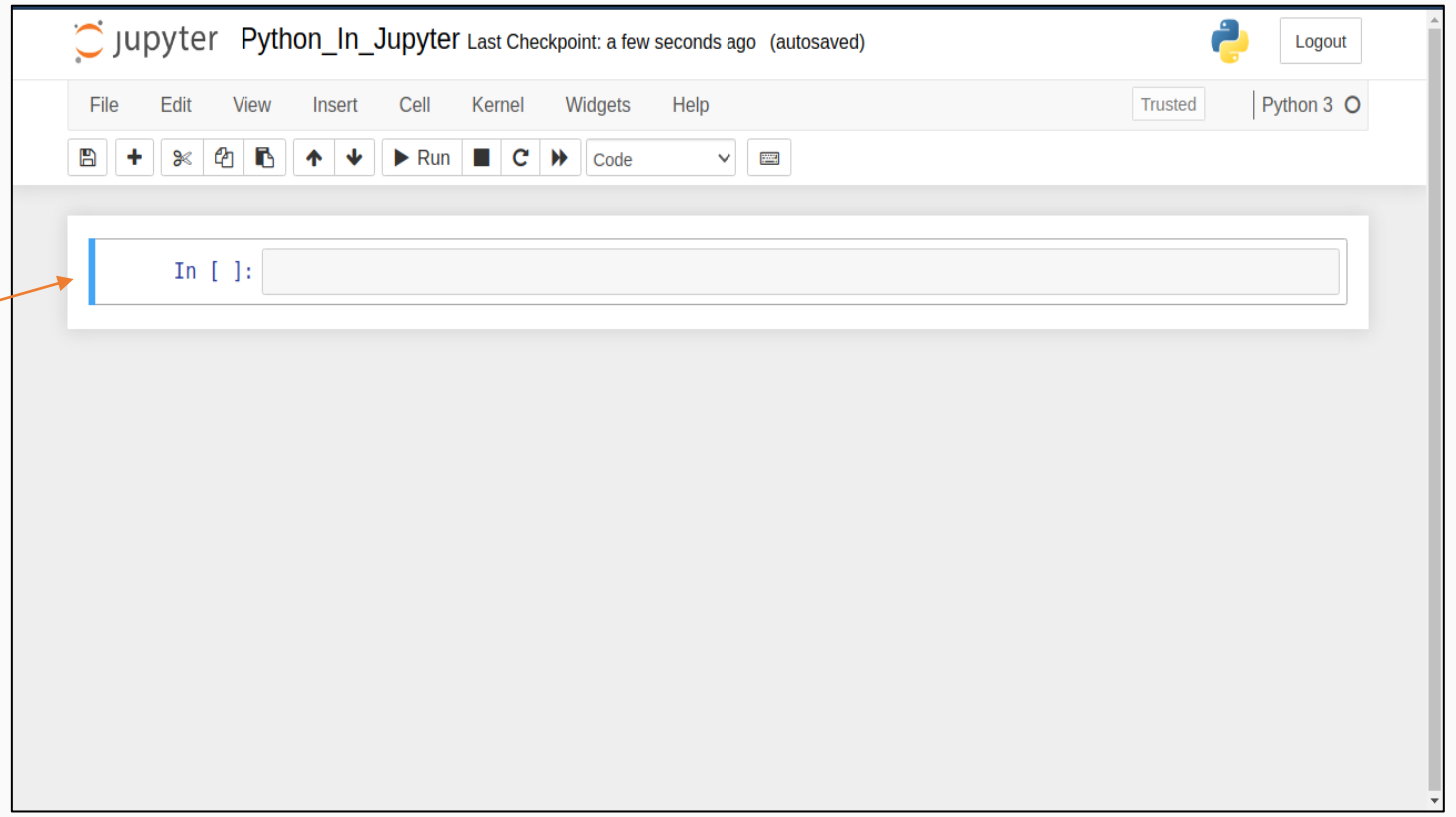
Anaconda for Ubuntu (11/11)

- This is how a Notebook looks like.



Implementing Python in Jupyter Notebook (1/2)

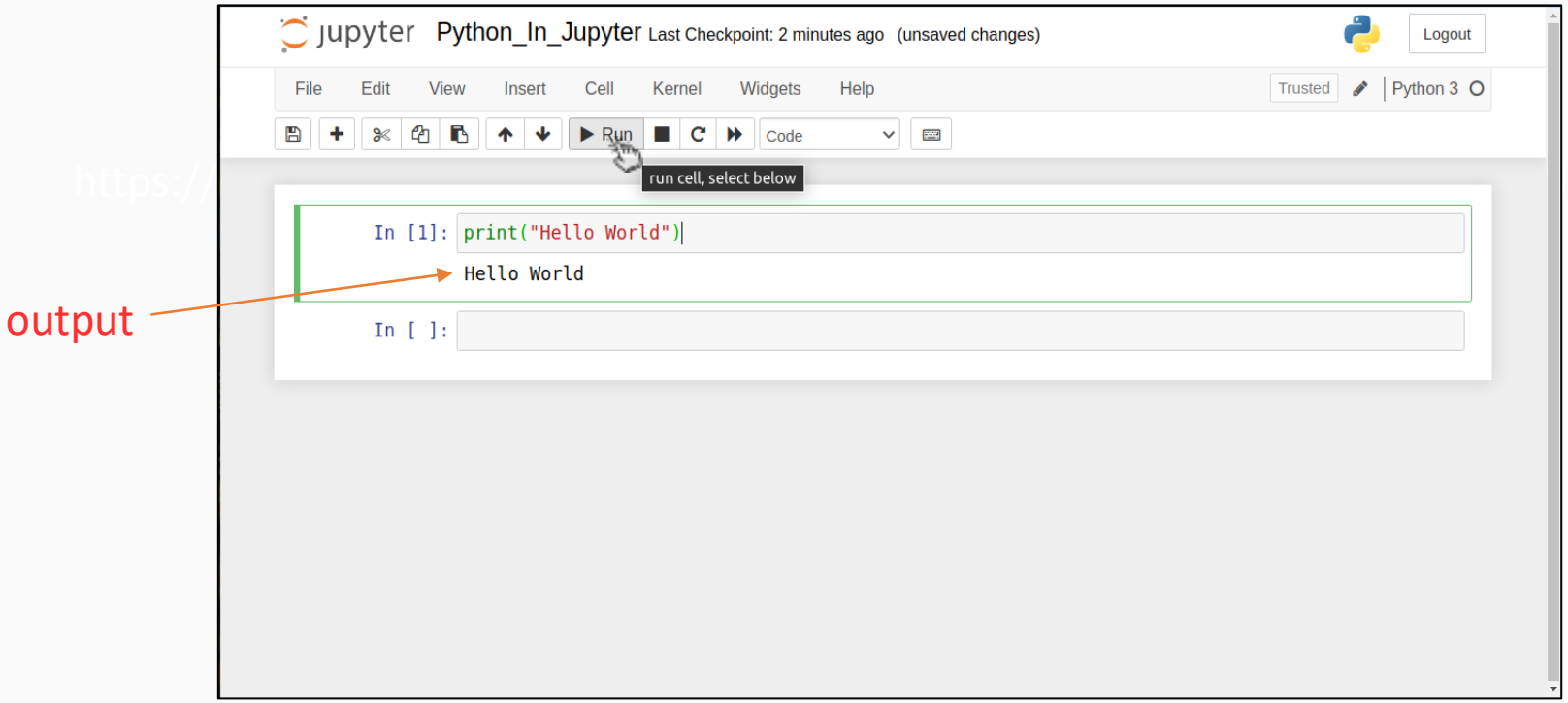
Upon launching Jupyter Notebook, this is what you will see!



This is where we write our code, called a cell. You can have as many of them as you want.

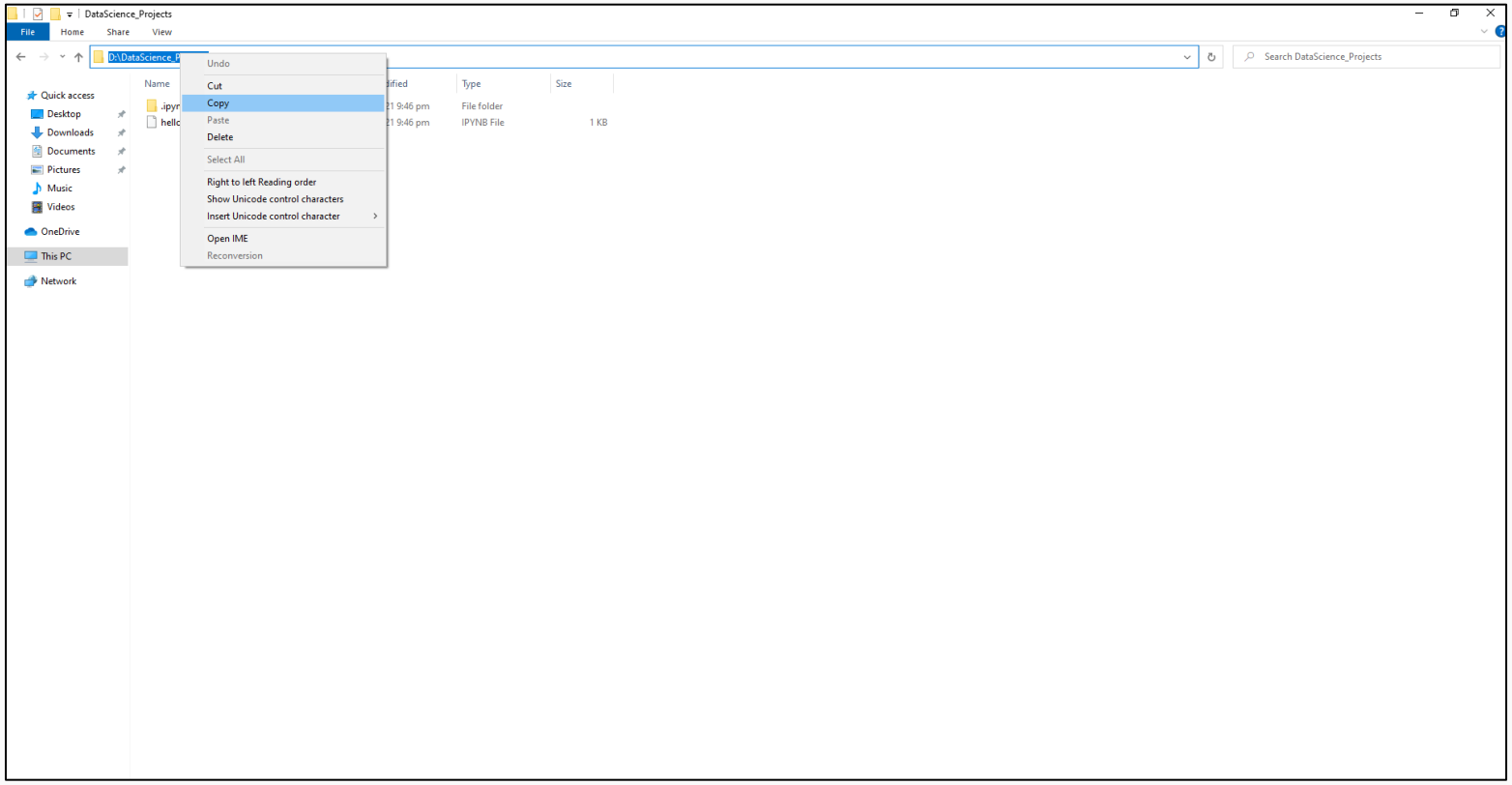
Implementing Python in Jupyter Notebook (2/2)

- Write your Python code inside a cell.
- Select the cell by clicking on it and then click on the Run button to execute the code.
- Output of a cell is produced right below it.



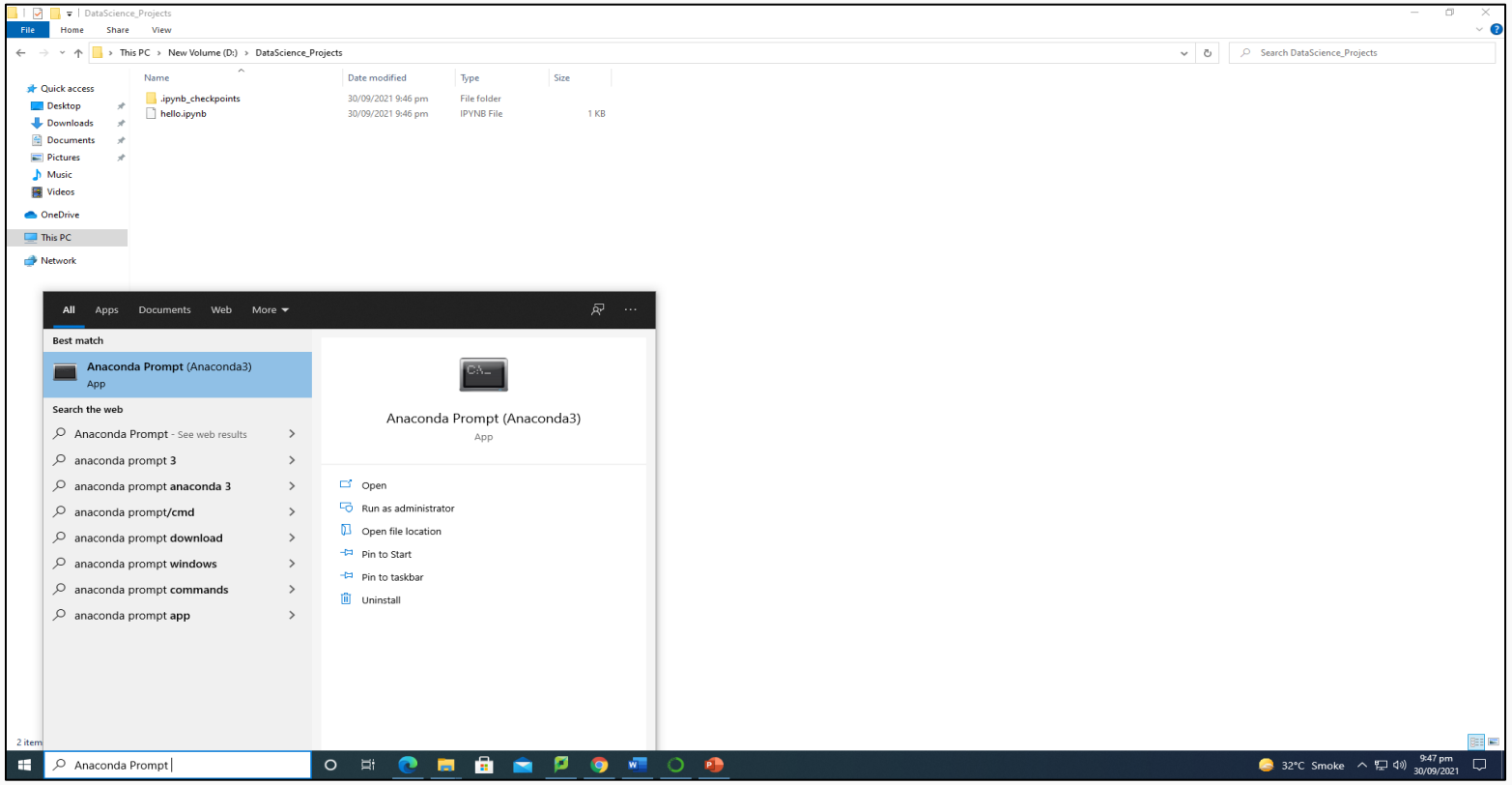
Managing Directories in Jupyter Notebook – Windows (1/7)

- To open Jupyter Notebook in a particular directory/folder in windows, copy the path of the directory.



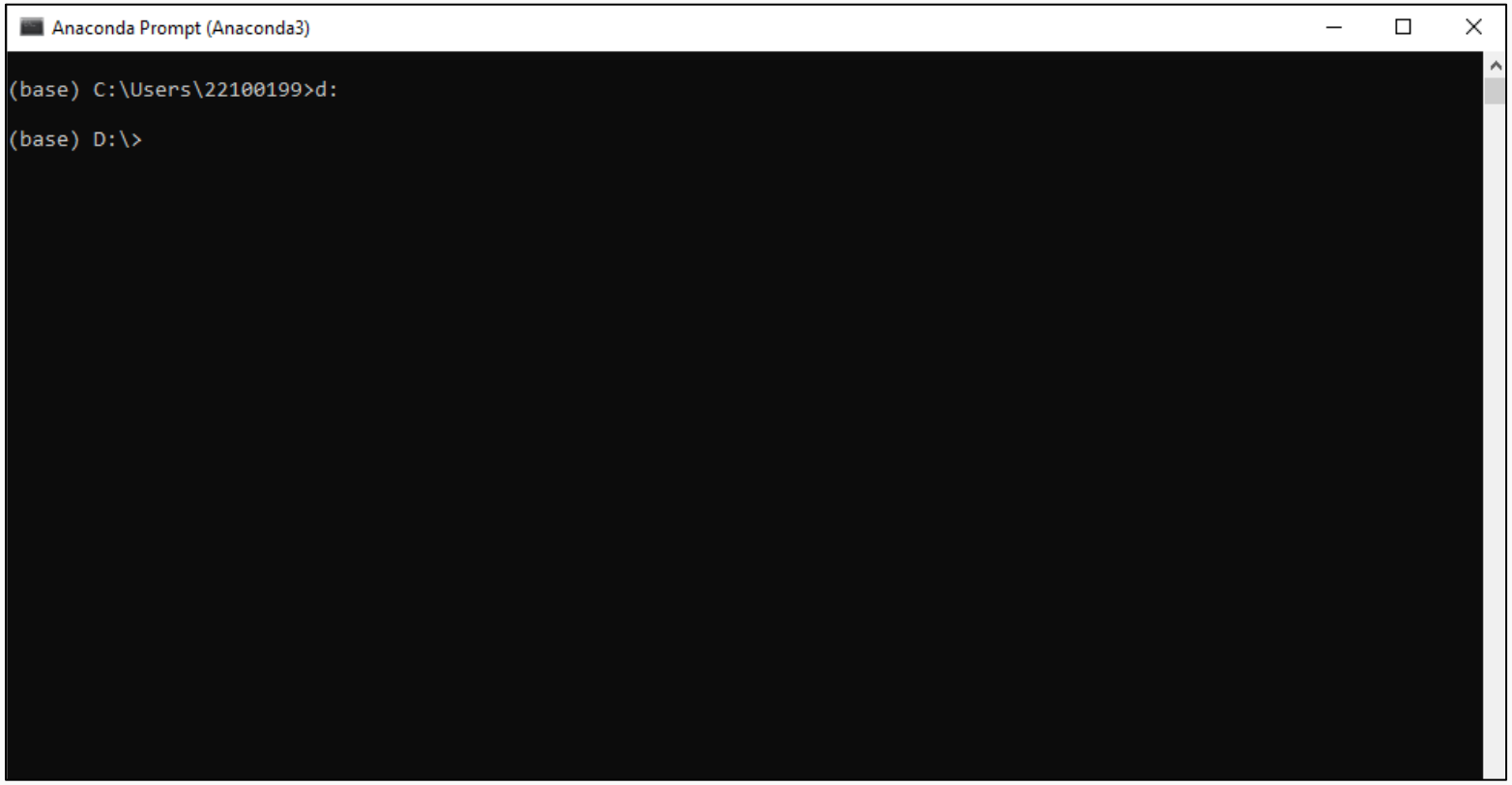
Managing Directories in Jupyter Notebook – Windows (2/7)

- In your search bar type Anaconda Prompt and launch it
  - Anaconda Prompt is the CLI of Anaconda



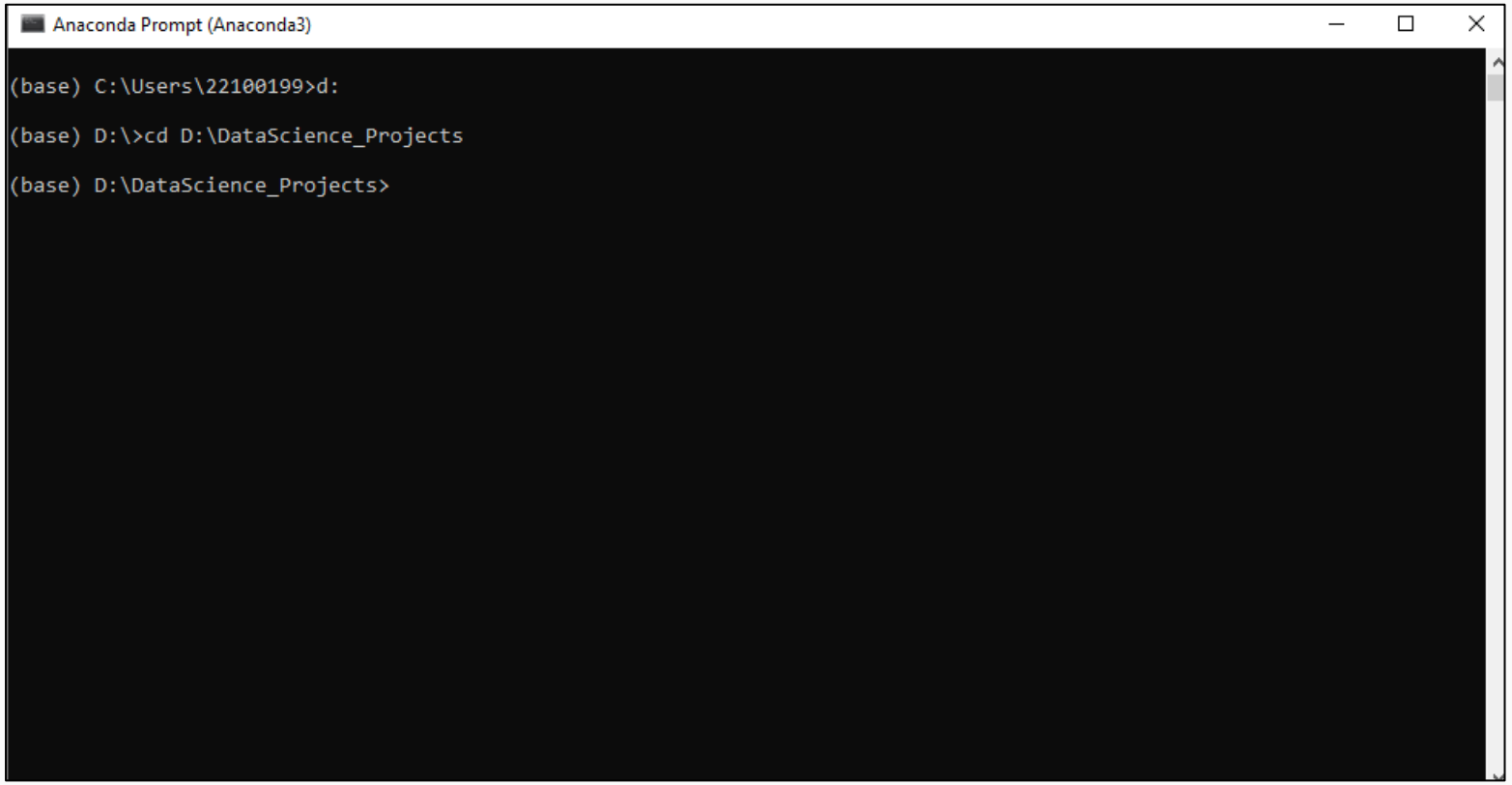
Managing Directories in Jupyter Notebook – Windows (3/7)

- Anaconda Prompt opens up in your C drive. Since we want to move to the D drive, we type d: and hit enter.



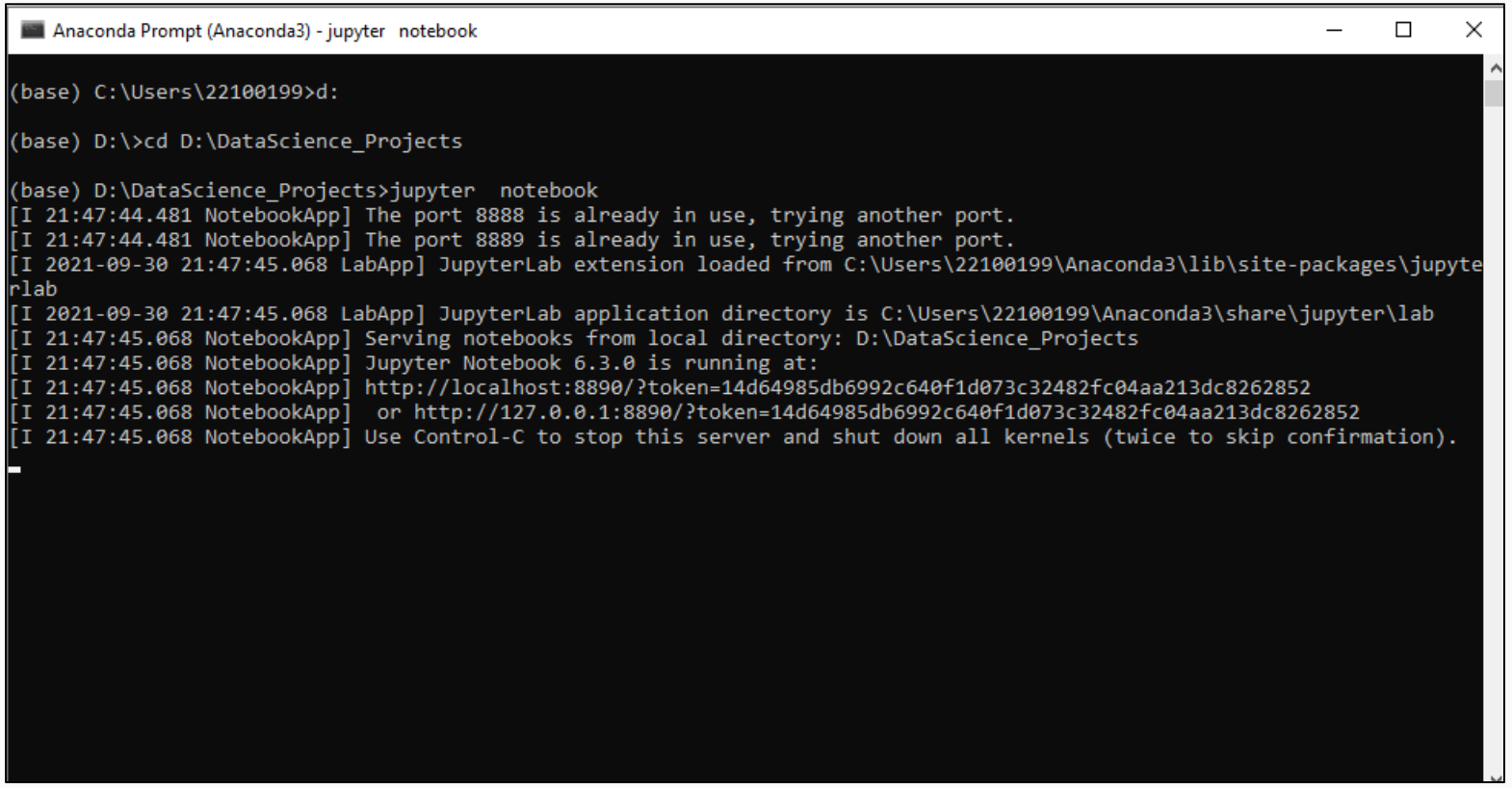
Managing Directories in Jupyter Notebook – Windows (4/7)

- Next, to switch to the desired directory we type **cd** and then **paste the path of the directory**.



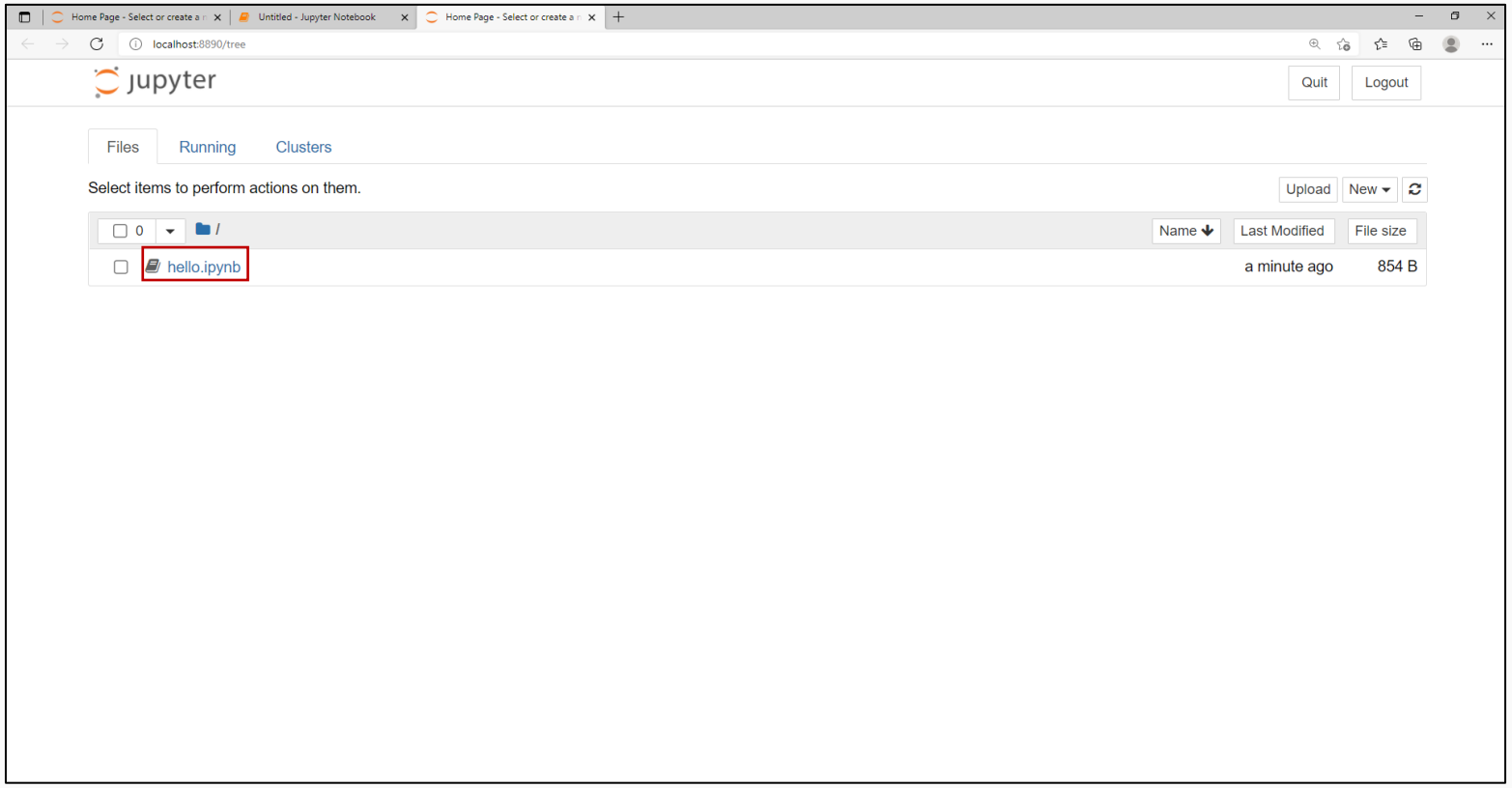
Managing Directories in Jupyter Notebook – Windows (5/7)

- Finally, we launch Jupyter Notebook by typing jupyter notebook.
- Jupyter Notebook takes a while to open.



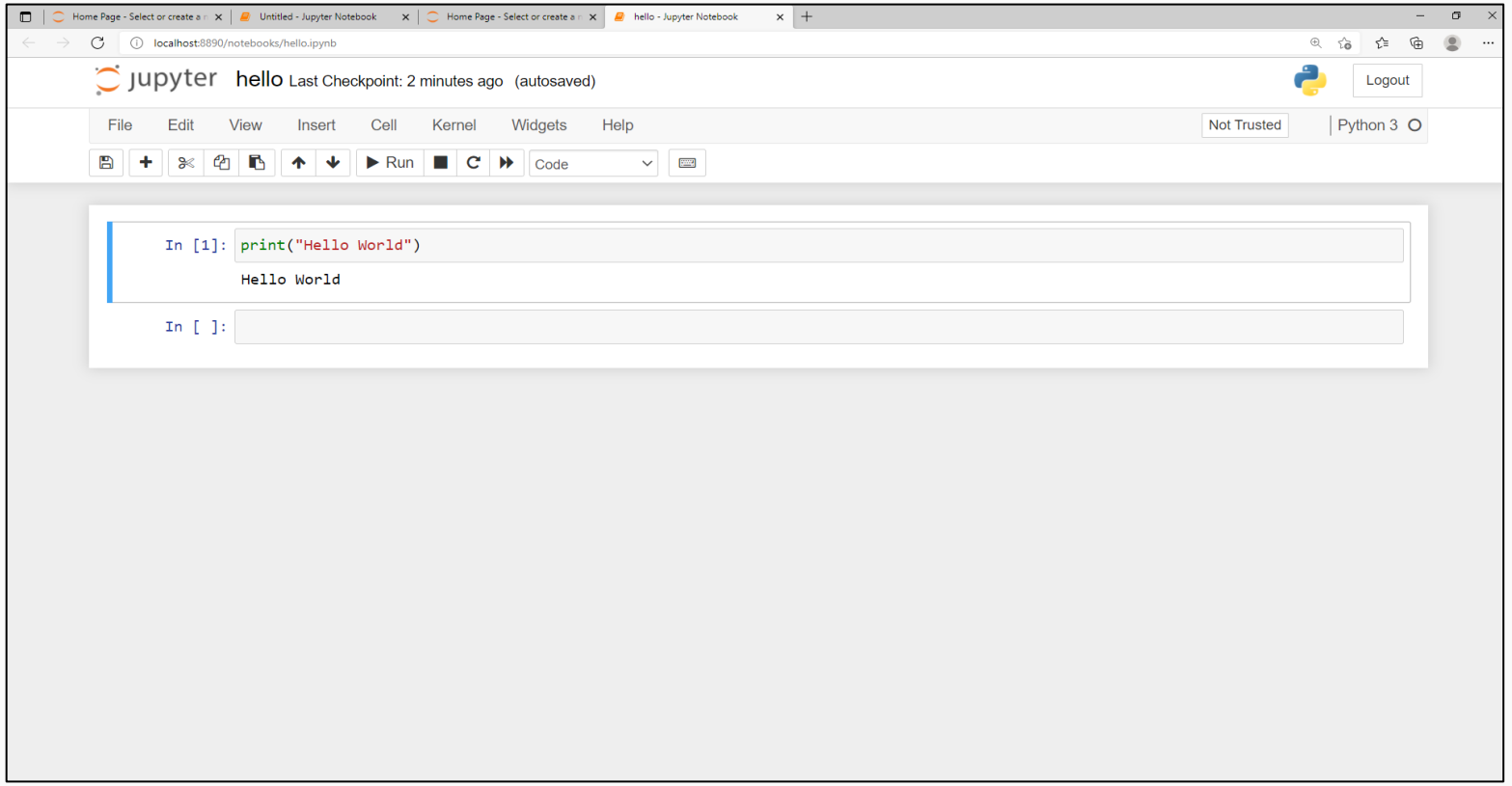
Managing Directories in Jupyter Notebook – Windows (6/7)

- Once it opens, we can either open an existing notebook or create a new one.
- In this case we open the existing notebook named hello.ipynb.



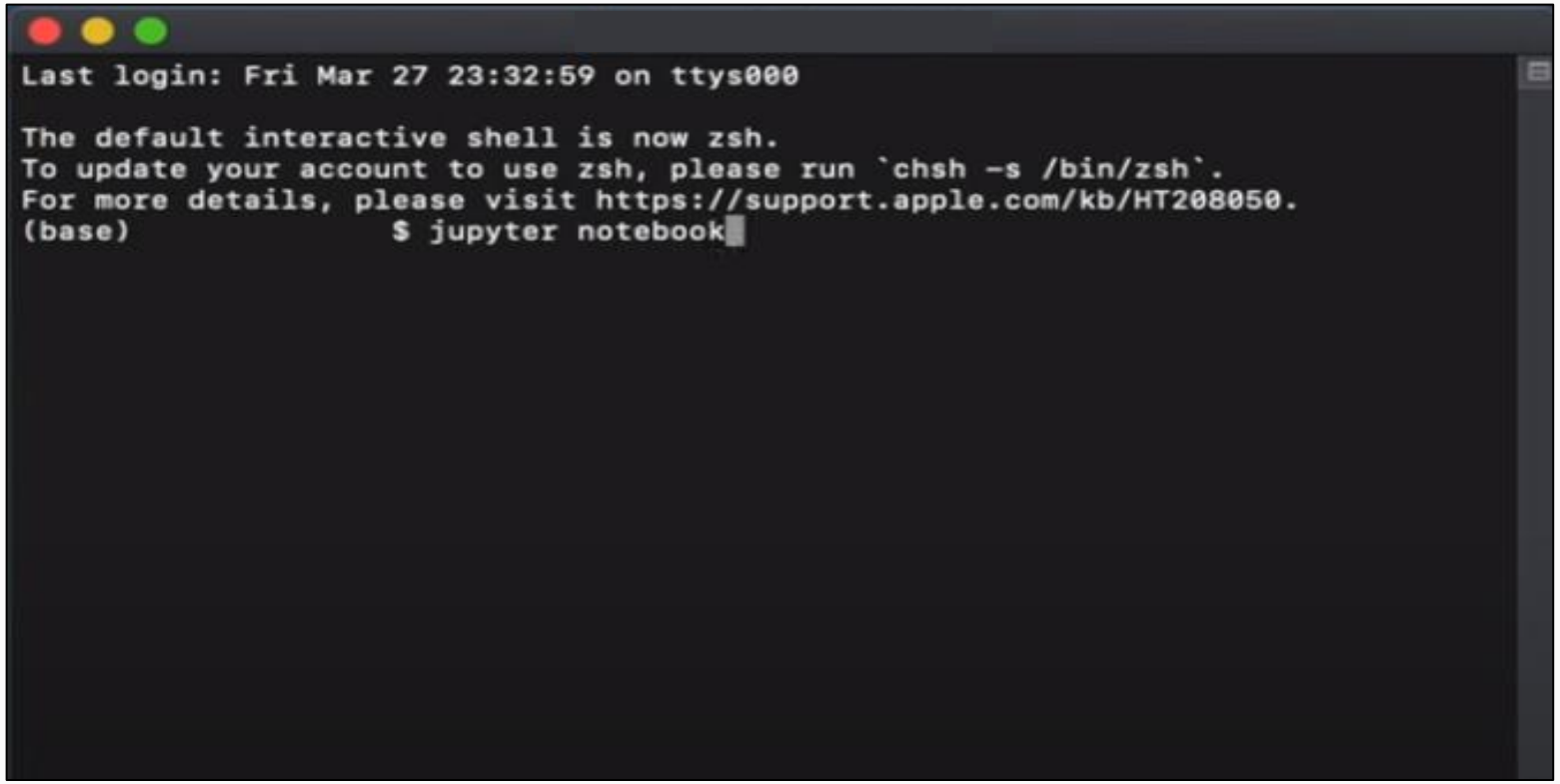
Managing Directories in Jupyter Notebook – Windows (7/7)

- This is how hello.ipynb looks like.



Managing Directories in Jupyter Notebook – Mac (1/3)

- To open Jupyter Notebook in a particular directory/folder in Mac, open that directory in terminal.
- Type ‘jupyter notebook’ without the quotation marks and hit Enter.

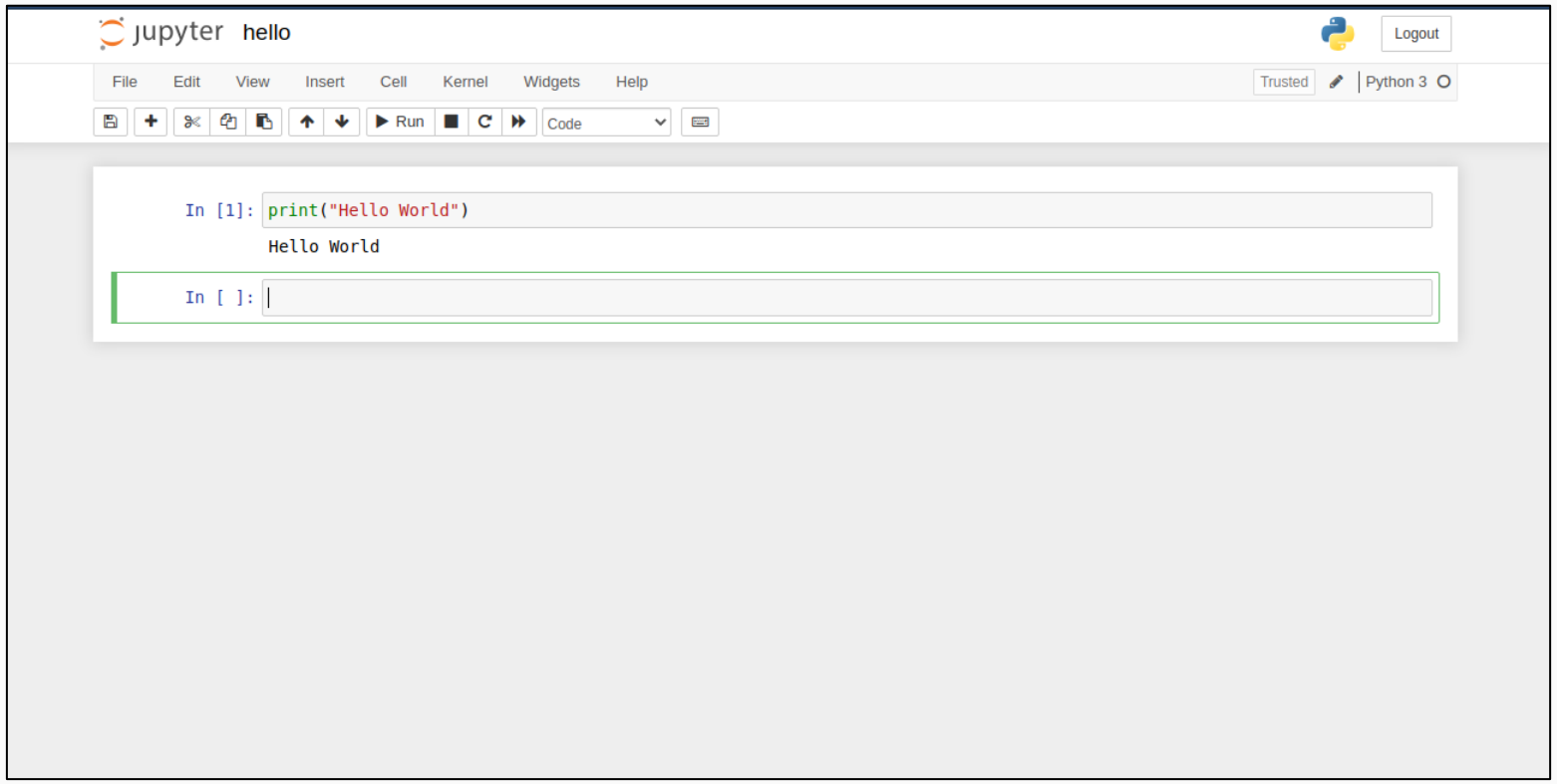


Managing Directories in Jupyter Notebook – Mac (2/3)

- Jupyter Notebook will launch in your browser.
- You can create a new Notebook or open an existing one.
- Let's open this 'hello.ipynb' file.

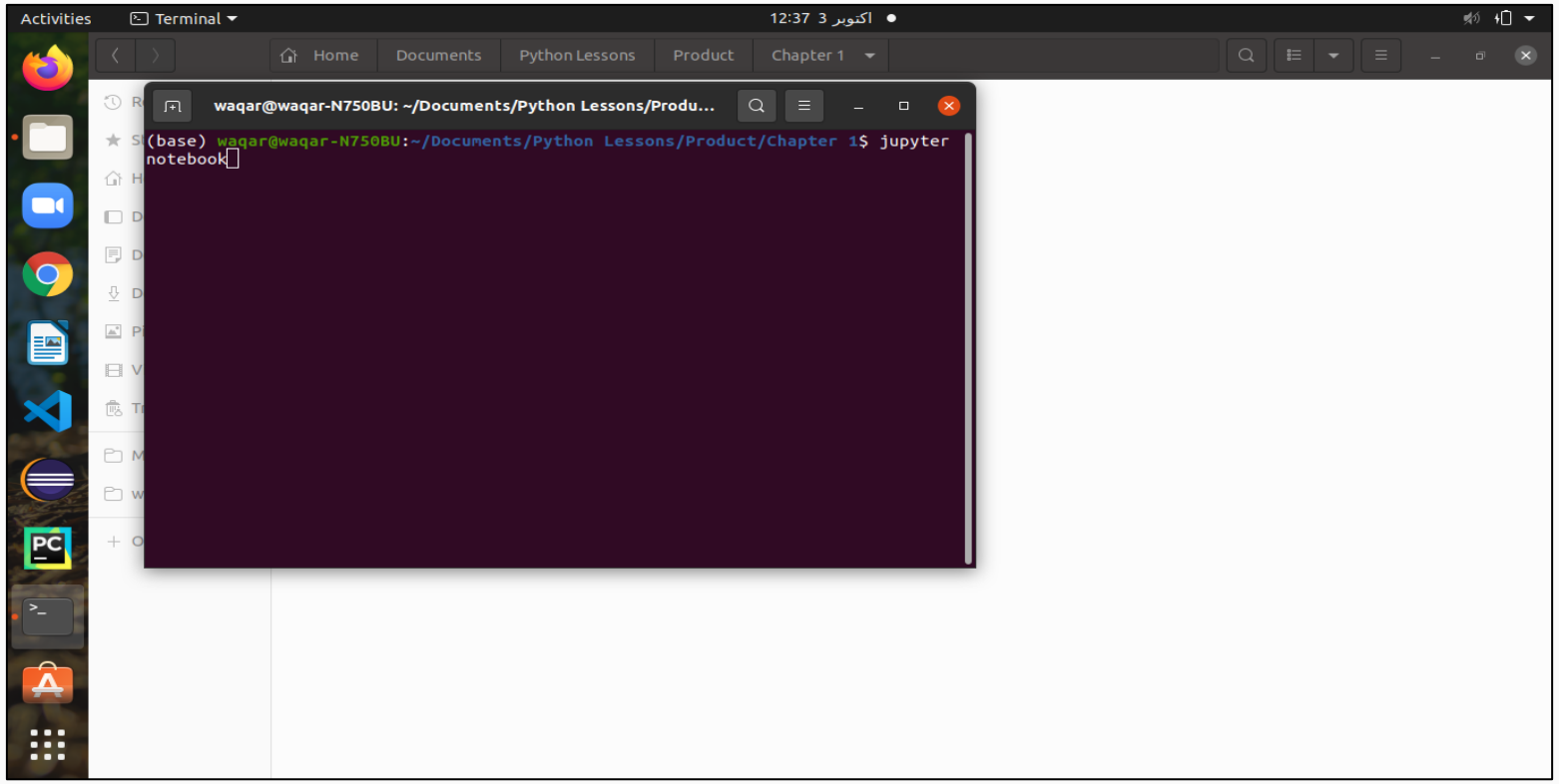
Managing Directories in Jupyter Notebook – Mac (3/3)

- ‘hello.ipynb’ contains a simple print statement.



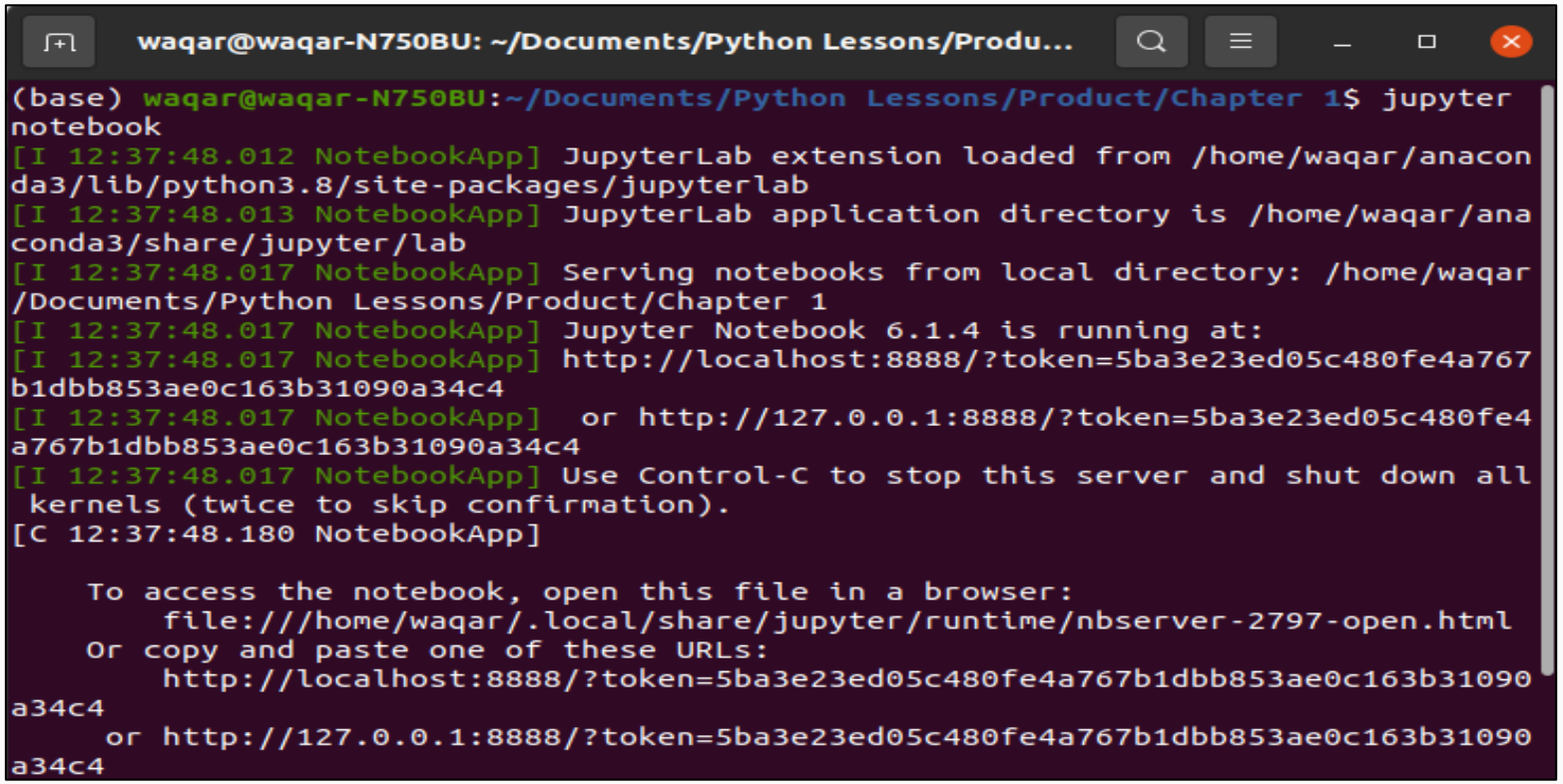
Managing Directories in Jupyter Notebook – Ubuntu (1/4)

- To open Jupyter Notebook in a particular directory/folder in Ubuntu, go to that directory, right click and select ‘open in terminal’.
- Type ‘jupyter notebook’ without quotation marks and hit Enter.



Managing Directories in Jupyter Notebook – Ubuntu (2/4)

- Wait for sometime.

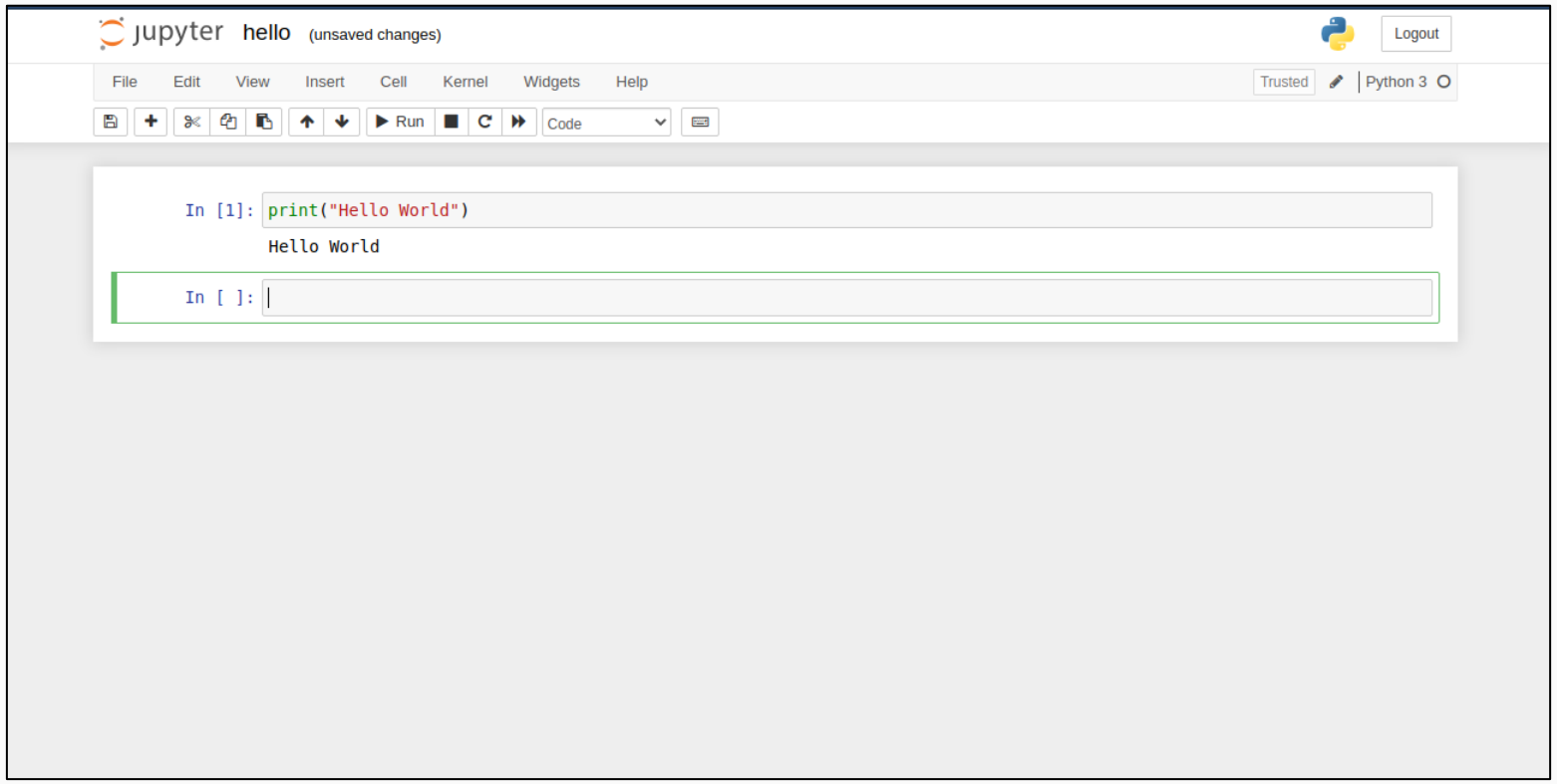


Managing Directories in Jupyter Notebook – Ubuntu (3/4)

- Jupyter Notebook will launch in your default browser.
- You can create a new Notebook or open an existing one.
- Let's open this 'hello.ipynb' file.

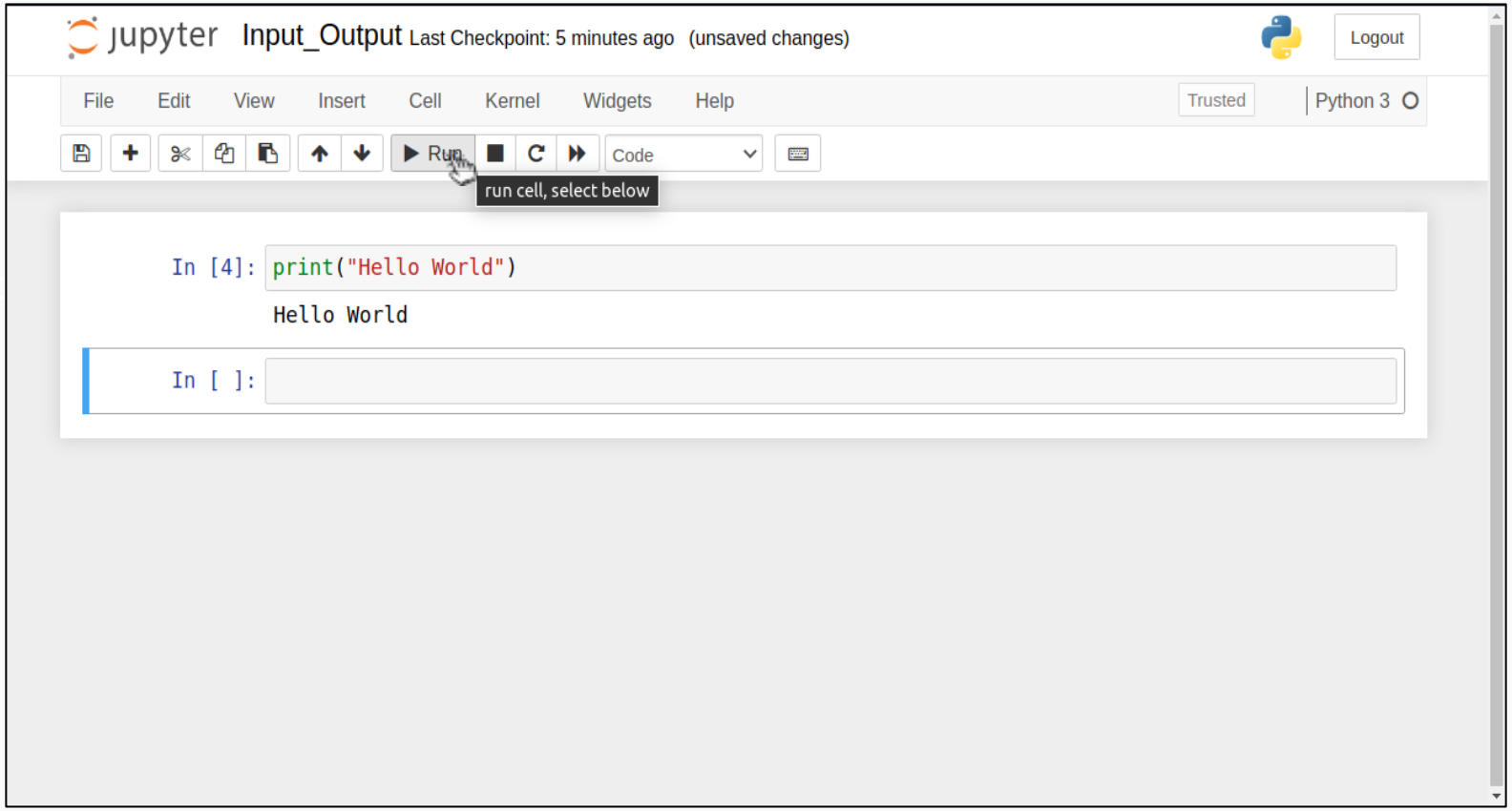
Managing Directories in Jupyter Notebook – Mac (4/4)

- ‘hello.ipynb’ contains a simple print statement.



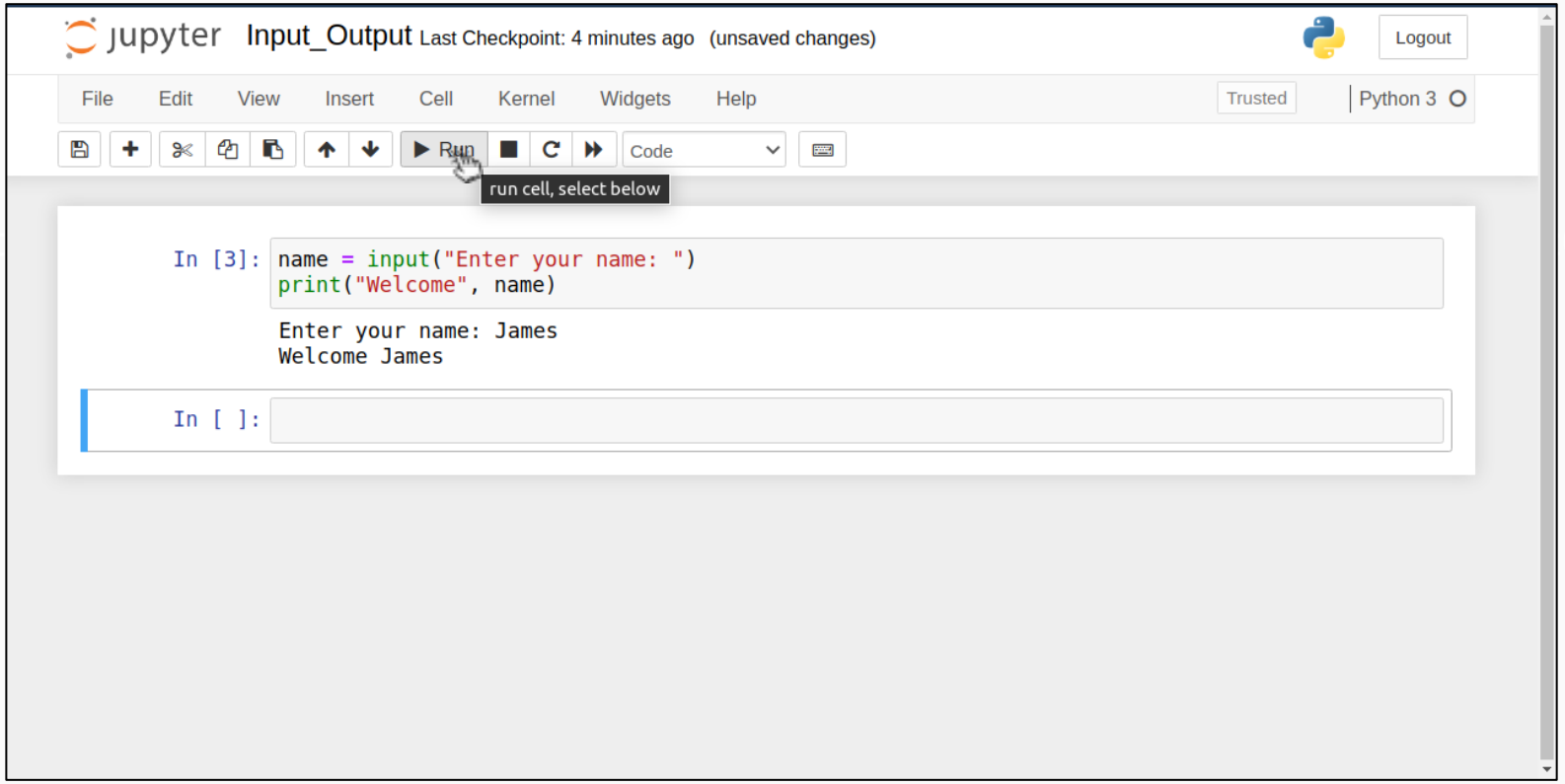
Output

Use the print() function to produce output.



Input

- Use the input() function to take input from the user.
- Here we are providing James as input.



Quiz Time

1. Which of the following is the correct input statement?
    a) Input(“Enter your name: “)
    b) input(“Enter your name: “)
    c) Input(Enter your name:)
    d) input(Enter your name:)

Quiz Time

1. Which of the following is the correct input statement?
    a) Input(“Enter your name: “)
    b) input(“Enter your name: “)
    c) Input(Enter your name:)
    d) input(Enter your name:)

Working with different data types

- Python has following primitive data types
  - int
  - float
  - string
  - bool

- Python also has following built-in non-primitive data types
  - Lists
  - Dictionaries
  - Tuples

Working with different data types

int

An int (integer) is a whole number such as 1, 5, 65, 1000

float

A float is a decimal point number such as 1.5, 7.8, 100.0

string

A string is a sequence of characters which can include alphabets, numbers, whitespaces, and special characters. In python, we use single, double or triple quotes to denote a string.

- E.g., ‘Hello World’, “Hello World”, and “’’Hello World’’’ are all valid strings

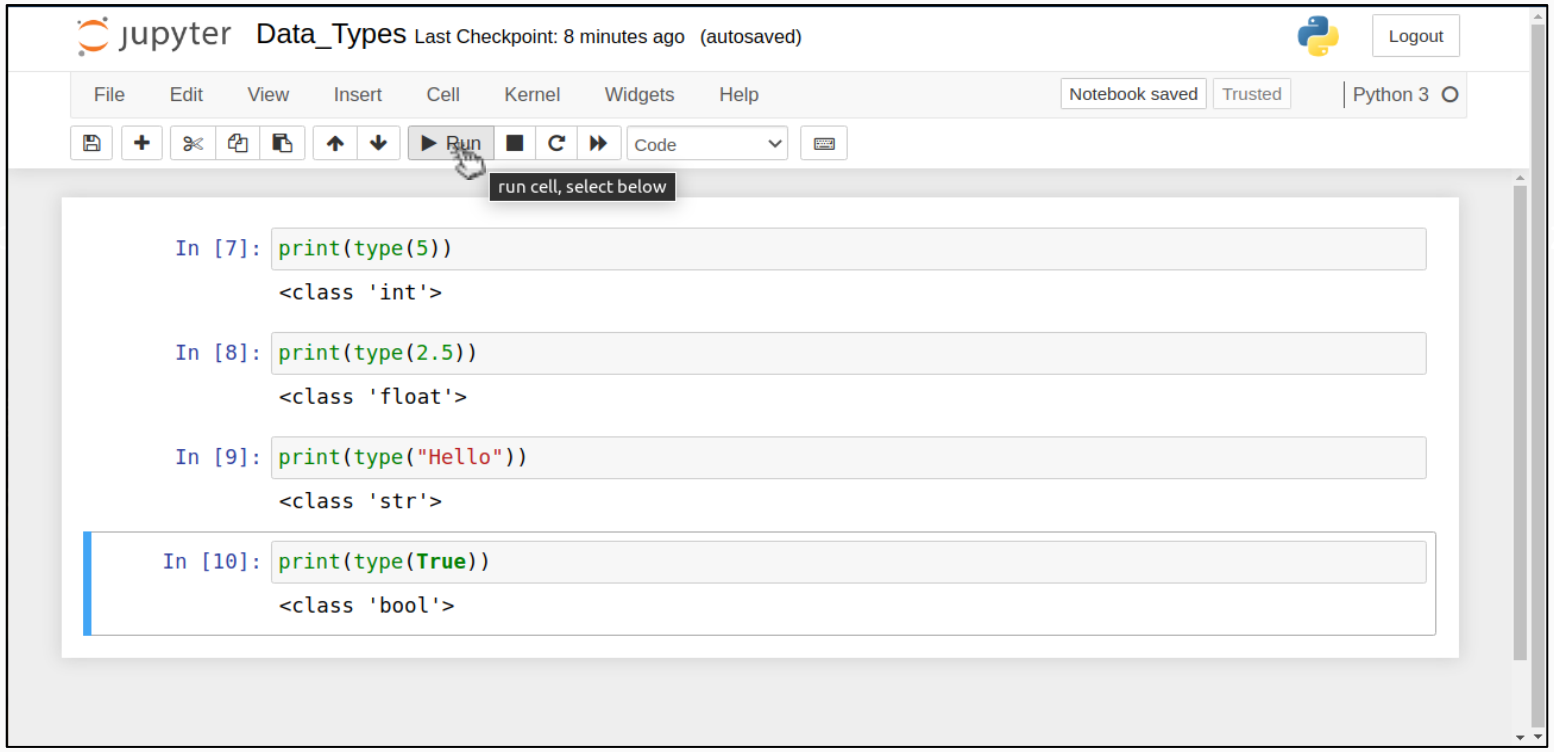
bool

A bool is either True or False

Working with different data types

type()

type() function is used to get the type of a value/variable.



Variables

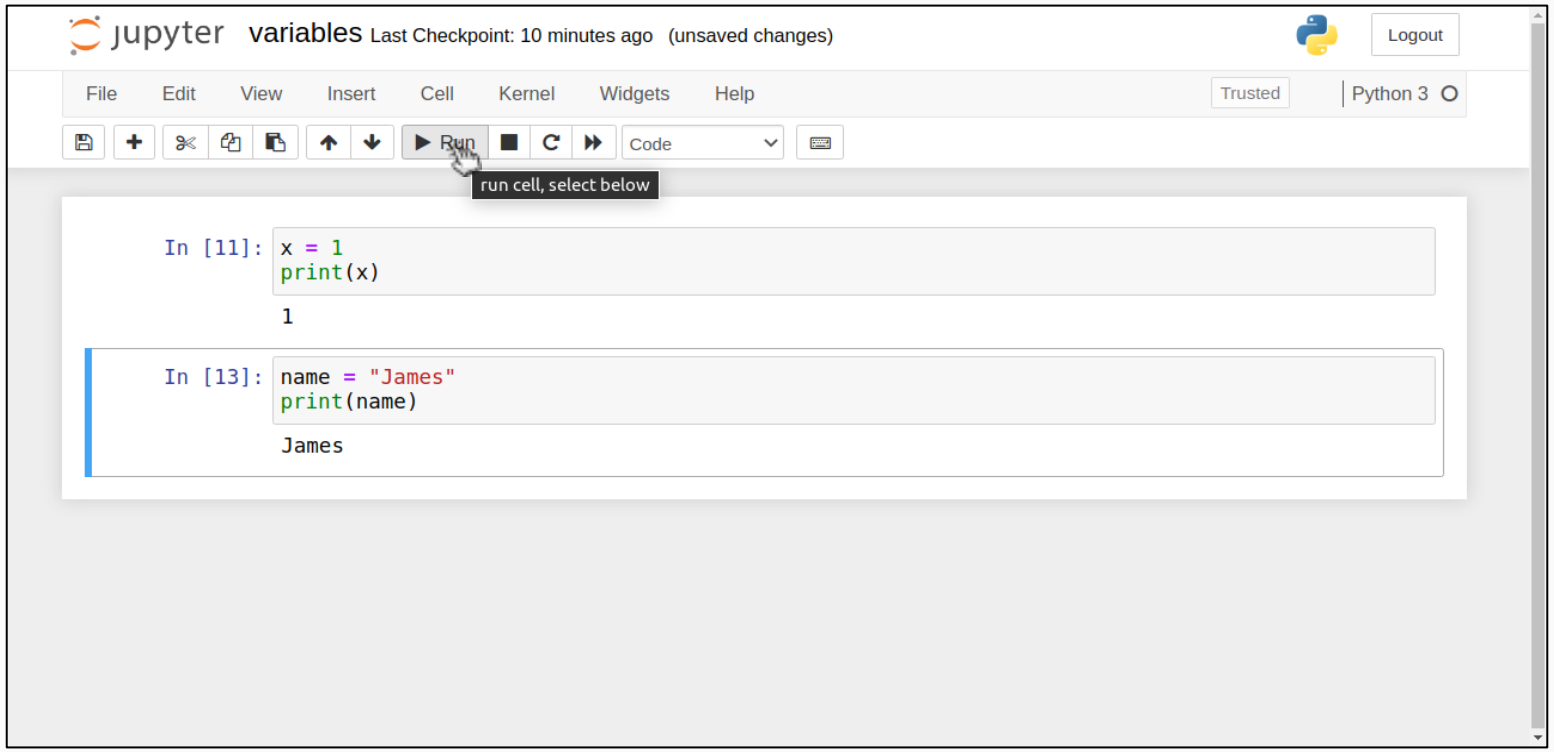
- Variables are placeholders for values.
- Python is a dynamically typed language, which means that the data types of variables are checked at runtime.
  - Hence, we do not need to declare the data type of a variable at the time of creating the variable.
- To print the value of a variable, pass the variable name to print() statement without quotation marks.

Rules for Variable Names

- Variable names can contain alphabets, integers, and/or underscore (_) character.
- Variable names can not begin with a number. They can only begin with a letter or an underscore (_).
- Variable names can not contain whitespaces and special characters.

Variables

Example in Jupyter



Quiz Time

1. Which of the following is a valid variable name?
    a) _name
    b) @name
    c) 1name
    d) n a m e

2. What is the correct way of printing the value of a variable called num?
    a) print(“num”)
    b) print(‘num’)
    c) print(“’num”’)
    d) print(num)

Quiz Time

1. Which of the following is a valid variable name?
   - **a) _name**
   - b) @name
   - c) 1name
   - d) n a m e

2. What is the correct way of printing the value of a variable called num?
   - a) print(“num”)
   - b) print(‘num’)
   - c) print(“’num”’)
   - **d) print(num)**

Arithmetic Operators (1/3)

Addition

Use the plus (+) operator for addition

Subtraction

Use the minus (-) operator for subtraction

Multiplication

Use the asterisk (*) operator for multiplication

Arithmetic Operators (2/3)

Division

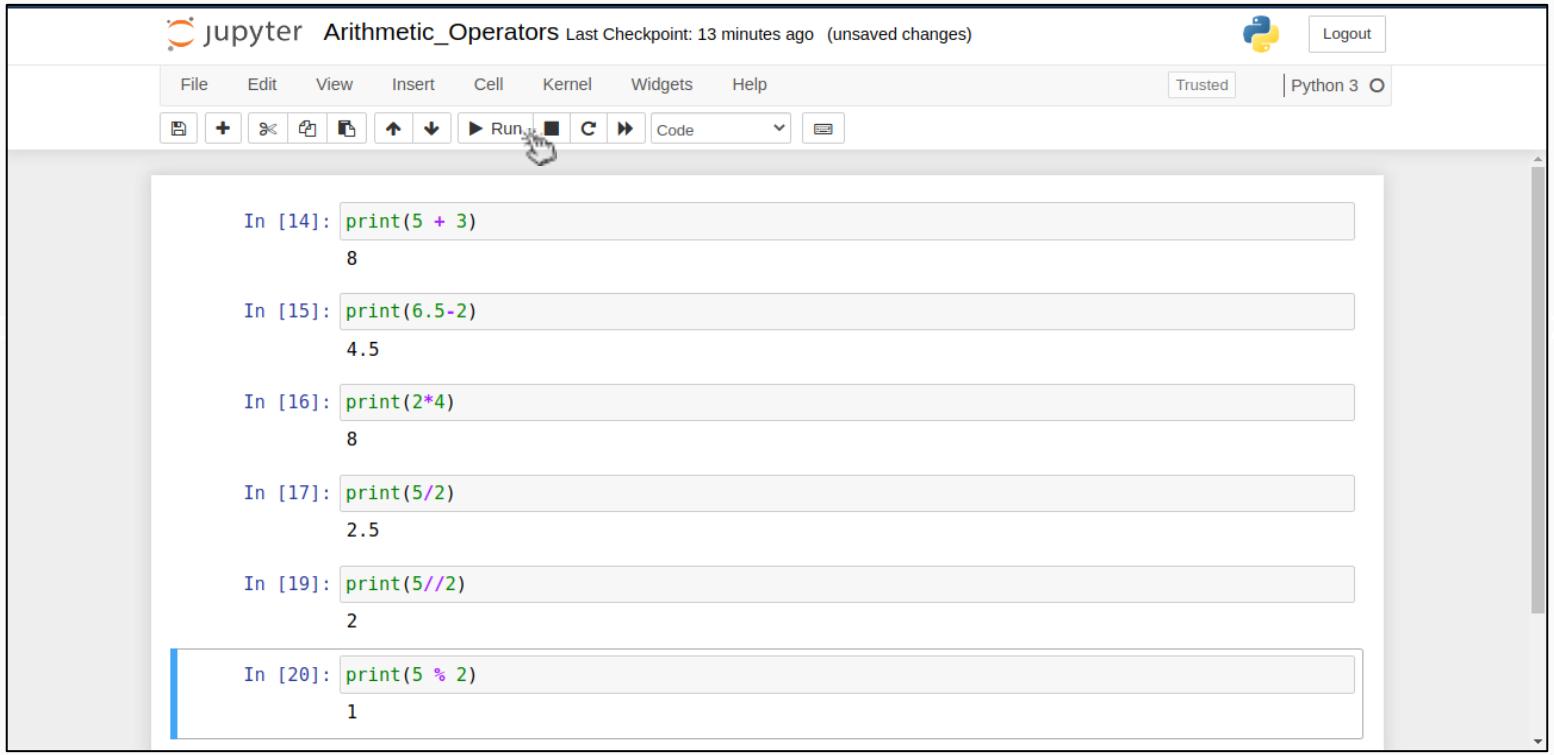
There are two types of division in Python

- Float Division: Use (/) for float division. If the result is an integer, it is converted to a float by adding .0
- Integer Division: Use (//) for integer division. If the result of is a float, it is converted to an integer by truncating the decimal part

Modulo

- Modulo operator returns the remainder of a division
- Use (%) operator modulo

Arithmetic Operators (3/3)



Quiz Time

1. What is the correct output of the float division 4/2?
    a) 2
    b) 2.0
    c) 0
    d) 1

2. What is the correct output of the integer division 4//2?
    a) 2
    b) 2.0
    c) 0
    d) 1

3. What is the correct output of the expression 9%2?
    a) 0
    b) 1
    c) 4.5
    d) 4

Quiz Time

1. What is the correct output of the float division 4/2?
    a) 2
    b) 2.0
    c) 0
    d) 1

2. What is the correct output of the integer division 4//2?
    a) 2
    b) 2.0
    c) 0
    d) 1

3. What is the correct output of the expression 9%2?
    a) 0
    b) 1
    c) 4.5
    d) 4

Comparison Operators

- Comparison operators are used to compare two values with each other.
- The result of a comparison operator is either True or False.
- We have the following comparison operators
  - Greater than (>)
    - returns True if the value on the left is bigger than that on the right
  - Less than (<)
    - returns True if the value on the left is smaller than that on the right
  - Greater than or equal to (>=)
    - returns True if the value on the left is either bigger than or equal to the value on the right
  - Less than or equal to (<=)
    - returns True if the value on the left is either smaller than or equal to the value on the right
  - Not equal to (!=)
    - returns True if the value on the left is not equal to the value on the right
  - Is equal to (==)
    - returns True if the value on the left is equal to the value on the right

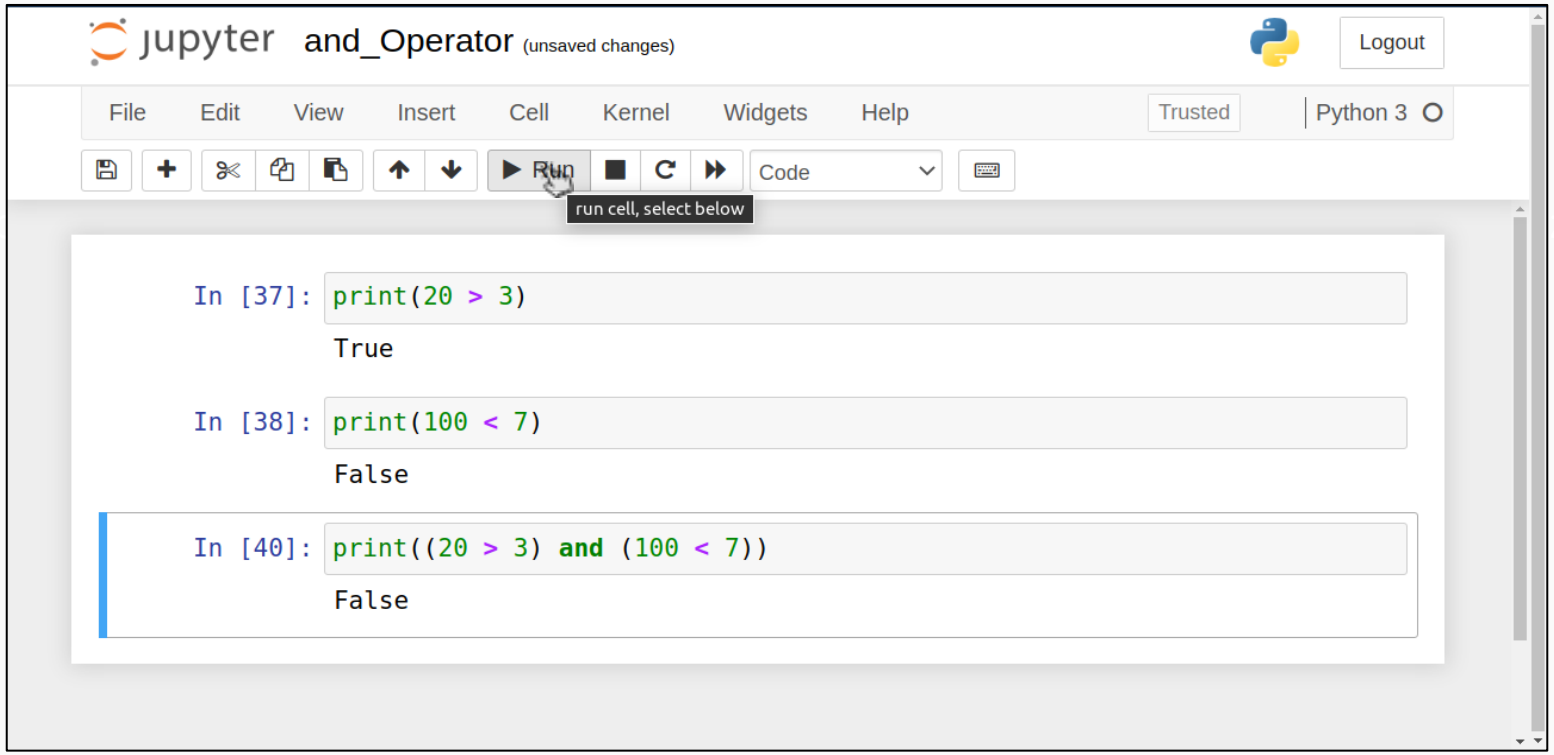
Logical Operators

- We can use logical operators to combine two or more conditions.
- Python has three logical operators
  - And operator (and)
  - Or operator (or)
  - Not operator (not)
- and/or operators are binary as they take two operands.
- Not is a unary operator as it takes only one operand.

Logical Operators

and operator

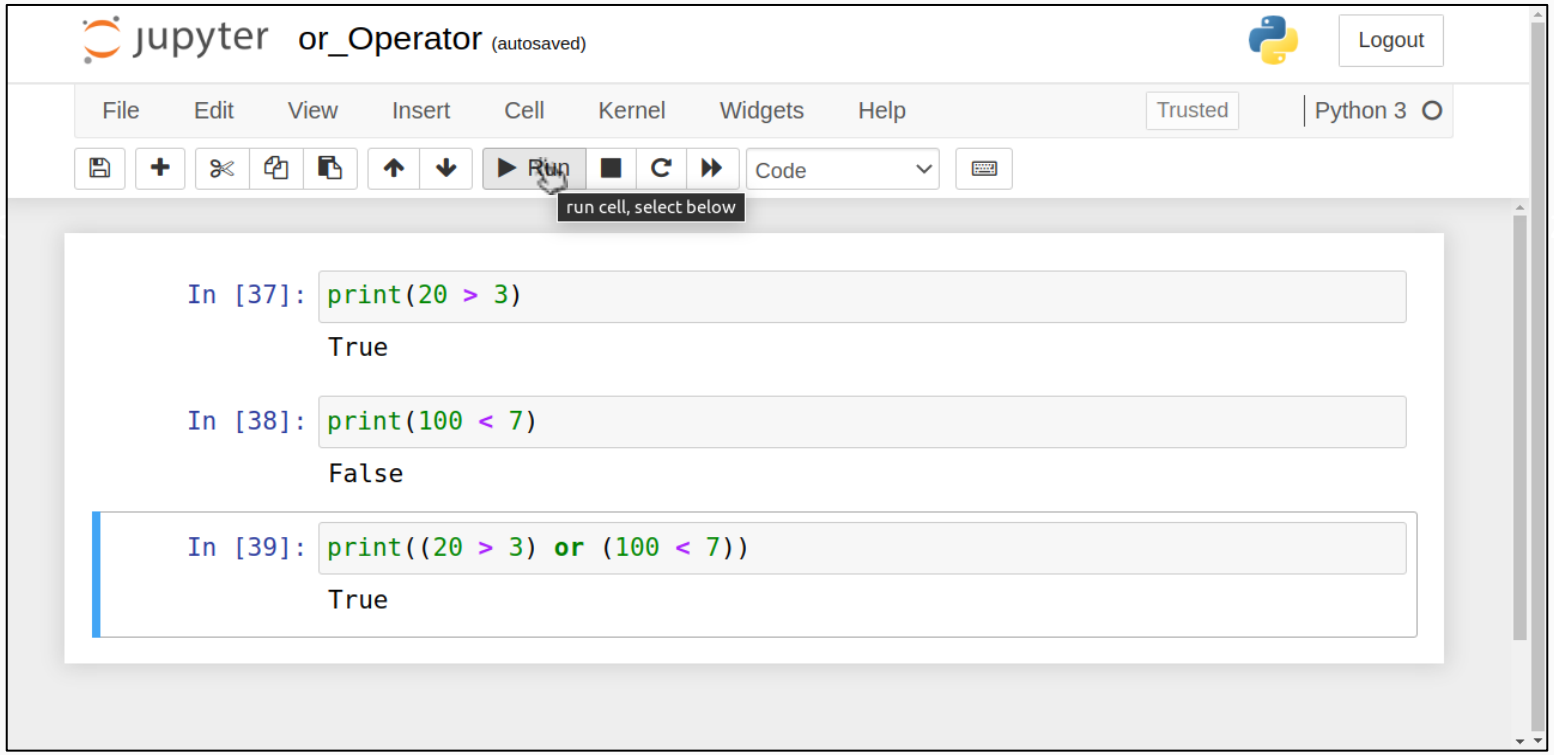
- and returns a True only if both the operands are True.
- In python, we use the keyword ‘and’ to perform and operation.



Logical Operators

or operator

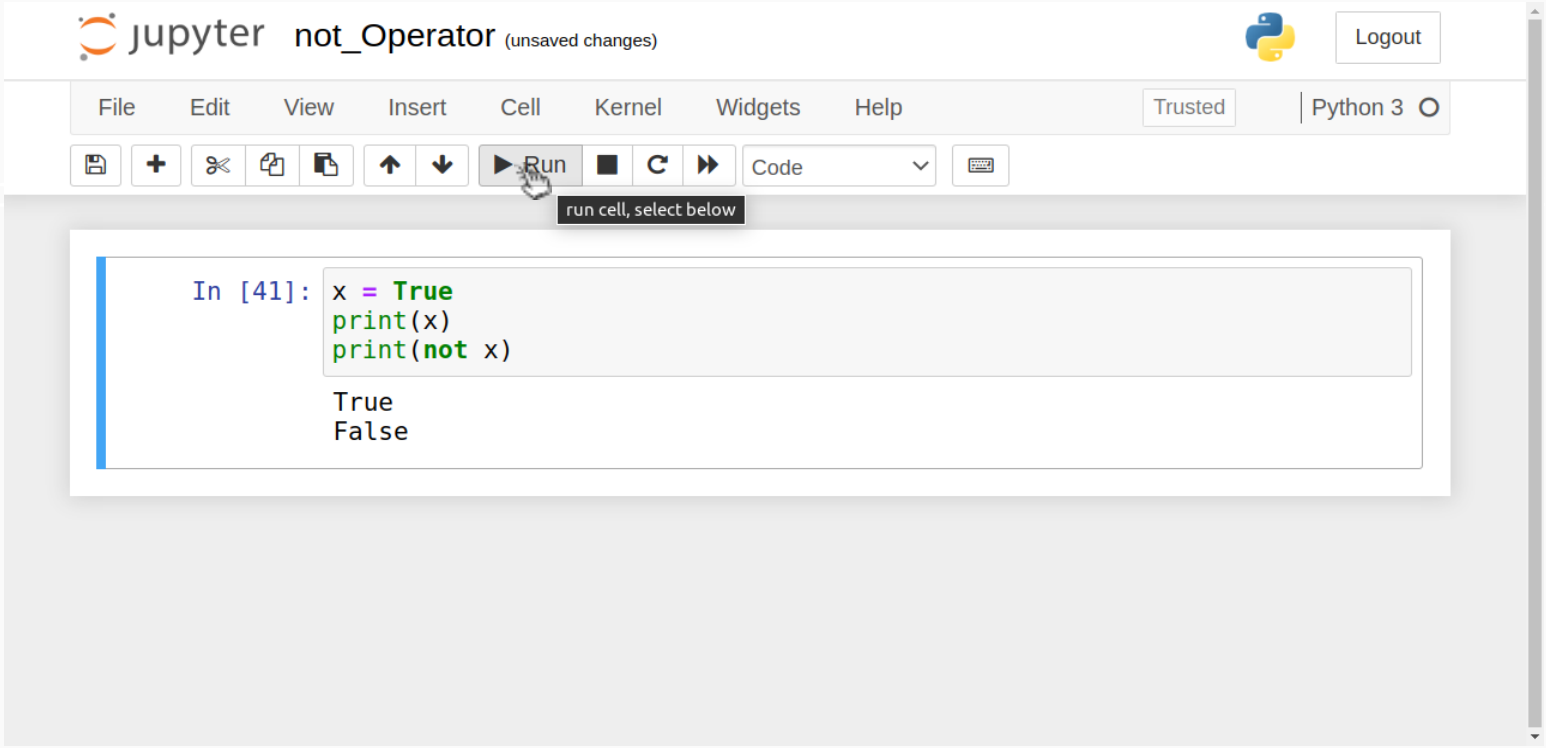
- or returns a True if at least one of the operands is True.
- In python, we use the keyword ‘or’ to perform or operation.



Logical Operators

not operator

- not is a unary operator.
- not changes True to False and vice versa.
- In python, we use the keyword ‘not’ to perform not operation.



Quiz Time

1. What is the correct output of the expression; (4 >3) and (5>3)?
    a) True
    b) False

2. What is the correct output of the expression; (7 < 2) or (8 < 3)?
    a) True
    b) False

3. What is the correct output of the expression; not True?
    a) True
    b) False

Quiz Time

1. What is the correct output of the expression; (4 >3) and (5>3)?
   - **a) True**
   - b) False

2. What is the correct output of the expression; (7 < 2) or (8 < 3)?
   - a) True
   - **b) False**

3. What is the correct output of the expression; not True?
   - a) True
   - **b) False**

Conditional Statements (1/4)

- Conditional statements are used to make decisions.
- In Python, we make decisions using if-else and elif statements.
- elif and else are both optional, however they must follow and if statement.
- We can have as many elif statements as we want. But there is only one if and one else statement.

Conditional Statements (2/4)

Control Flow

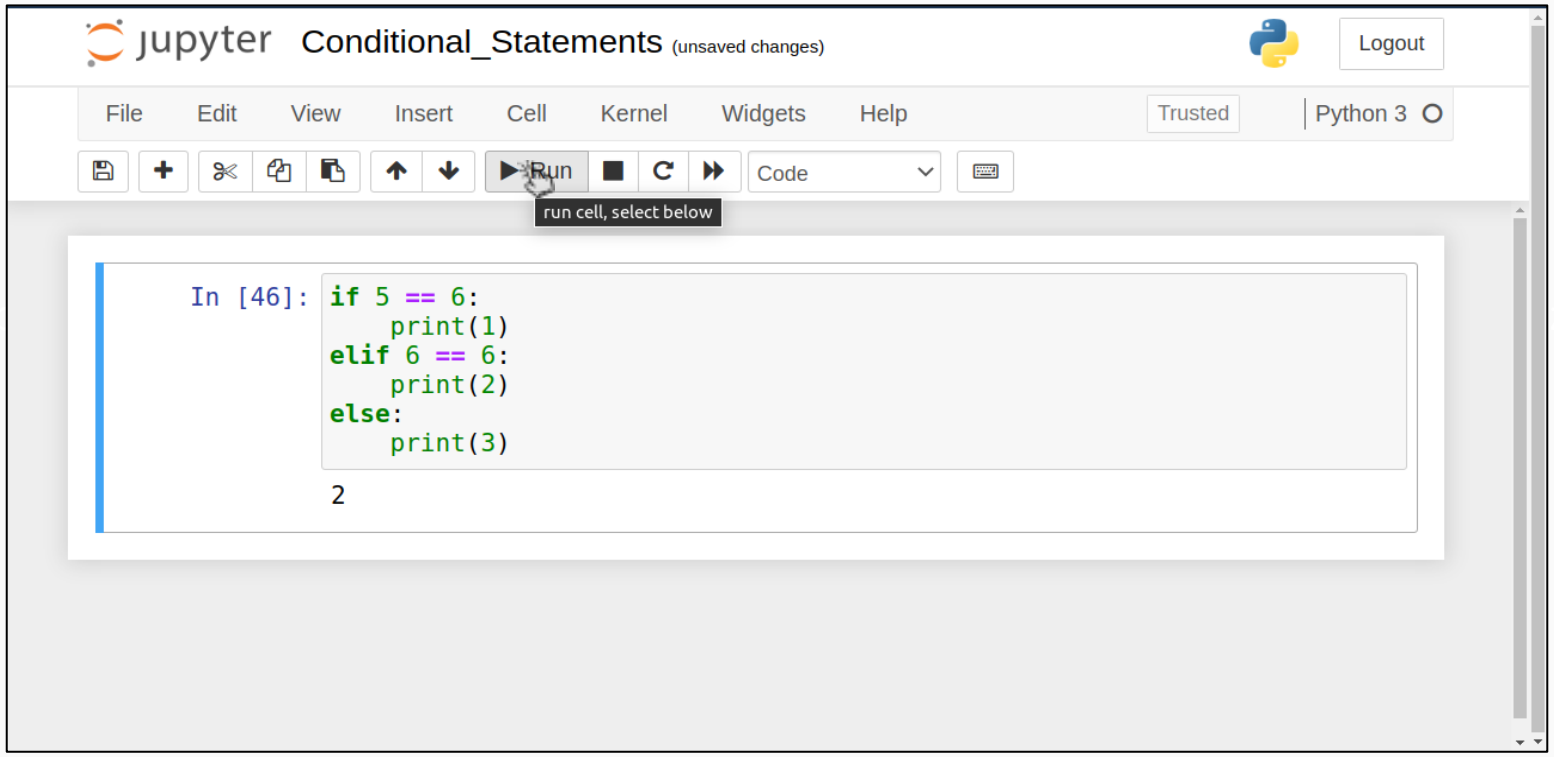
- Here is the general syntax of conditional statements

```
if condition:
        block of statements to execute
elif condition:
        block of statements to execute
else:
        block of statements to execute
```


- Conditions are evaluated in a top-down manner.

Conditional Statements (3/4)

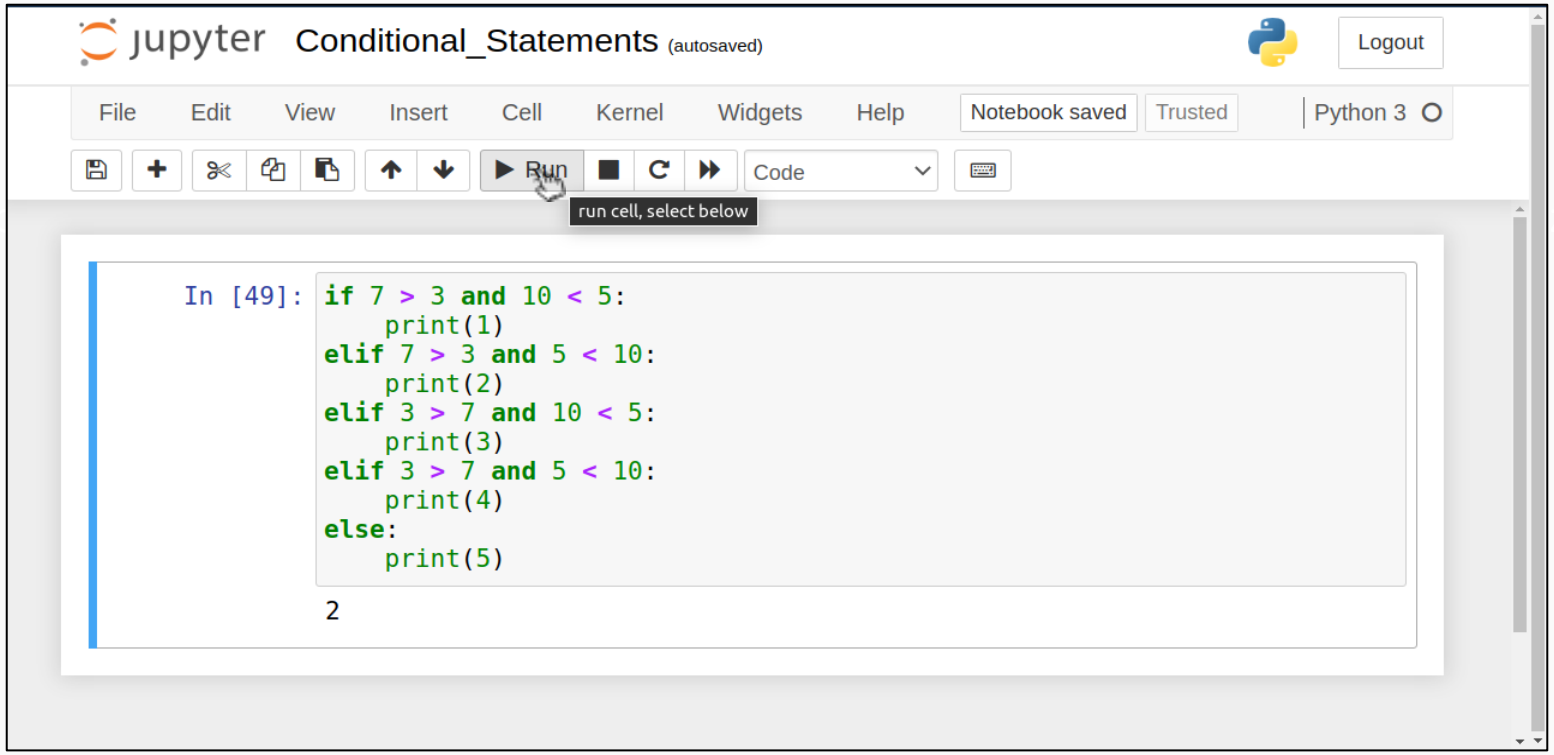
Example in Jupyter



Conditional Statements (4/4)

Combining Conditions

- We can combine two or more conditions together using logical operators.



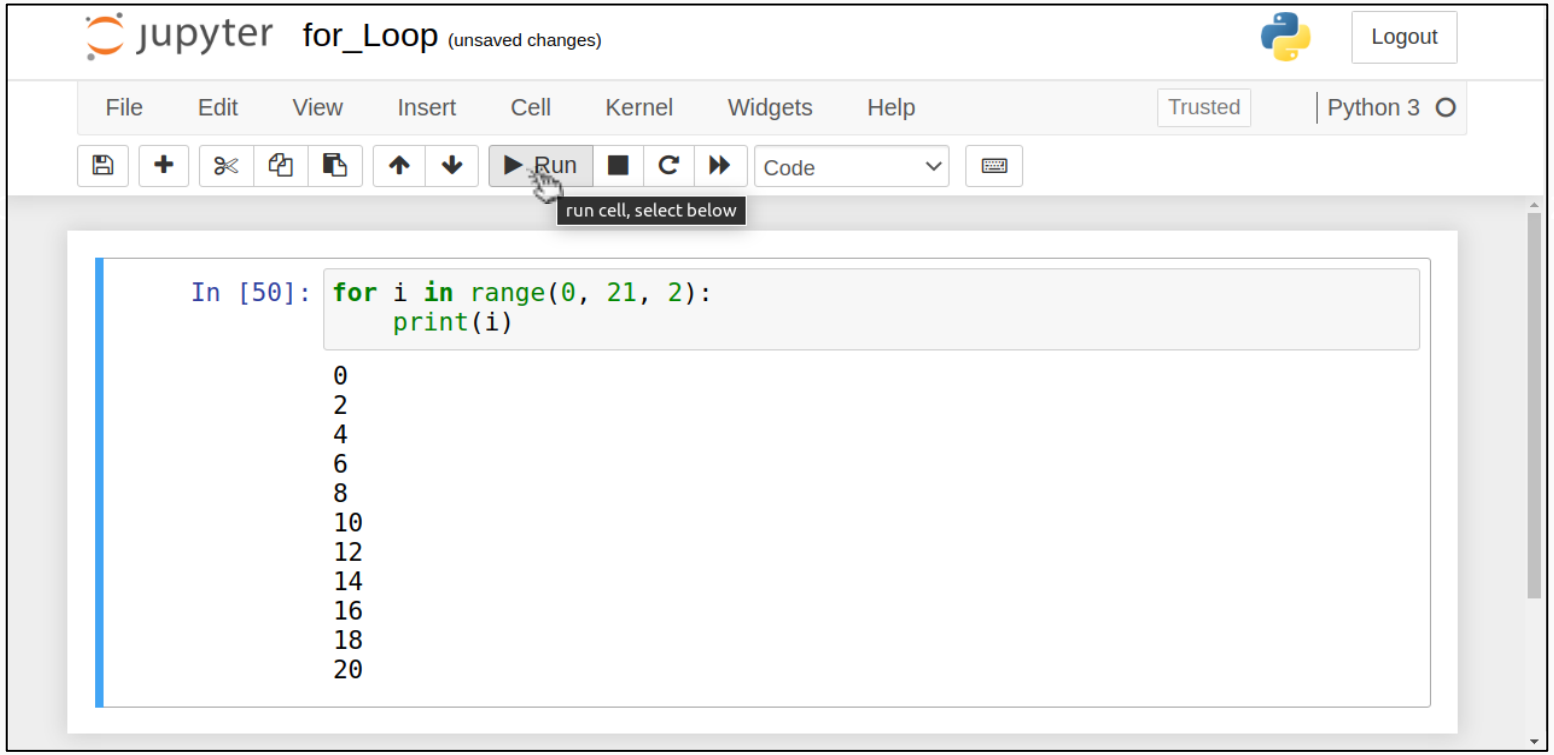
Loops

- Loops are used for repeating a set of statements.
- There are two types of loops in Python
  - For loops
  - While loops

https://unibo.zoom.us/j/83422537997?pwd=Z3VENmI0QTV2Q2xJWDlPQzhrVU5PZz09

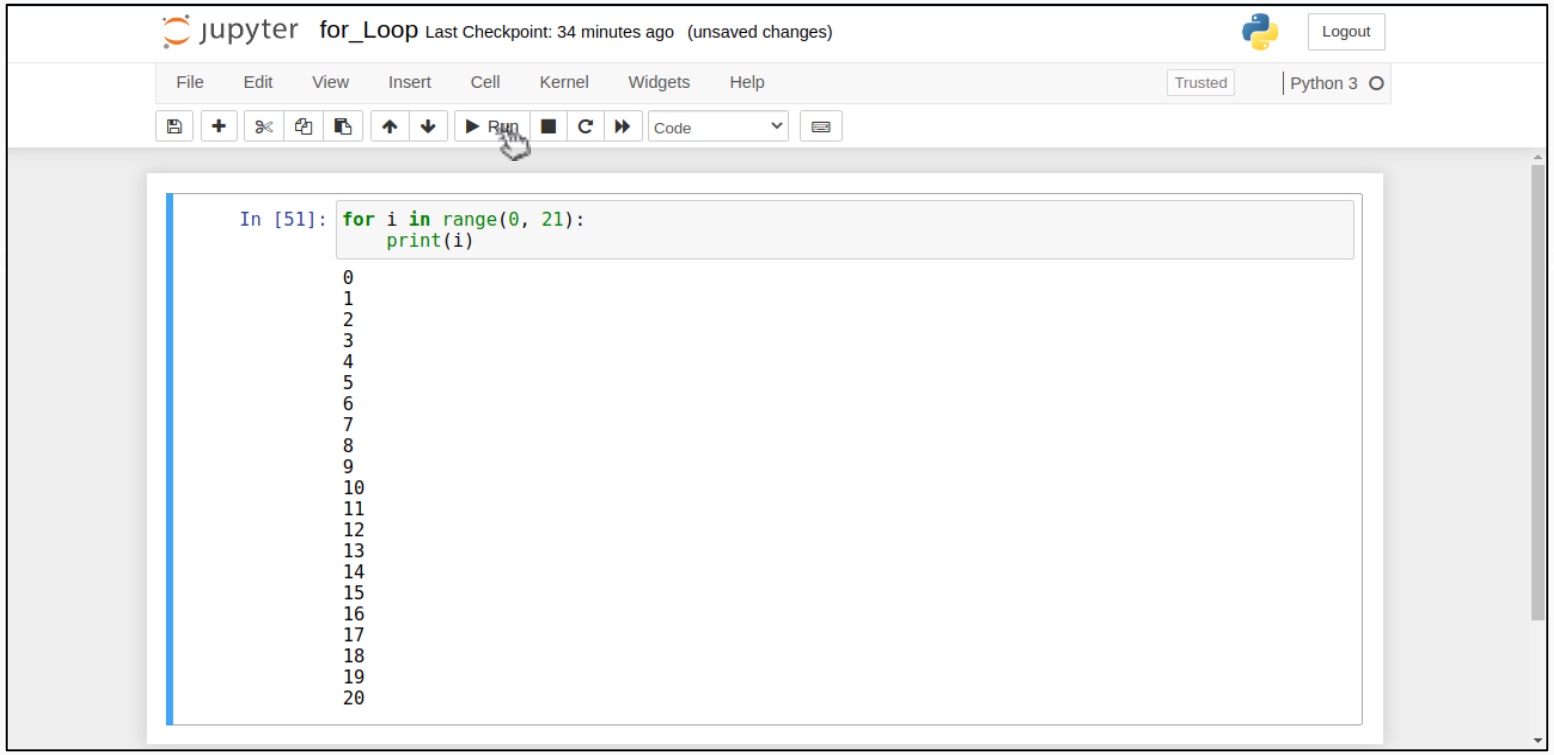
for Loop (1/2)

- for loop takes a range() function.
- range() takes three arguments; start, stop, step.
- loop runs from start to stop - 1 with an increment equal to step after every iteration.



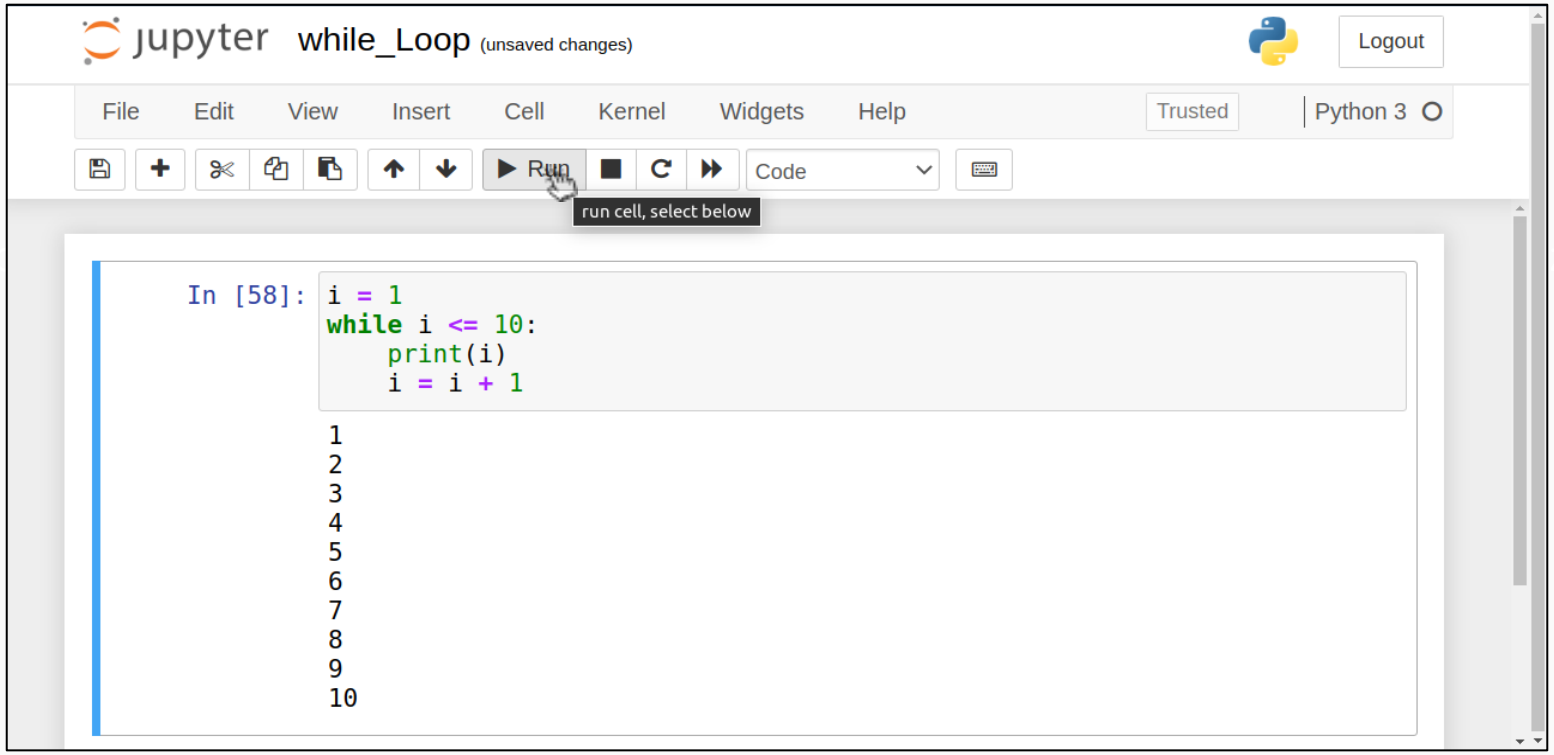
for Loop (2/2)

- If we omit the step value, it is set to 1 by default.



while Loop (1/2)

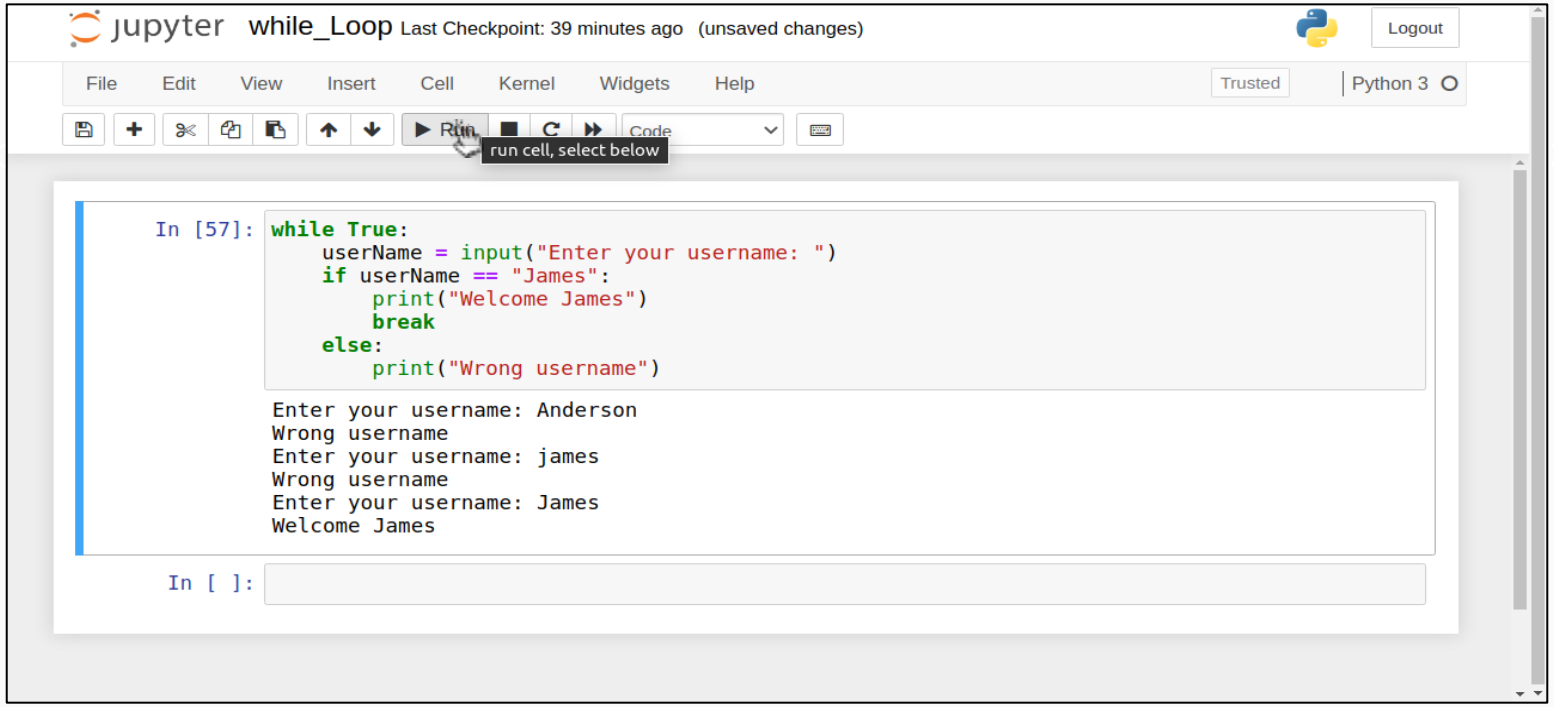
- while loop takes a condition and runs as long as the condition is True.



while Loop (2/2)

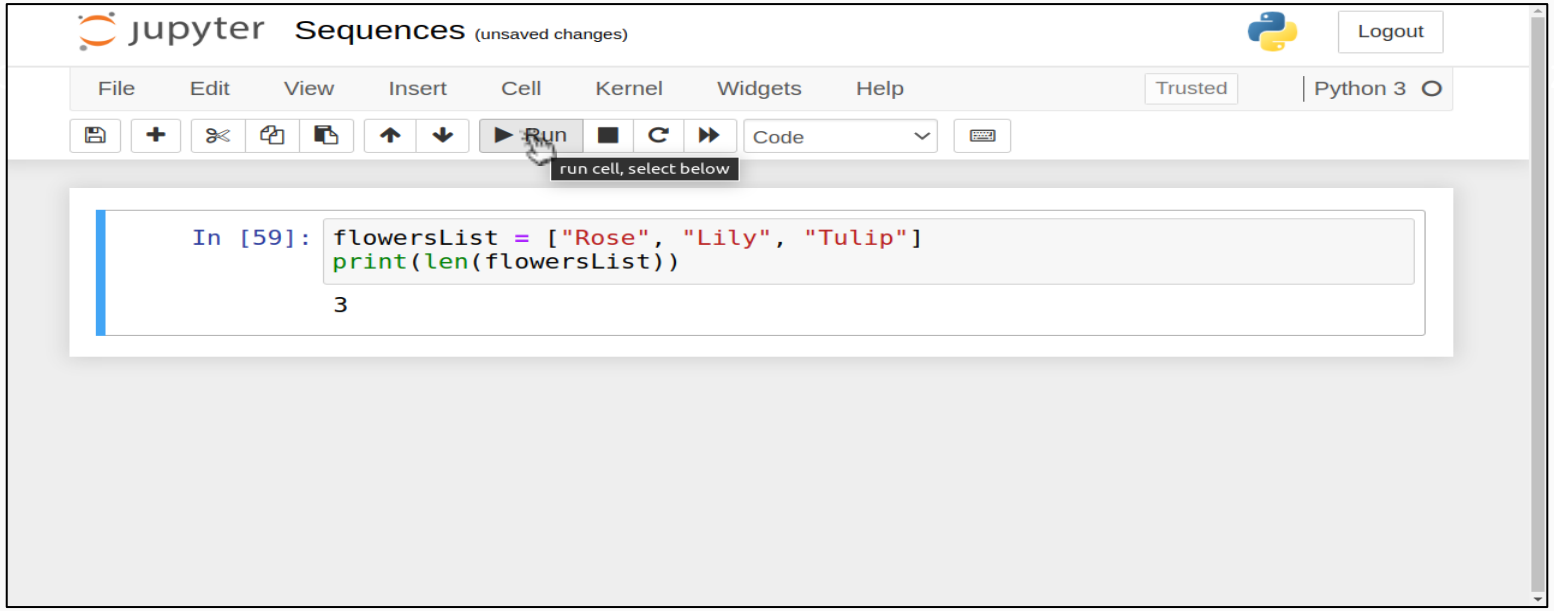
Infinite Loop

- We can create an infinite loop using a while loop.
- We give a terminating condition inside the infinite loop and use the keyword ‘break’ to break out of the infinite loop.



Sequences

- Python has following built-in sequences (non-primitive data types)
  - Lists
  - Dictionaries
  - Tuples
- We can use the len() function to find the total number of elements in any sequence.



Lists (1/5)

- Lists are used to store multiple values.
- A list can have values of different data types.
- Lists are ***mutable*** i.e., they can be modified.

Lists (2/5)

List Indexing

- We can get the element at a particular position in the list using list indexing.
  - Remember, indices start at 0 and go until len(list) - 1.

Rose	Lily	Tulip
0	1	2

- Consider the list above that contains names of three flowers. Hence, it's length is 3.
- If we call this list flowersList, then;
  - flowersList[0] will give us Rose
  - flowersList[1] will give us Lily
  - flowersList[2] will give us Tulip

Lists (3/5)

List Slicing

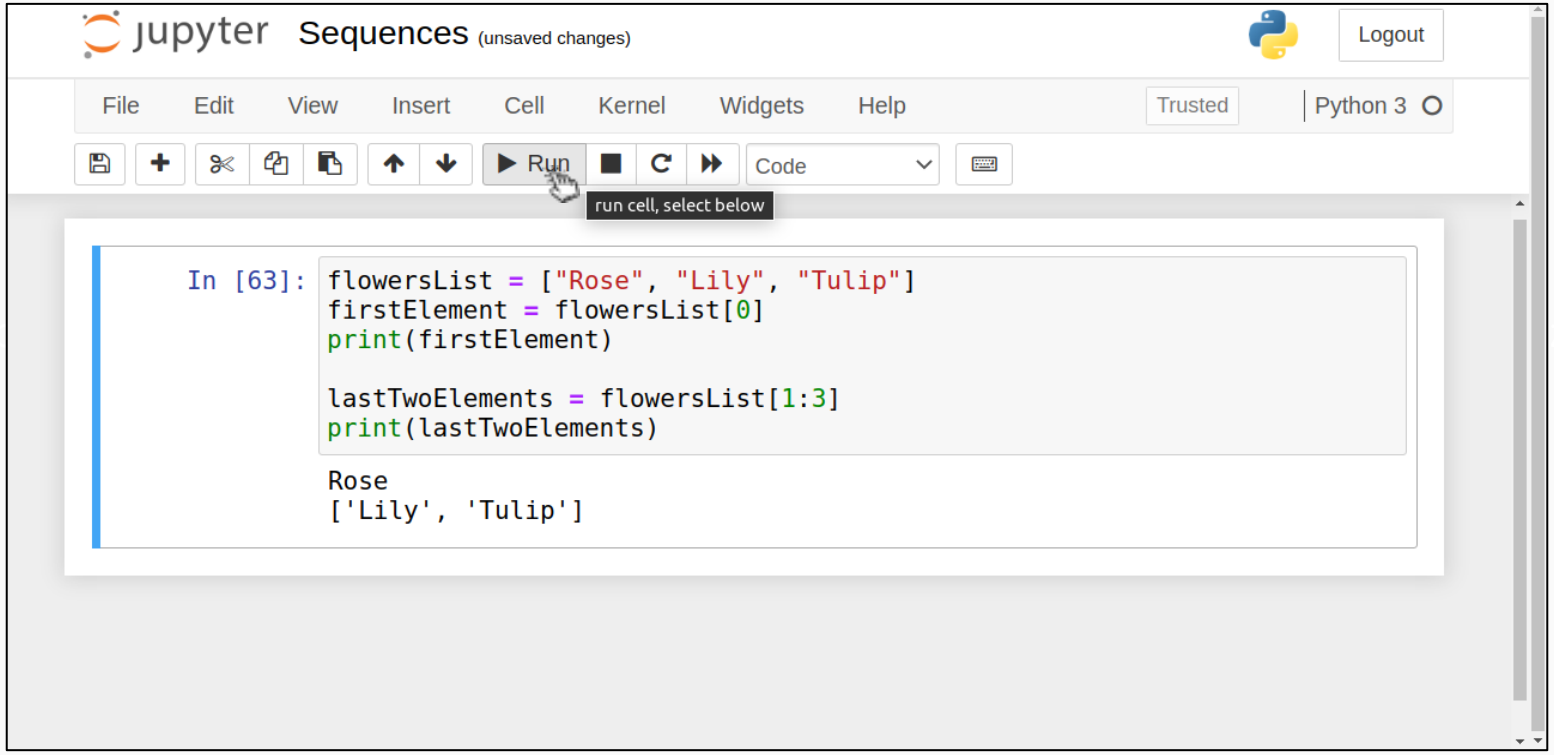
- We can also slice a list.
- For slicing a list, we give a starting-index (inclusive) and an end-index (exclusive).

Rose	Lily	Tulip
0	1	2

- If we call this list flowersList then;
  - flowersList[0:2] will give us [Rose, Lily]
  - flowersList[0:3] will give us [Rose, Lily, Tulip]
  - flowersList[1:2] will give us [Lily]

Lists (4/5)

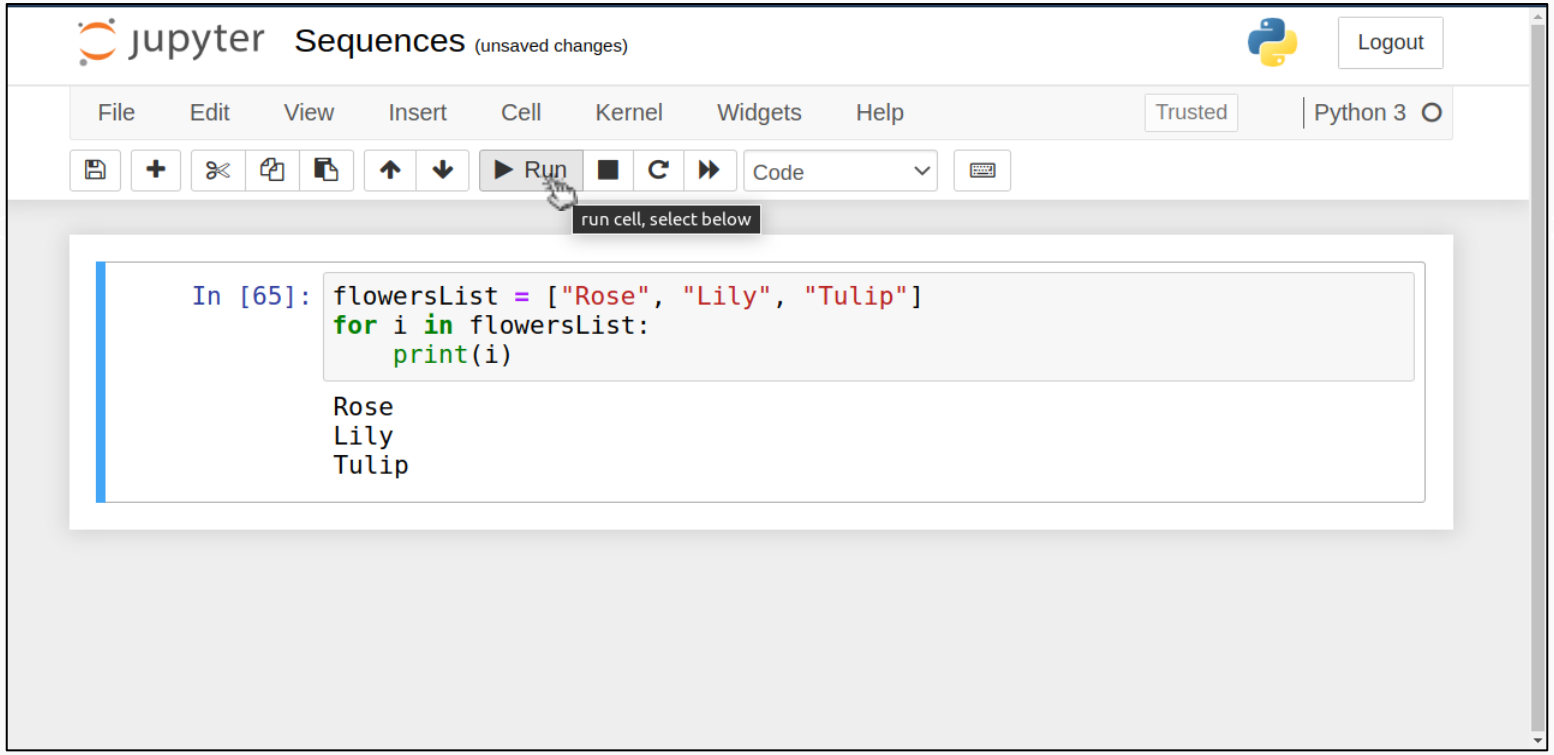
Example in Jupyter



Lists (5/5)

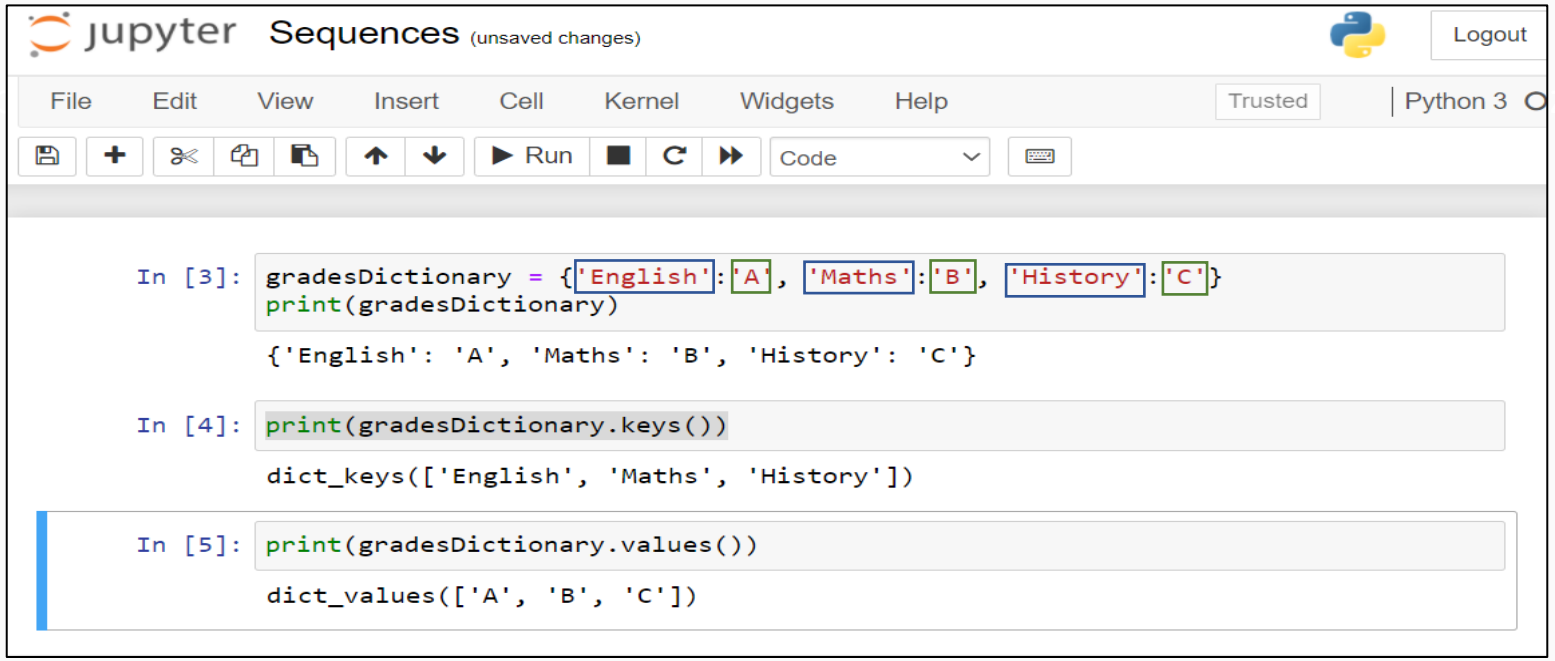
Iterating Over List

- We can loop over a list using a modified form of for loop.



Dictionaries (1/3)

- Dictionaries are used to store key, value pairs.
- A key and a value are separated by a colon (:).
- Each key, value pair is separated by a comma (,).
- Dictionaries are ***mutable*** i.e., they can be modified.
- We can use .keys() function to get all the keys and .values() function to get all the values in the dictionary.

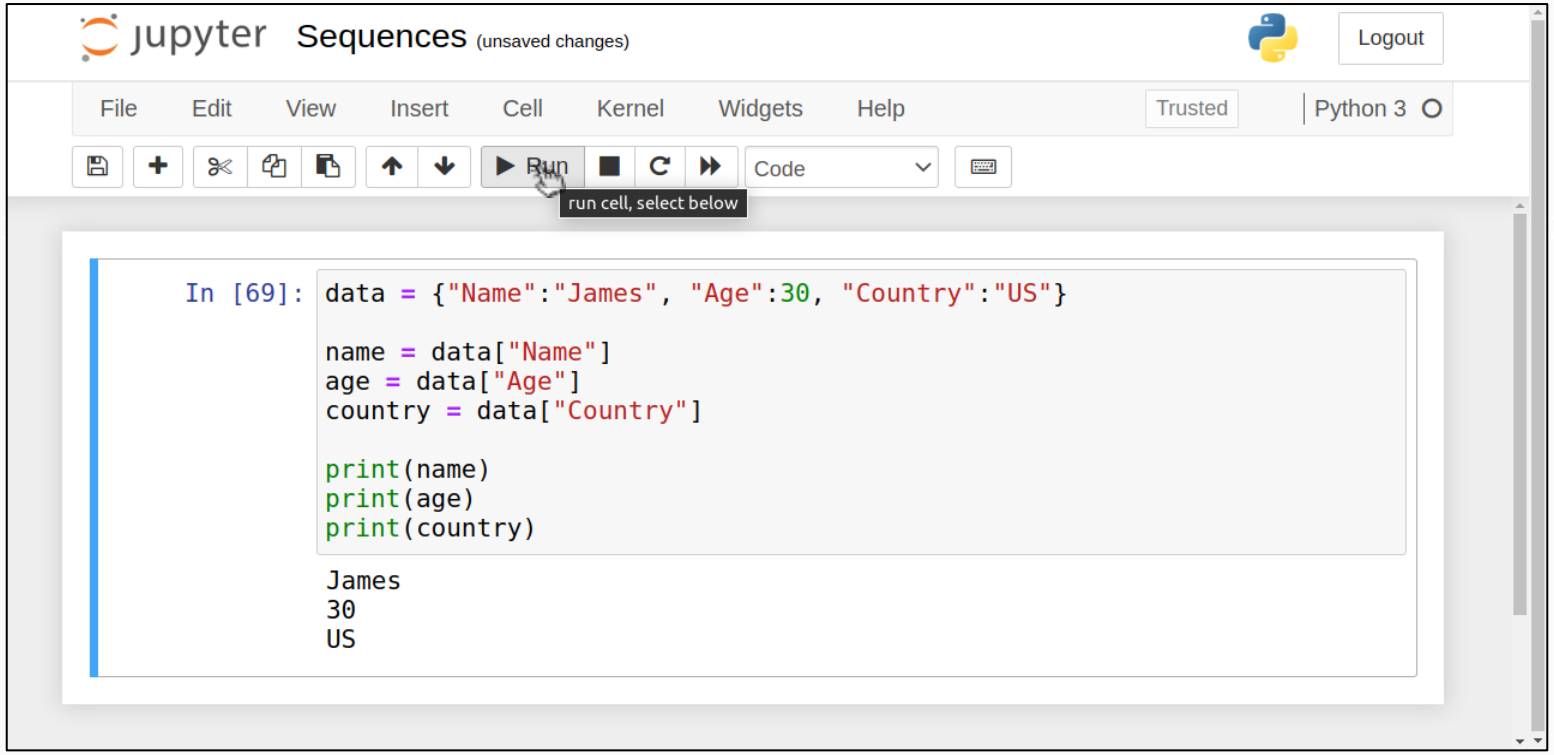


Keys

Values

Dictionaries (2/3)

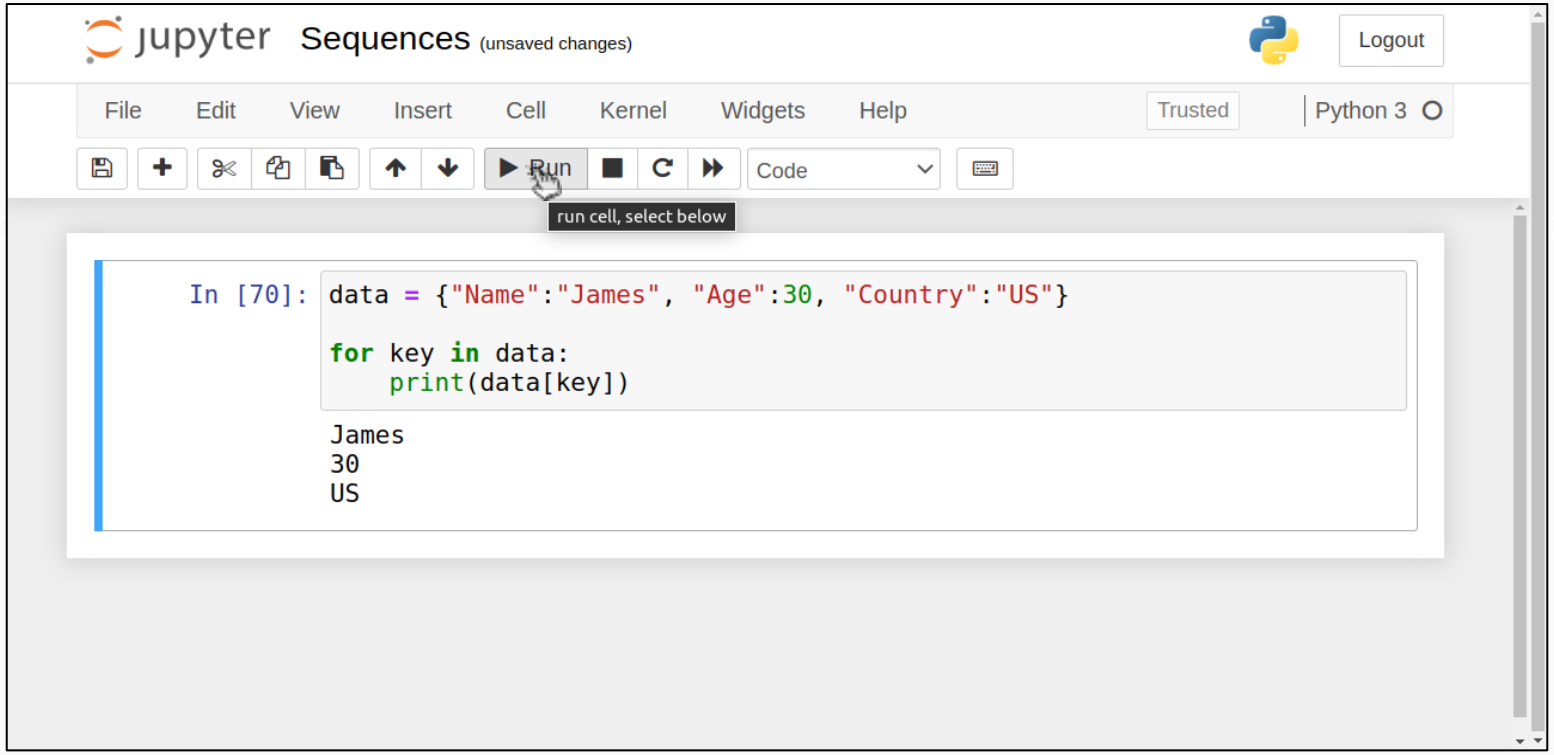
- We can get a value associated with a key by passing the key.



Dictionaries (3/3)

Iterating Over Dictionary

- We can get all the key, value pairs in a Dictionary using a modified form of for loop.



Tuples (1/5)

- Tuples are also used to store multiple values.
- Tuples can have values of different data types.
- Tuples are ***immutable*** i.e., they can not be modified.

Tuples (2/5)

Tuple Indexing

- We can index a tuple just as we can index a list.

Rose	Lily	Tulip
0	1	2

- Consider the tuple above that contains names of three flowers. Hence, it's length is 3.
- If we call this tuple flowersTuple, then;
  - flowersTuple[0] will give us Rose
  - flowersTuple[1] will give us Lily
  - flowersTuple[2] will give us Tulip

Tuples (3/5)

Tuple Slicing

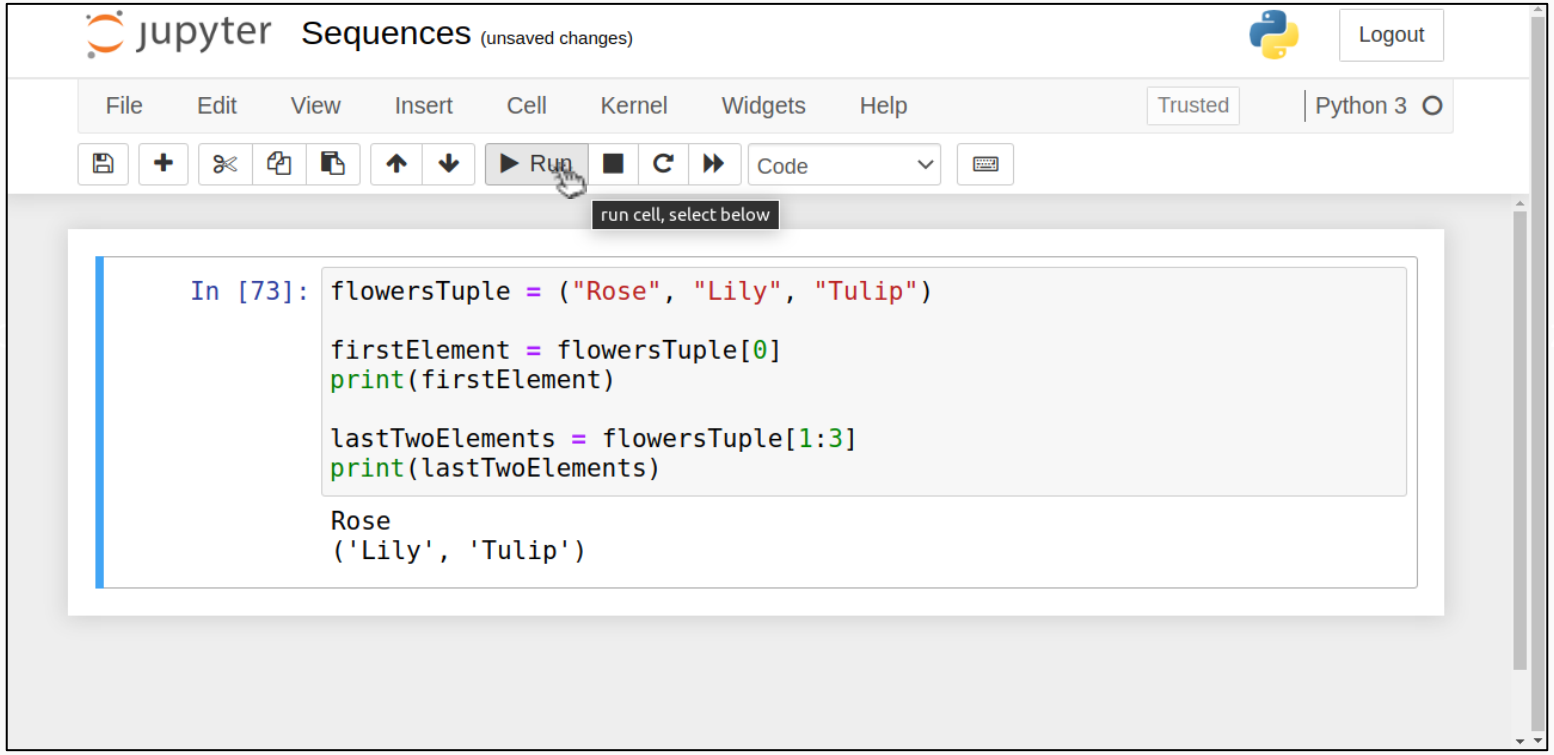
- Tuple slicing works the same way as list slicing.

Rose	Lily	Tulip
0	1	2

- If we call this tuple flowersTuple then;
  - flowersTuple[0:2] will give us (Rose, Lily)
  - flowersTuple[0:3] will give us (Rose, Lily, Tulip)
  - flowersTuple[1:2] will give us (Lily)

Tuples (4/5)

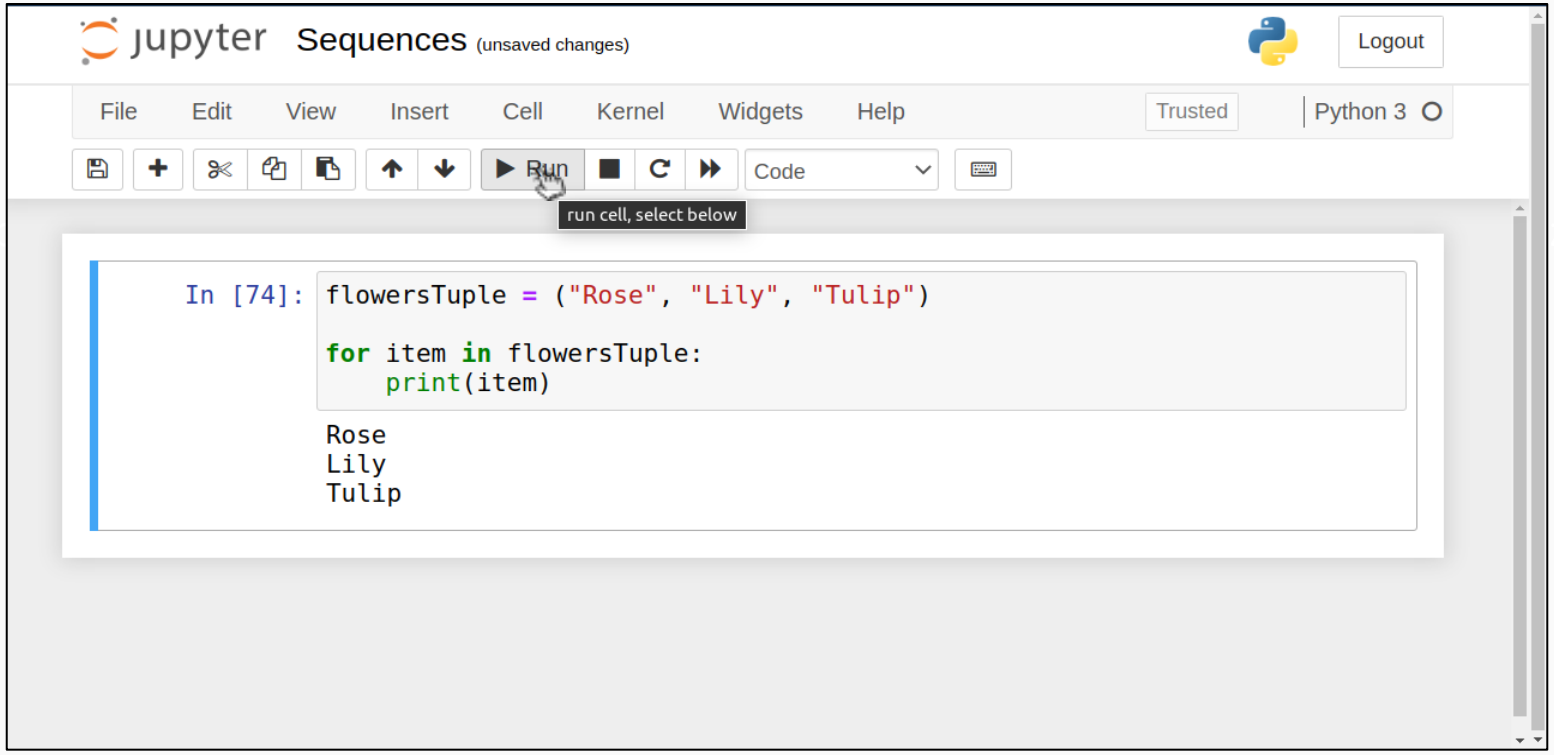
Example in Jupyter



Tuples (5/5)

Iterating Over Tuple

- We can loop over a tuple using a modified form of for loop.



Quiz Time

1. Which of the following is the immutable sequence?
    a) List
    b) Dictionary
    c) Tuple

2. Consider a list x = [“red”, 1, True]. What will x[1] give?
    a) “red”
    b) 1
    c) True

3. Consider a dictionary y = {“name”:”James”, “age”:1,}. What is “age” in this dictionary?
    a) Key
    b) Value
    c) None

Quiz Time

1. Which of the following is the immutable sequence?
    a) List
    b) Dictionary
    c) Tuple

2. Consider a list x = [“red”, 1, True]. What will x[1] give?
    a) “red”
    b) 1
    c) True

3. Consider a dictionary y = {“name”:”James”, “age”:1,}. What is “age” in this dictionary?
    a) Key
    b) Value
    c) None

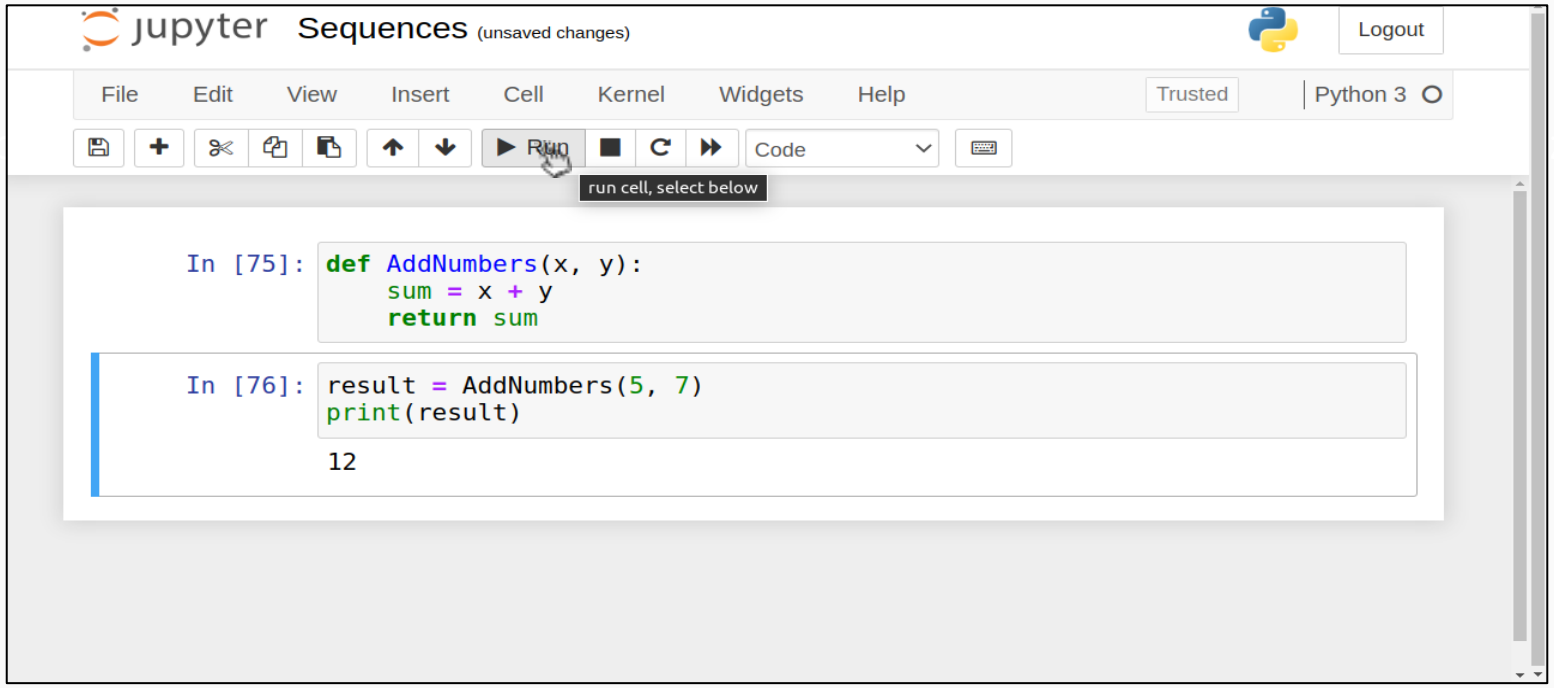
Built-in Functions

- Python has a lot of built-in functions e.g.,
  - len(), which returns the length of a sequence
  - abs(), which returns the absolute value
  - max(), which returns the maximum value in a sequence
- You can find a detailed list of built-in functions in Python at the following link:
  https://docs.python.org/3/library/functions.html

https://unibo.zoom.us/j/83422537997?pwd=Z3VENmI0QTV2Q2xJWDlPQzhrVU5PZz09

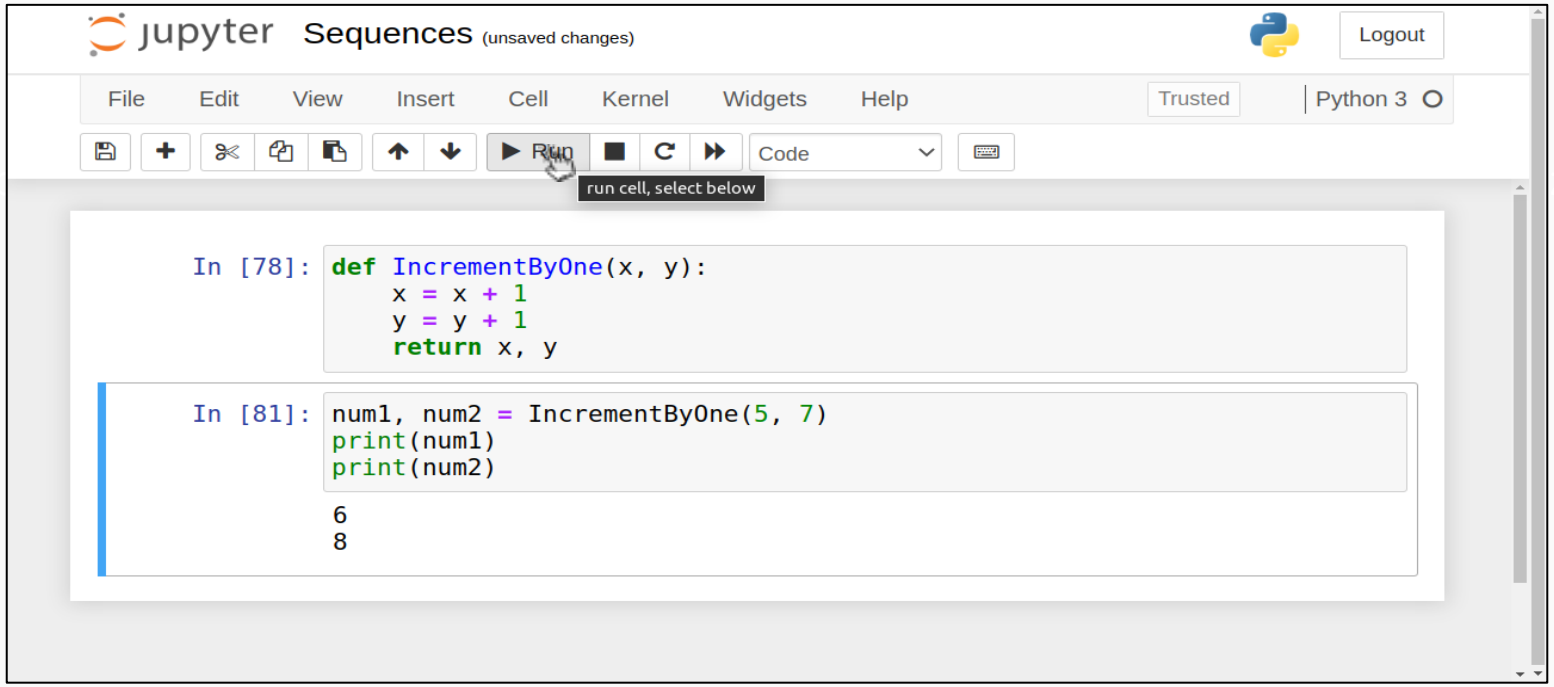
User-defined Functions (1/2)

- Apart from built-in functions, users can create their own functions as well.
- We use the ‘def’ keyword to define a function.
- A function should be defined before it is called.
- We do not need to declare the return type while defining a function.



User-defined Functions (2/2)

- We can also return more than one value from a function.
- In this case we are returning 2 values from the functions, hence we need to store them in 2 variables, namely num1 and num2.



Quiz Time

1. A function should be defined
    a) After it is called
    b) Before it is called
    c) Doesn't matter

2. A function in Python can only return 1 value. Is this statement True or False?
    a) True
    b) False

Quiz Time

1. A function should be defined
    a) After it is called
    b) Before it is called
    c) Doesn't matter

2. A function in Python can only return 1 value. Is this statement True or False?
    a) True
    b) False

Resources

Here are a few resources for you to revise Python syntax

- https://www.w3schools.com/python/
- https://www.youtube.com/watch?v=rfscVS0vtbw
- https://www.tutorialspoint.com/python/index.htm

References

- How to install python on Mac.

https://www.youtube.com/watch?v=M323OL6K5vs

- How to install Anaconda in Ubuntu

https://www.youtube.com/watch?v=dGm10q_y3xw

- How to install Anaconda in Mac

https://www.youtube.com/watch?v=PJWnGNWQ1Dc

- Managing Directories in Jupyter Noteook – Mac

https://www.youtube.com/watch?v=G1Vm6wR0XIQ