PYTHON 

INTRODUCTION
INSTALLATION - INSTALL ON WINDOWS - INSTALL ON MAC
CODE EDITOR - PYCHARM INSTALLATION
PYTHON SYNTEX - FIRST PROGRAM - VARIABLES - DATA TYPES
USER INPUT - IF ELSE CONDITION - PRACTICE : QUIZ GAME
WHILE LOOP - PRACTICE : GUESS THE NUMBER
LIST - FOR LOOP
SET - TUPLES
DICTIONARY - PRACTICE - REFACTOR QUIZ GAME
FUNCTIONS - PRACTICE - REFACTOR QUIZ GAME WITH FUNCTION
MODULES & PACKAGES - EXCEL AUTOMATION
OBJECT ORIENTED PROGRAMMING
DATA ANALYSIS & VISUALISATION WITH PYTHON
PROJECT - DATA ANALYSIS & VISUALISATION WITH PYTHON

INTRODUCTION

Python is a programming language, just like java or javascript. It is very easy to learn and very easy to get started with.

Advantage -

- Easy to learn
  - Simple syntax
  - Easy to setup

- Large Ecosystem
  - large community
  - many community

- Multi purpose language
  - It is flexible language

- Makes you more valuable at your work 😎😎

Usage -

Famous Usages -

Web Development | **Machine Learning**

Data Science | **Web Scrolling**

Artificial Intelligent | **Automation**

Less Likely Usage as better alternatives are available -

- Mobile Development
- Game Development
- Desktop Development

INSTALLATION

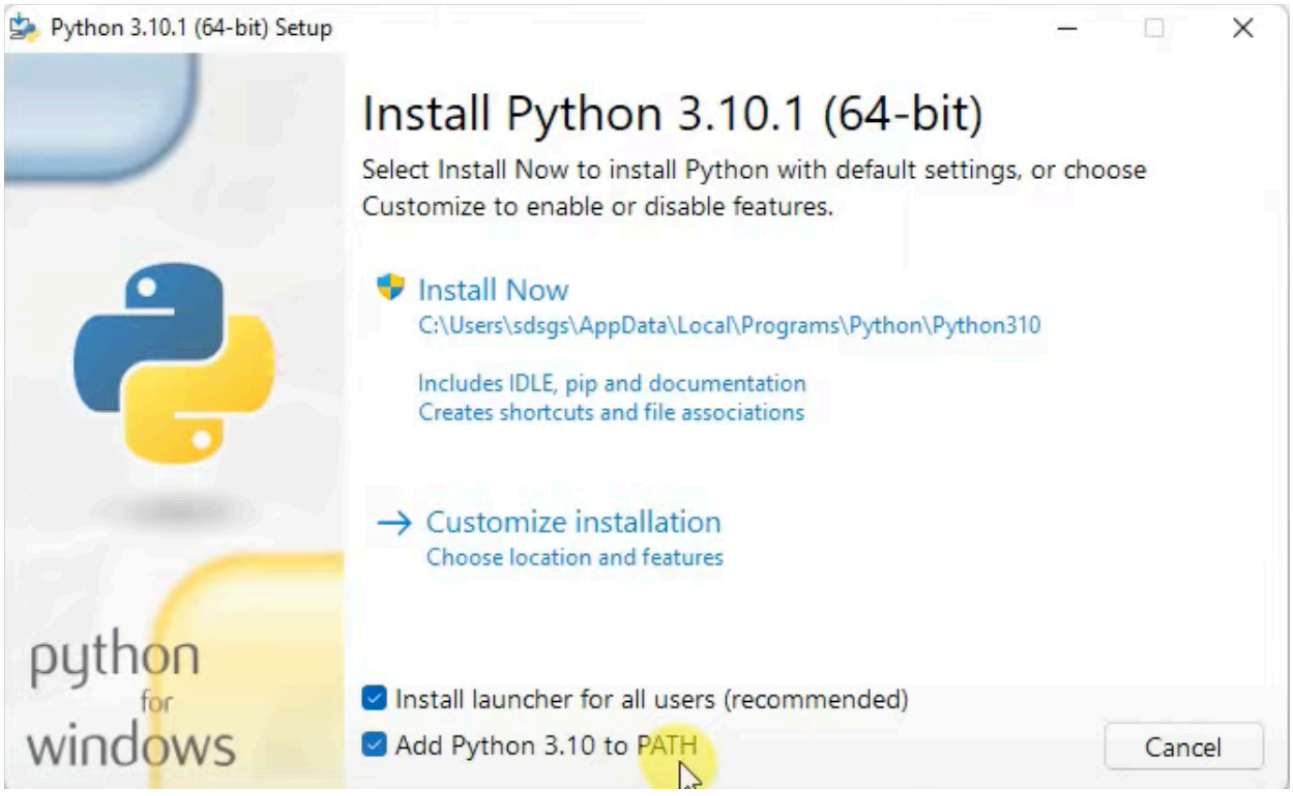
Download Python from this below link -

https://www.python.org/downloads/

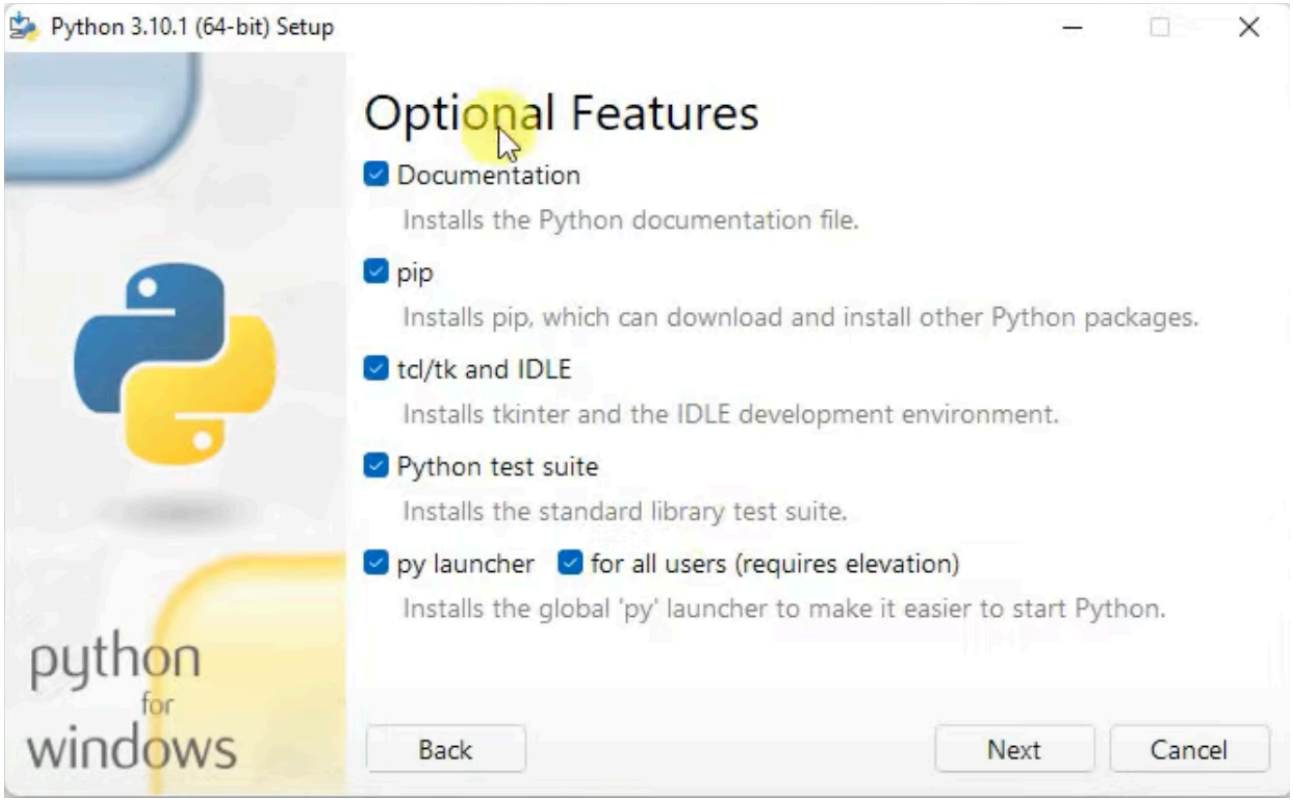
Installation on Windows

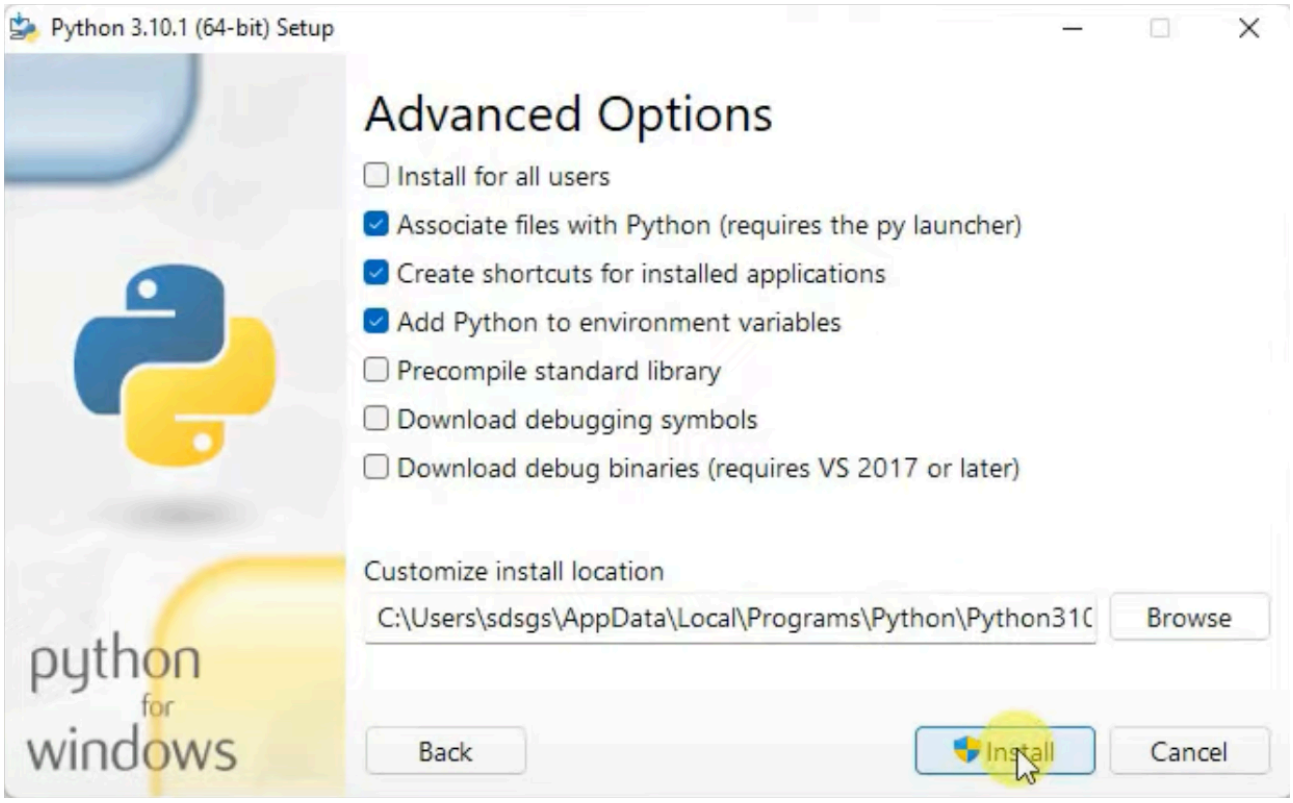
Run the python installer once downloaded

*****Very Important - Make sure to select “Add Python 3.10 to PATH” & “Install launcher for all user“ checkbox**

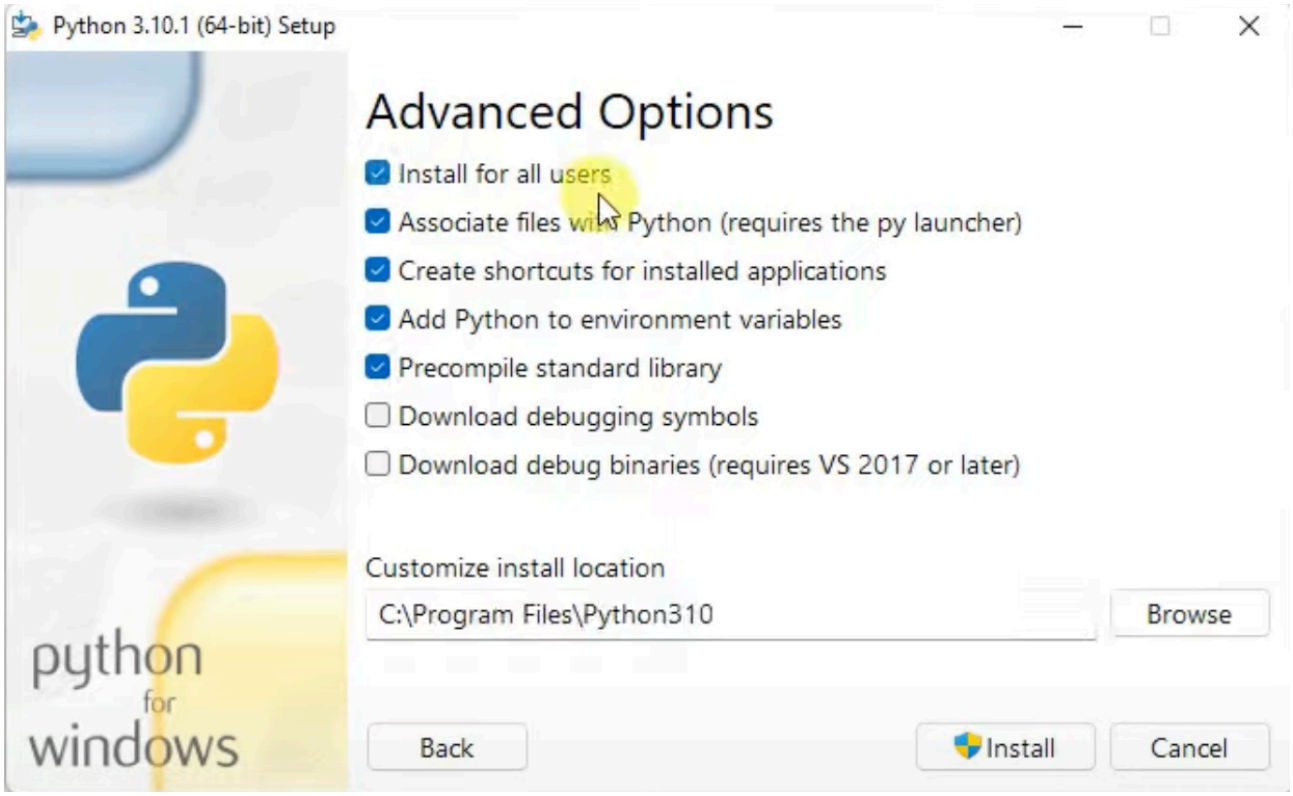


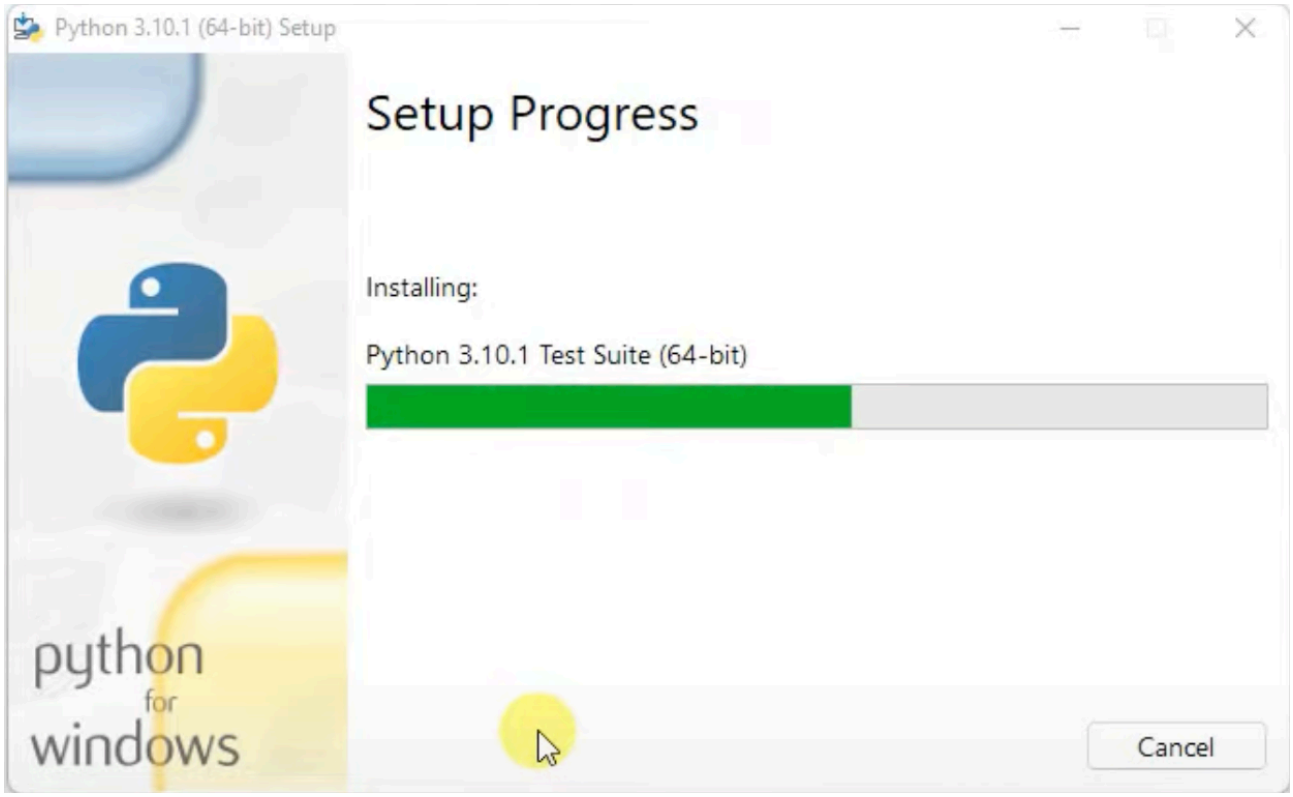
Click on “Customise Installation”

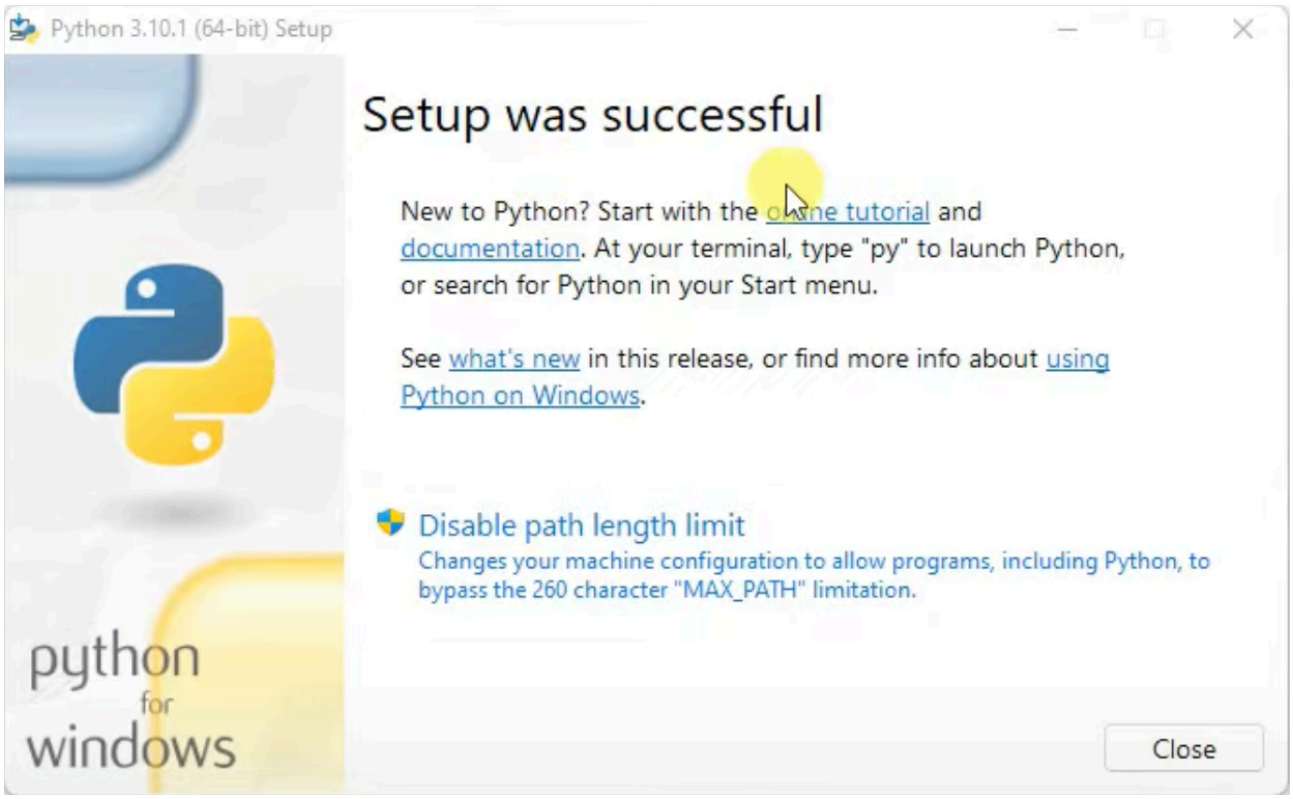




Select “Install for all users” and “Precompile standard library” checkbox
And also change the install location if you want and then click on Install

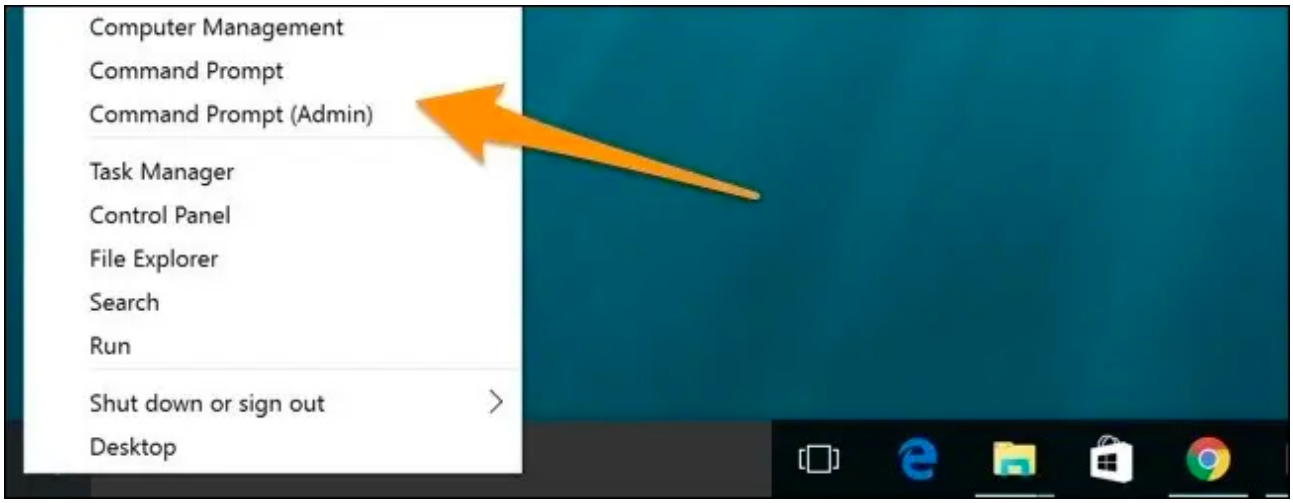




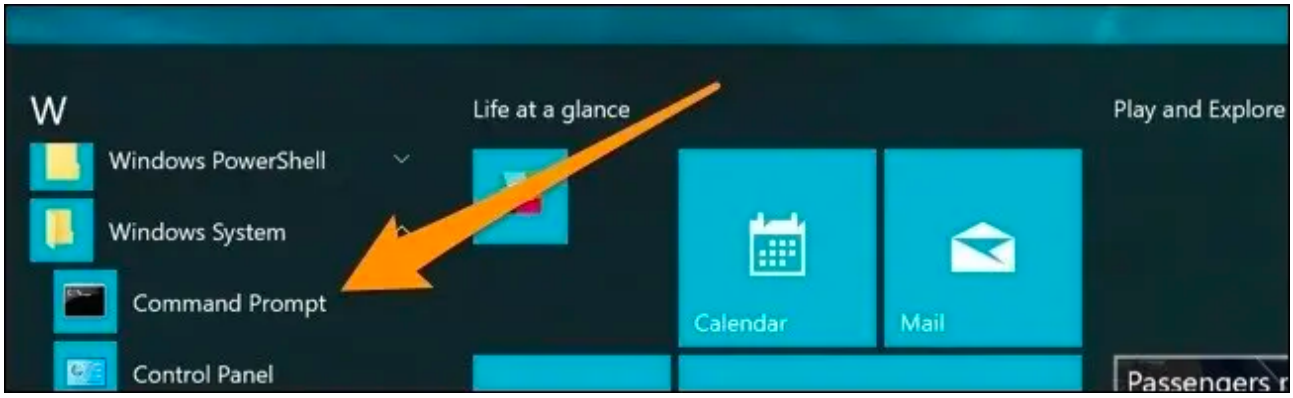


Verify if installation was done successfully, -

Press Windows+X to open power users menu and then click “Command Prompt” or “Command Prompt (Admin)”

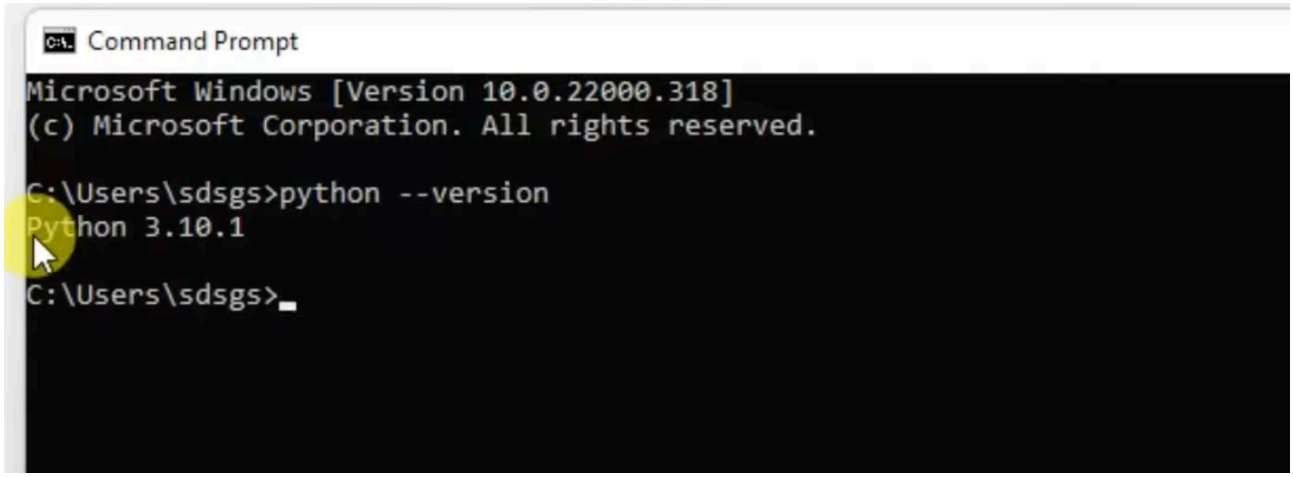


Or Click start. Scroll down and expand the “Windows System” folder, click “Command Prompt”



In Command prompt type -

python —version



INSTALL ON MAC

MAC already has python installed, but this won't be latest so to get latest python version, download python form https://www.python.org/downloads/

Installation on MAC is fairly simple, Run the python installer once downloaded and just click on continue and install once asked

Once Installed, Open "Terminal" through spotlight search, and type "python —version"

For "python —version" you will get "Python 2.x" as result as this is the default python installed on Mac. For "python3 —version" you will get "Python 3.x", which we just installed.

```
Last login: Mon Dec 20 10:57:59 on ttys002
➜  ~ python --version
Python 2.7.16
➜  ~ python3 --version
Python 3.10.0
➜  ~
```

Code Editor - Pycharm Installation

Pycharm is a code editor for python. Pycharm is an intelligent code editor which provides smart code completion, code inspection, error highlighting and quick fixes. It comes with build-in development tools which makes developers life much easier.

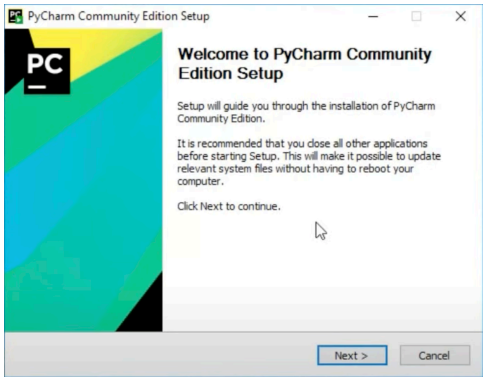
You can use any other code editor like Visual Studio code as well, Or even write python code without any code editor but Pycharm is the most popular and highly used code editor for whiting python code, so we will use PyCharm for this tutorial.

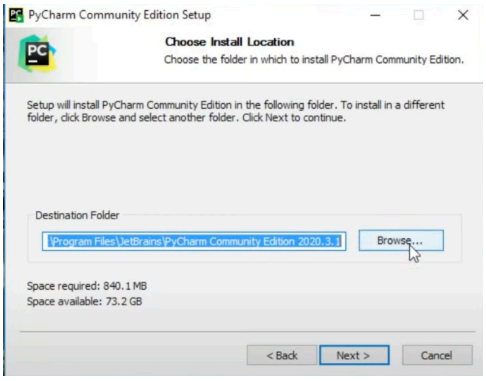
Download from this below link -
https://www.jetbrains.com/pycharm/download/#section=mac

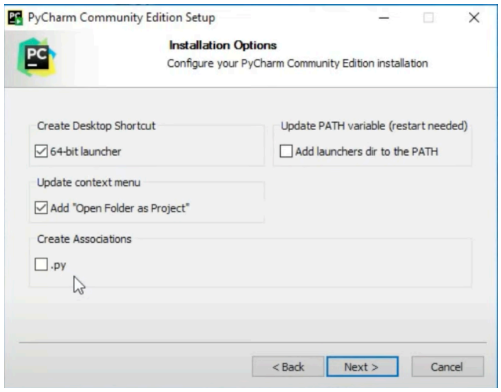
Download the free and open source community version, as that is more than enough for us.

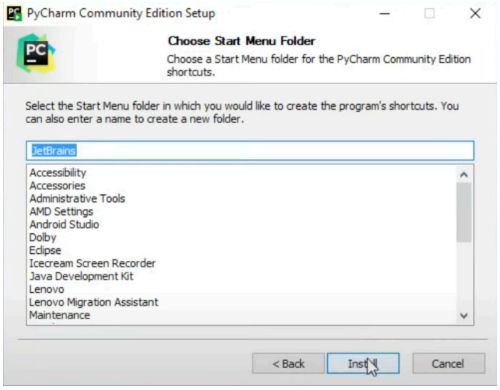
Installation of pycharm is very simple on both Windows and MAC, just click on the installer once it is downloaded and follow the screens to install. For Windows, change the Install Location if needed. And for MAC, Once installed move PyCharm to Applications.

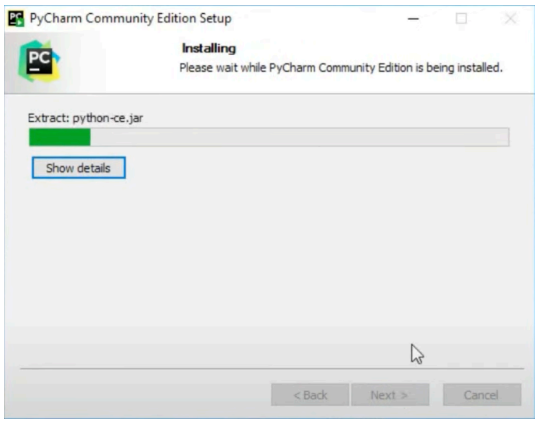
For Windows - Follow the below screenshots **-**

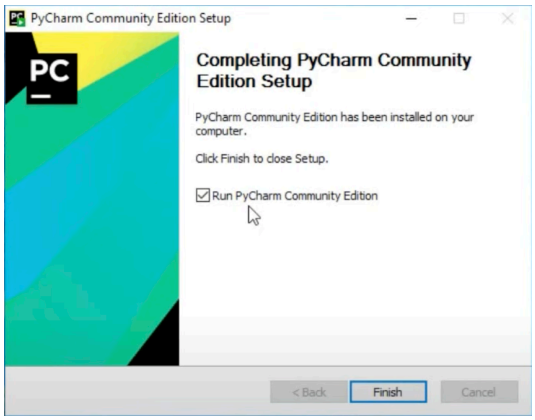




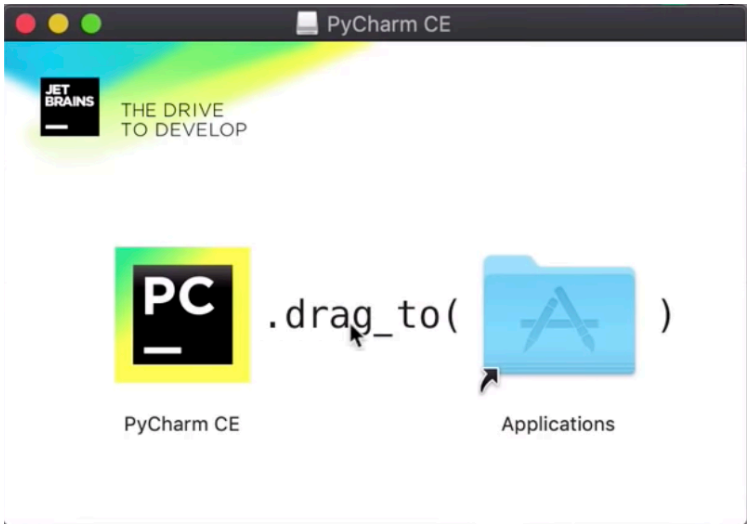






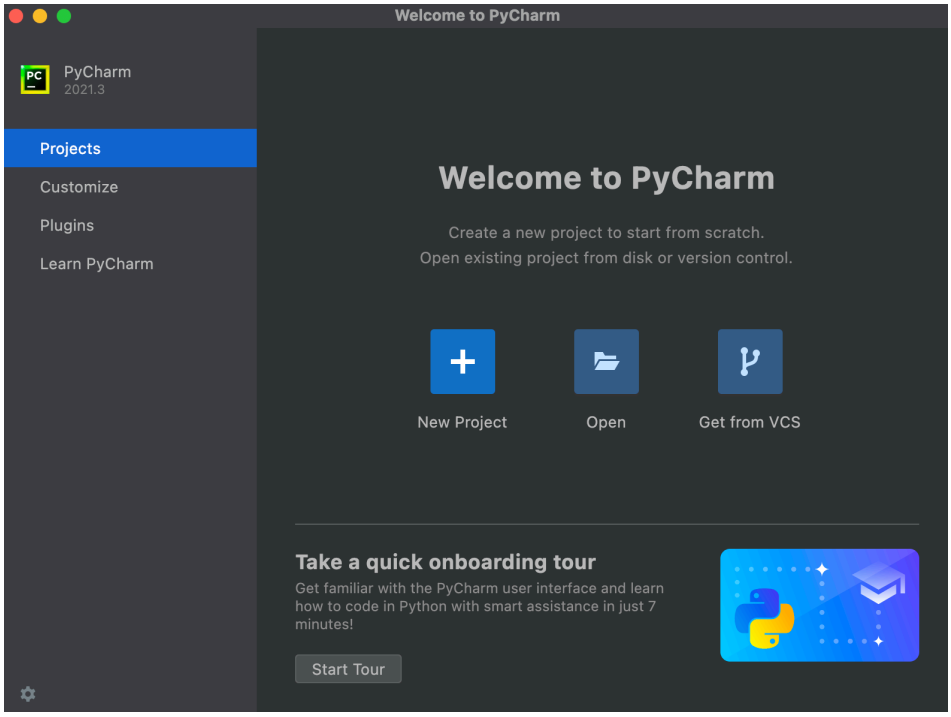


For MAC - once PyCharm .dmg is fully downloaded, just click on it and move it to Applications folder. As shown below.

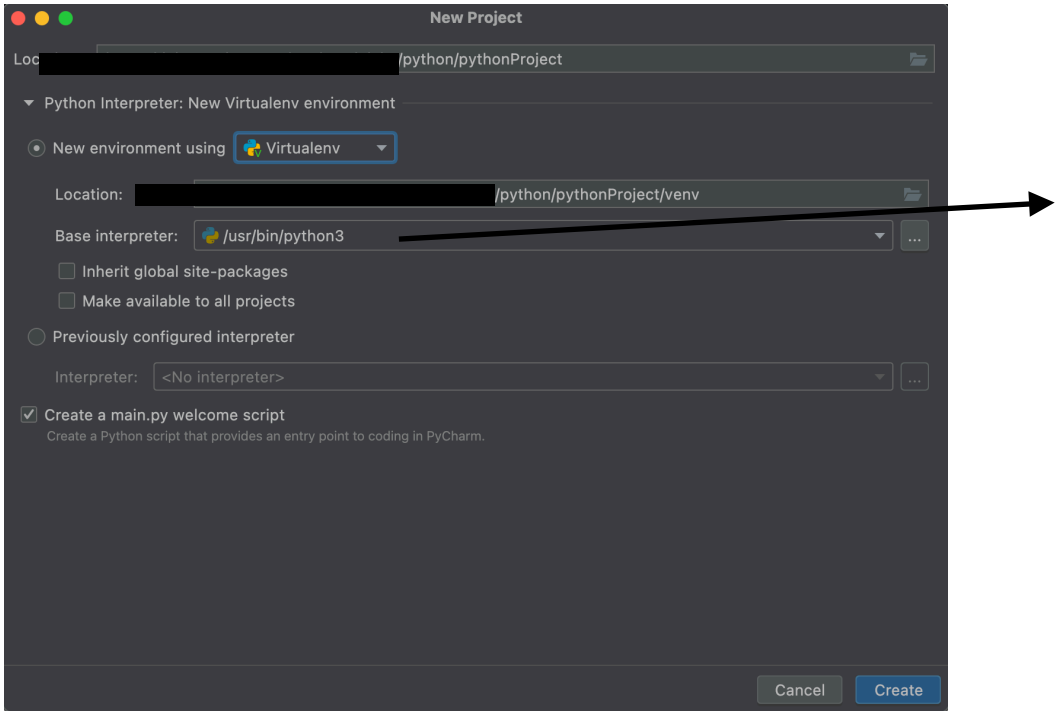


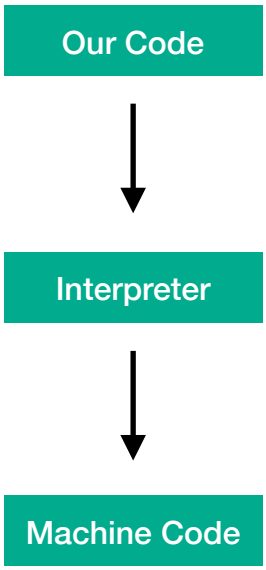
Thats it! We are all set to create our First Python Application with PyCharm.

Once Installed Open PyCharm & Click on New Project to create your first project for Python!



Enter your project name in the next screen, and change you project location to your preferred location. For MAC, make sure Python3 is selected in Base Interpreter.





NOTE* Base Interpreter - Executes our Python program by converting our python code to machine readable binary code.**

PYTHON SYNTEX

First Program -

Print Hello World

```
print("Hello World")
```

- output

Hello World

Variables - A variable is a container that holds information that you can access later.

```
age = 20

name = "Rahul"

print(age)
print(name)
```

- output

20
Rahul

```
age = 30
```

- output

30

Data Types - Type of the variables

Basic Data types -

String - Strings can hold any text like your name, address

```
name = "Rahul"
address = "34th Street, 5th Avenue"
```

Int - Int can hold any number like your age, score in game

```
age = 20
score = 100
```

Float - Float can hold decimal number like rating

rating = 4.8

bool - bool can hold true/false value like weather you are vegetarian or not

is_vegetarian = True
is_non_vegetarian = False

There are many other data type, which we will later on.

User Inputs - In python we use input() function to get user input.

```
name = input("What is your name? ")
print("Hi " + name)
```

- output

What is your name? Rahul
Hi Rahul

IF Else Condition - To take decision based on some condition we use if else condition in programming.

```
name = input("What is your name? ")

is_veg = False
is_egg = True

    if is_veg:
        print(name + " is vegetarian")
    elif is_egg:
        print(name + " is eggetarian")
    else:
        print(name + " is not vegetarian")
```

- output

What is your name? Rahul
Rahul is eggetarian

PRACTICE : Quiz Game - Lets create a quiz game with everything we have learned so far.

Problem Statement : We will ask user three question and check it with correct answer.

```
answer1 = input("Which animal is called King of Jungle? ")
if answer1 == "Lion":
    print("Correct Answer!!")
else:
    print("Game Over")
    quit()

answer2 = input("Which is the tallest animal on the earth? ")
if answer2 == "Giraffe":
    print("Correct Answer!!")
else:
    print("Game Over")
    quit()

answer3 = input("What type of bird lays the largest eggs? ")
if answer3 == "Ostrich":
    print("Correct Answer!!")
else:
    print("Game Over")
    quit()
```

While Loop : We use while loop to execute a piece of code in loop.

Syntax -
while condition:
    // Code that executes in loop until the while condition is true
else:
    // code that executes once, when while condition is false

Ex -
    i = 0
    while i < 5:
        I = I + 1
        print(f“Hello {I}”) // This is how we do string formatting in python
    else:
        print(“Number is bigger then 5”)

        - Output
            Hello 1
            Hello 2
            Hello 3
            Hello 4
            Hello 5
            Number is created then 5

PRACTICE : GUESS THE NUMBER - Make a game where user need to guess a number, and user will be given 3 attempts to guess the correct number.

```
number = 10

attempt_limit = 3

i = 0


while i < attempt_limit:

    guess_number = input("Guess the number ")

    if int(guess_number) == number:

        print("You Win!!")

    else:

        print("You Lost, try again!")

    i = i + 1

else:

    print("You Lost, Game Over")
```

List - List data type is use to store multiple items, list is changeable, ordered and allow duplicate values

```
fav_fruits = [“Apple”, “Banana”, “Orange”]

print(fav_fruits)     // Output - ['Apple', 'Banana', ‘Orange']
```

Change a item in List -

```
fav_fruits[0] = “Grapes”
print(fav_fruits)     // Output - ['Grapes', 'Banana', ‘Orange']
```

Add new item in List -

```
fav_fruits,append(“Mango”)
print(fav_fruits)     // Output - ['Grapes', 'Banana', ‘Orange’, ‘Mango’]
```

Remove new item in List -

```
fav_fruits.remove(“Orange”)
print(fav_fruits)     // Output - ['Grapes', 'Banana', 'Mango']
```

For Loop - In python we can use for loop to iterate through each item in any List or sequence.

Ex -

```
for fruit in fav_fruits:
        print("I like " + fruit)
```

- Output -

```
I like Grapes
I like Banana
I like Mango
```

```
for i in "PYTHON":
        print(i)
```

- Output -

```
P
Y
T
H
O
N
```

Set - This also stores multiple values but you can not have duplicates and sets are unordered.

fruits_set = {“apple”, “banana”, “orange”}

Add new item in Set -
fruits_set.add(“mango”)

Remove item from Set -
fruits_set.remove(“orange”)

Tuples - This also stores multiple values. Tuples are unchangeable, immutable and ordered.

fav_fruits = (“apple”, “banana”, “orange”)

fav_fruits[0]. // output - apple

Dictionary - With dictionary also we store multiple values, the benefit is we can store key and value in dictionary, and it does not allow duplicate keys.

Ex -
```
person_dict = {
                    “Name” : “Rahul” ,
                    “Age” : 30,
                    “Occupation” : “Software Engineer”
              }
```

```
print(person_dict[“Name”])          // output - Rahul
print(person_dict[“Age”])           // output - 30
print(person_dict[“Occupation”])    // output - “Software Engineer”
```

Change value for a key -
```
person_dict[“Occupation”]  =  “Senior Software Engineer”
print(person_dict[“Occupation”])    // output - “Senior Software Engineer”
```

Add new key value -
```
person_dict[“Address”]  =  “34th Street”
print(person_dict)
```

output -
{'name': 'Rahul', 'age': 30, 'Occupation': 'Software Engineer', 'address': '34th Street'}

Delete key value -
```
person_dict.pop(“Address”)
```

output -
{'name': 'Rahul', 'age': 30, 'Occupation': 'Software Engineer'}

Loop through each key value in dictionary -

```
for key, value in person_dict.items():
        print(f"{x} {y}")
```

output -

```
name Rahul
age 30
Occupation Senior Software Engineer
```

PRACTICE : Refactor Quiz Game With Dictionary -

```
question_dict = {
    "Which animal is called King of Jungle? ": "Lion",
    "Which is the tallest animal on the earth? ": "Giraffe",
    "What type of bird lays the largest eggs? ": "Ostrich"
}

for question, answer in question_dict.items():
    user_answer = input(question)
    if user_answer == answer:
        print("Correct Answer!!")
    else:
        print("Game Over")
        break
```

Functions - As the name suggest, function is task. Like eating, running and jumping in real world is a task.

To perform any task in programming we use function. A function is block of code which is written to perform specific task.

We can pass data, known as parameter into a function.

A function can return data as a result.

Ex -

```
def function_name():
    print("Hello from a function")

function_name()
```

- Output -

Hello from a function

Function with return values and parameter -

```
def square_root(x):
    return x*x

print(square_root(3))
```

- Output -

9

PRACTICE - Refactor Quiz Game with Functions -
Create quiz game for new set of questions as well, for example - quiz with capitals related questions. And make sure to write reusable code.

```
def quiz_game(question_dict):
    for question, answer in question_dict.items():
        user_answer = input(question)
        if user_answer == answer:
            print("Correct Answer!!")
        else:
            print("Game Over")
            break
animal_question_dict = {
    "Which animal is called King of Jungle? ": "Lion",
    "Which is the tallest animal on the earth? ": "Giraffe",
    "What type of bird lays the largest eggs? ": "Ostrich"
}
quiz_game(animal_question_dict)
capitals_question_dict = {
    "What is capital of India? ": "Delhi",
    "What is capital of USA ? ": "Washington DC",
    "What is capital of UK? ": "London"
}
quiz_game(capitals_question_dict)
```

Modules & Packages -

Module - A python module is simply a python file with .py extension. The purpose of module is to make python modularised and reusable. We have many built in modules which python gives us to help us write code easily. Similarly we can create our own modules.

Package - A package is group of module and a special file __init__.py. Package can also have sub packages as well. Purpose of package is to group similar modules with same agenda and distribute it to developers for reusability.

Ex - Django - Third Party Package to make web application

How can we install third party packages? We can check all the third party package from https://pypi.org/ and install them using the instruction given. Like to install Django package we can use below command -

pip install django

Excel Automation - For excel automation we need to use a third party package "openpyxl". So first, we need to install this package using - pip install openpyxl

And then we can write below code to automate excel -

```
import openpyxl as xl


wb = xl.load_workbook('employee.xlsx')

sheet = wb['Sheet1']


for row in range(2, sheet.max_row + 1):
    cell = sheet.cell(row, 3)
    if sheet.cell(row, 2).value == "Senior Software Engineer":
        increased_salary = cell.value + (cell.value * 0.2)
    else:
        increased_salary = cell.value + (cell.value * 0.1)
    increased_salary_cell = sheet.cell(row, 4)
    increased_salary_cell.value = increased_salary


wb.save('employee2.xlsx')
```

Object Oriented Programming

Object Oriented Programming, also knows as OOPS, is a programming paradigm which many programming languages follow. In this paradigm, everything in software design revolves around class and objects.

Class - Class is a blueprint of an actual or real world scenario.

Object - Object is an implementation/ instance of a class

Python Class & Objet Syntax -

```
def ClassName:

    def __init__(self, parameter_name):
        self.variable_name = parameter_name

    def any_function_name(self):
        print("function definition")
```

Object -

```
object_name = ClassName(parameter_value)
object_name.any_function_name()
```

Example

```
def User:

    def __init__(self, username_name):
        self.username_name = username_name

    def change_user_name(self, new_user_name):
        self.username_name = new_user_name
        print(f"Username changed to {self.username_name}")
```

Object -

```
user = User("Rahul")
user.change_user_name("Rohit")
```

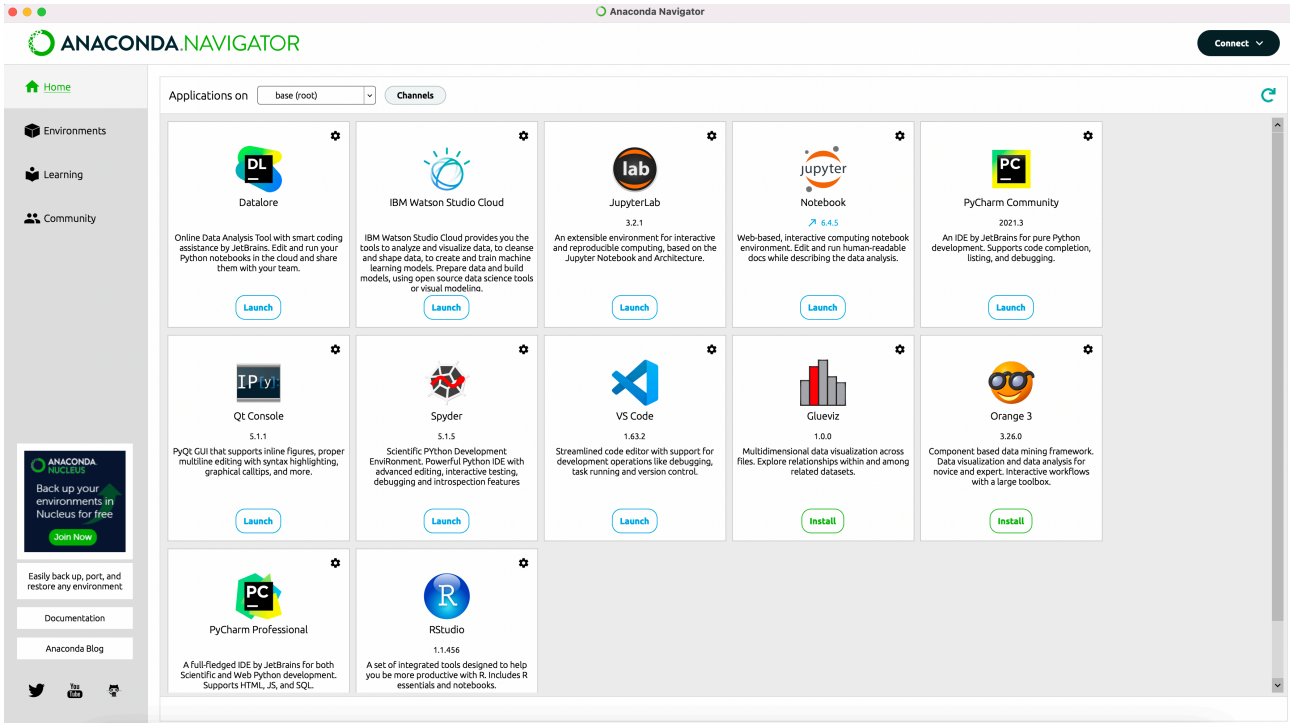
- **Output -**
  Username changed to Rohit

Data Analytics & Visualisation with Python

For data analytics and visualisation with python, we use a web application tool called Jupiter Notebook. As visualisation can not be done very well with pycharm. Jupiter is python based web application through which you can easily create data analysis, machine learning and data science project.

To install Jupiter, you need to install Anaconda - https://www.anaconda.com/products/individual
And just follow the installer window to install Anaconda in your system.

Once installed you should be able to see Anaconda Navigator in your launchpad or start menu. Click on that and you should see this below window -



Anaconda, installs everything that you may require for data analysis, machine learning and also for data science. We will only work with Jupiter Notebook. So click on the launch button to launch the Jupiter notebook or you can open command prompt/ terminal and run the command - “jupiter notebook”, and this should open Jupiter notebook in your preferred browser.

Now in browser, select the folder where you want to create your project. and then click on “New”, to create new project, and select python3 notebook .

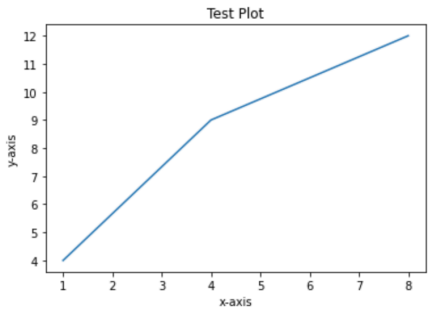
Data analysis and Visualisation project -

Simple chart -

In [3]:
```
import pandas as pd
```

In [4]:
```
from matplotlib import pyplot as plt
```

In [7]:
```
x = [1, 4, 8]
y = [4, 9, 12]
plt.plot(x , y)
plt.title('Test Plot')
plt.xlabel('x-axis')
plt.ylabel('y-axis')
plt.show()
```

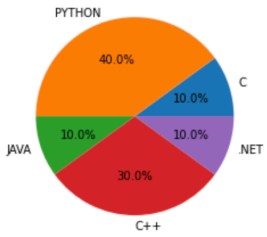


Simple Pie Chart -

In [8]:
```
lang = ['C', 'PYTHON', 'JAVA', 'C++', '.NET']
```

In [9]:
```
student_count = [10, 40, 10, 30, 10]
```

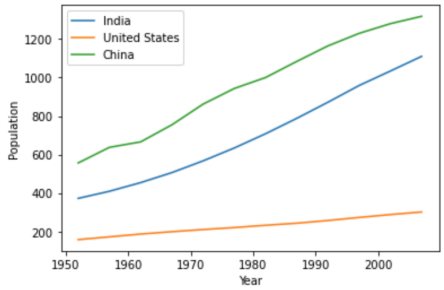
In [12]:
```
plt.pie(student_count, labels = lang, autopct = '%0.1f%%')
plt.show()
```



Compare India, United States and China Population - Download the CSV file and keep it in the same folder as this project and then write below code to get the comparison chart -

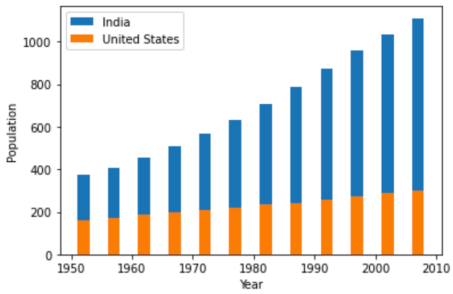
```
In [13]: data = pd.read_csv('countries.csv')

In [15]: india = data[data.country == 'India']

In [17]: united_state = data[data.country == 'United States']

In [30]: china = data[data.country == 'China']
```

```
In [27]: plt.plot(india.year, india.population / 10**6)
         plt.plot(united_state.year, united_state.population / 10**6)
         plt.plot(china.year, china.population / 10**6)
         plt.legend(['India', 'United States', 'China'])
         plt.xlabel('Year')
         plt.ylabel('Population')
         plt.savefig('data_analysis_on_population_growth.png', dpi = 300)
         plt.show()
```



Bar Chart -

```
In [29]: plt.bar(india.year, india.population / 10**6, width = 2)
         plt.bar(united_state.year, united_state.population / 10**6, width = 2)
         plt.legend(['India', 'United States'])
         plt.xlabel('Year')
         plt.ylabel('Population')
         plt.savefig('data_analysis_on_population_growth.png', dpi = 300)
         plt.show()
```




```
In [ ]:
```