Python Fundamentals

Introduction to Computer Programming

What is Computer Programming?

- **A process that professionals use to write code that instructs how a computer, application or software program performs.**

- **Computer program is a set of instructions to facilitate specific actions.**

Common tasks a computer programmer

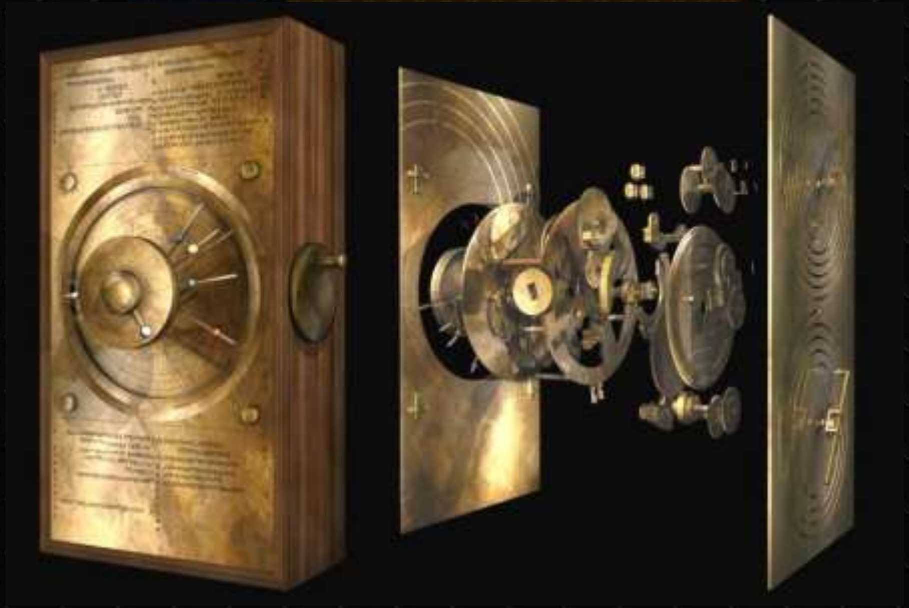
- Testing software performance.
- Resolving computer software problems.
- Modifying software programs to improve performance.
- Writing computer programming code.
- Collaborating with others to resolve information technology issues.

Python Fundamentals

History of Computer Programming

Antikythera mechanism (200 BC – 70 BC)

- Clockwork devices are probably the first know first “programming”.
- The Antikythera mechanism is an ancient Greek hand-powered orrery, described as the oldest example of an analogue computer used to predict astronomical positions and eclipses decades in advance.



“The Writer” 1770s

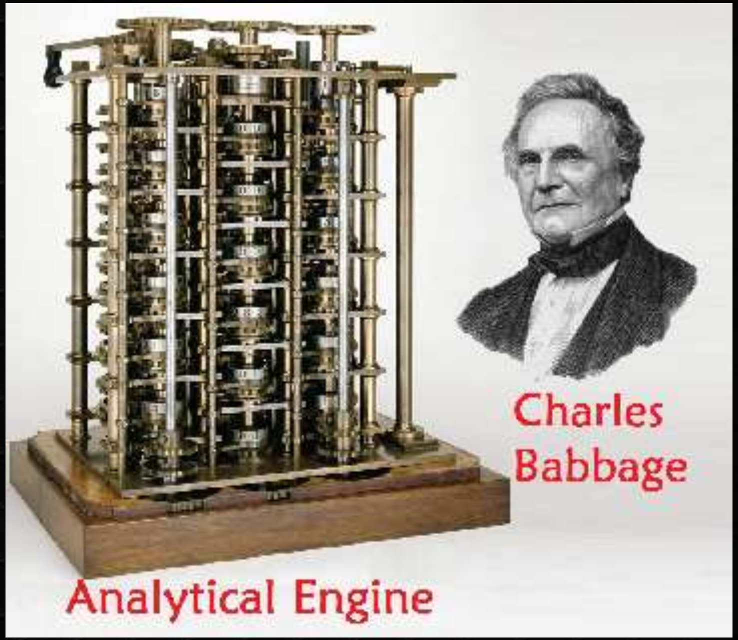
- Many clockwork devices were so intricate that they could be “programmed” to complete a series of complex tasks, such as dancing or writing.
- The Writer”, an automaton (mechanical doll), designed and built in the 1770s by Pierre Jaquet-Droz, a Swiss watchmaker, is one particularly spectacular example.

The First Binary System - 1804

- The Jacquard Loom punch card system, in 1804, is probably the first known example of the binary system (at least an on/off instruction format)
- A series of punched cards were fed into the loom. If there was a hole in the card, the needle rose, if there was no hole, the needle stayed down. The shuttle then travelled across the loom creating a pattern in the fabric. Punch cards were later used to store other types of data

The first computer programmer - 1840s

- In the 1840s, Ada Lovelace became the first computer programmer.
- She was working on the Analytical Engine (the computer that she designed the programs for) constructed by Charles Babbage

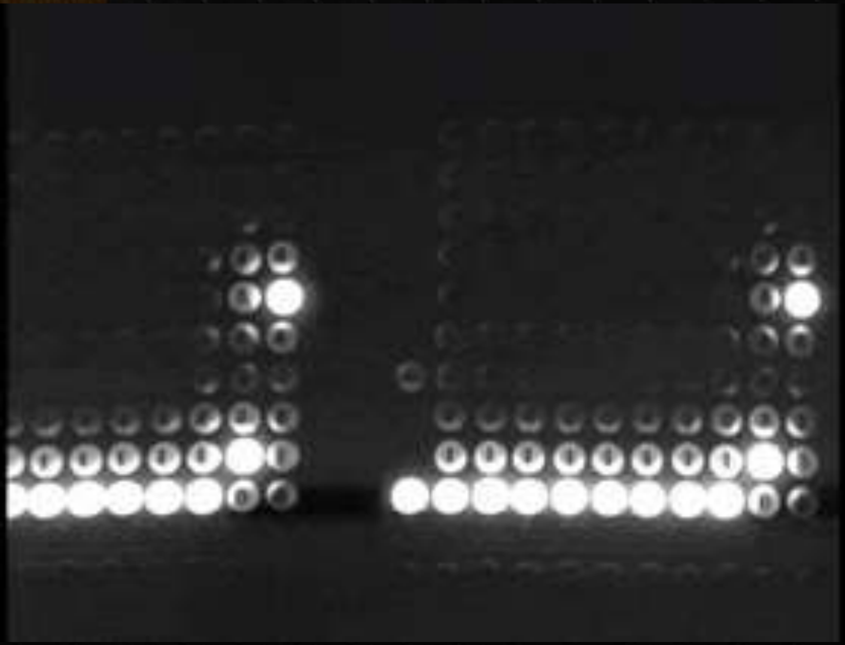


The First Computer programmer (on an electrical computer)

- 1941, Konrad Zuse became, what was probably, the first person to program an electrical computer and, unlike Lovelace, the computer was actually able to perform the operation!

The First Computer programmer (on an electrical computer)

- In 1945 the first full-time, paid computer programmers, charged with the ENIAC (the first electronic general-purpose digital computer) were Kay McNulty, Betty Jennings, Betty Snyder, Marlyn Wescoff, Fran Bilas and Ruth Lichterman
- They were dismissively labeled “refrigerator ladies”.



The First Computer Programming Language

- n 1952 – American computer scientist, Grace Hopper, developed a system that could convert plain English into computer code.
- This would later become COBOL, a computer language still widely used today for data processing!

The mother of computer programming: Grace Hopper at the UNIVAC keyboard, c. 1960

Recently Trending Programming Languages

TOP 10 Programming languages for 2022
JS
GO
Java
C++
C#
R
Kotlin
Swift

Programming Basics

Algorithm and Flowchart

Algorithm and Flowchart

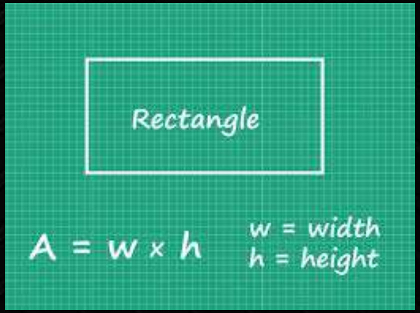
- Algorithms and flowcharts are two different tools that are helpful for creating new programs.
- An algorithm is a step-by-step analysis of the process.
- A flowchart explains the steps of a program in a graphical way.

Algorithm

- The word "algorithm" relates to the name of the mathematician Al-khowarizmi, which means a procedure or a technique.
- Software Engineer commonly uses an algorithm for planning and solving the problems.
- An algorithm is a sequence of steps to solve a particular problem

- Algorithm has the following characteristics :
  - Input: An algorithm may or may not require input
  - Output: Each algorithm is expected to produce at least one result
  - Definiteness: Each instruction must be clear and unambiguous.
  - Finiteness: If the instructions of an algorithm are executed, the algorithm should terminate after finite number of steps

HOW TO WRITE ALGORITHMS



- Step 1 Define the algorithm's input
    - Eg: To calculate the area of rectangle input may be the rectangle height and rectangle width.
- Step 2 Define the Variables
    - Eg: The two variables for rectangle height and rectangle width as HEIGHT and WIDTH
- Step 3 Define the Operations
    - Eg: Multiply the HEIGHT and WIDTH variable and store the value in new variable AREA
- Step 4 Declare the Outputs
    - Eg: Area of rectangle output will be the value stored in variable AREA

SAMPLE ALGORITHM

Step-1 Start

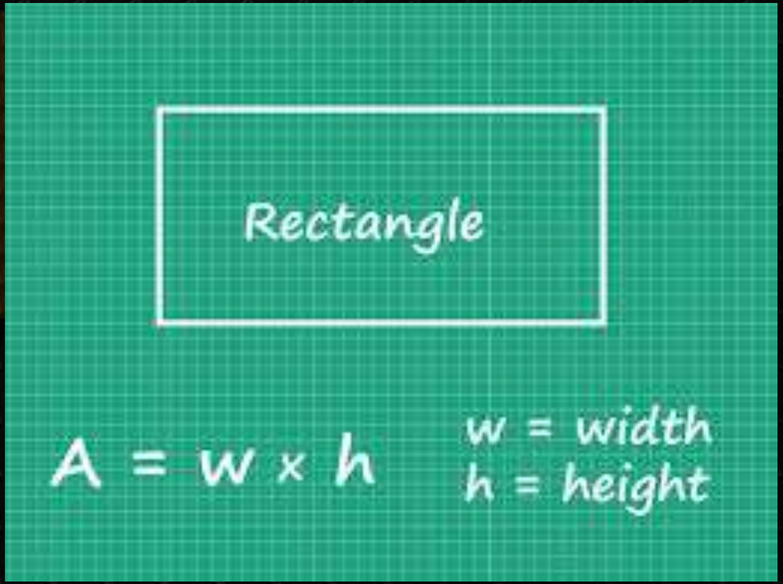
Step-2 Input Width say WIDTH

Step-3 Input Height say HEIGHT

Step-4 AREA = WIDTH * HEIGHT

Step-5 Display AREA

Step-6 Stop

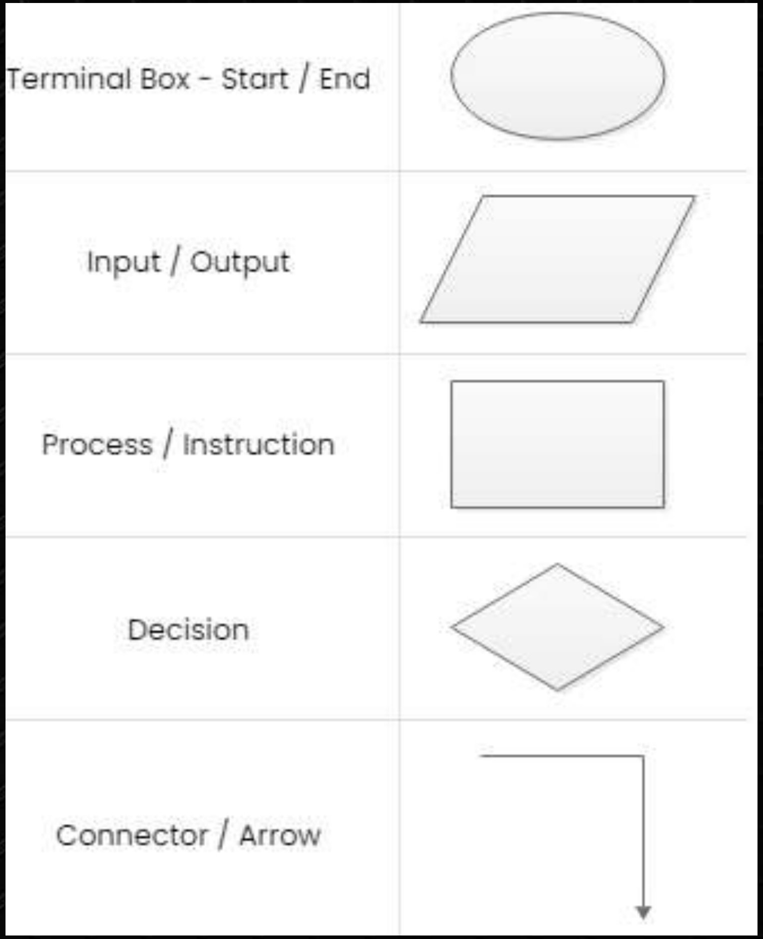


Flowchart

- A flowchart is the graphical or pictorial representation of an algorithm with the help of different symbols, shapes, and arrows to demonstrate a process or a program.

- The main purpose is to analyze different methods used in the program.

- Several standard symbols are applied in a flowchart:



SAMPLE FLOW CHART FOR RECTANGLE AREA

Step-1 Start

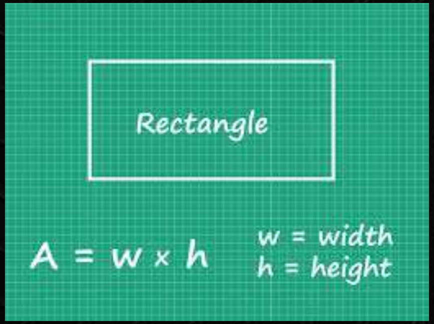
Step-2 Input Width say WIDTH

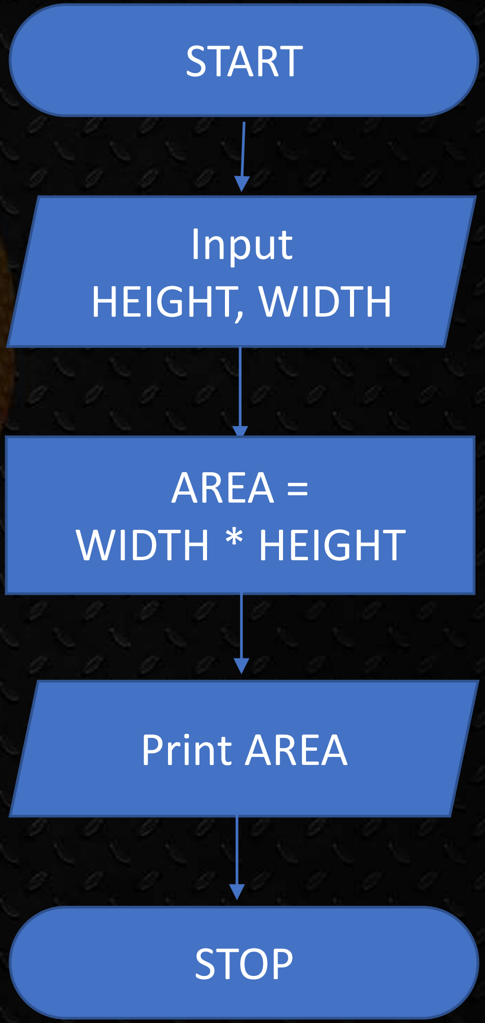
Step-3 Input Height say HEIGHT

Step-4 AREA = WIDTH * HEIGHT

Step-5 Display AREA

Step-6 Stop





Exercises:

1) Algorithm & Flowchart to find the sum of two numbers

    $C = A + B$

2) Algorithm & Flowchart to convert temperature from Celsius to Fahrenheit

    $C = 5.0/9.0\ (F - 32\ )$

3) Algorithm & Flowchart to find the smallest of two numbers

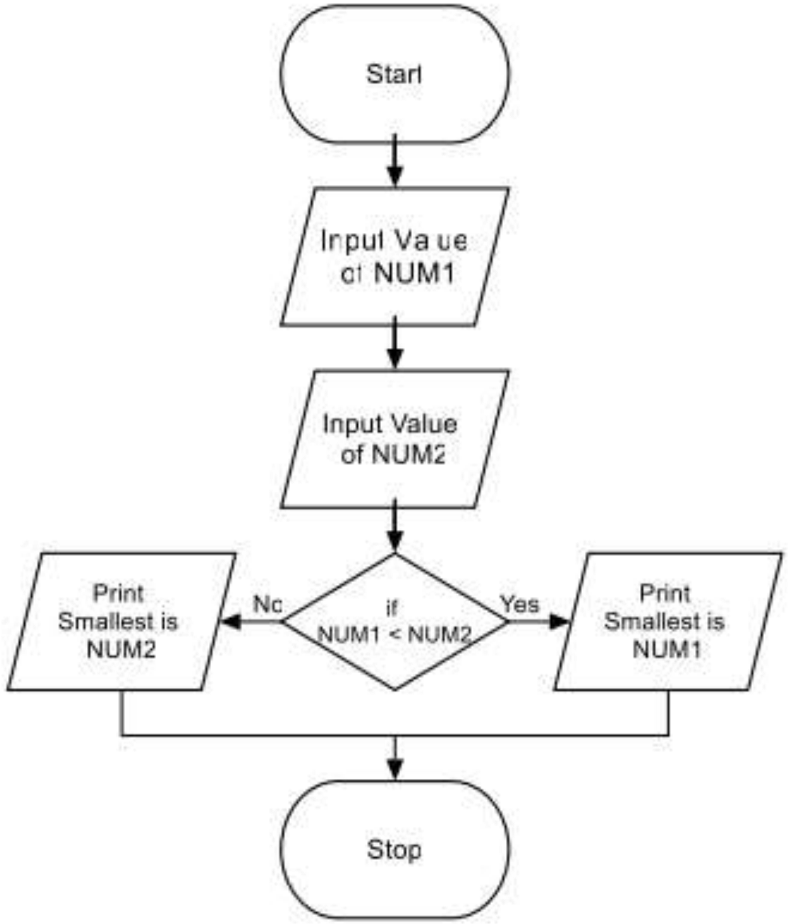
```
IF NUM1 < NUM2 THEN print smallest is NUM1
ELSE PRINT NUM2
```

Exercises:

Algorithm & Flowchart to find the smallest of two numbers

Algorithm

```
Step-1   Start
Step-2   Input two numbers say
NUM1,NUM2
Step-3   IF NUM1 < NUM2  THEN
              print smallest is NUM1
         ELSE
              print smallest is NUM2
         ENDIF
Step-4   Stop
```



Python Fundamentals

Introduction to Python

Why Python ?

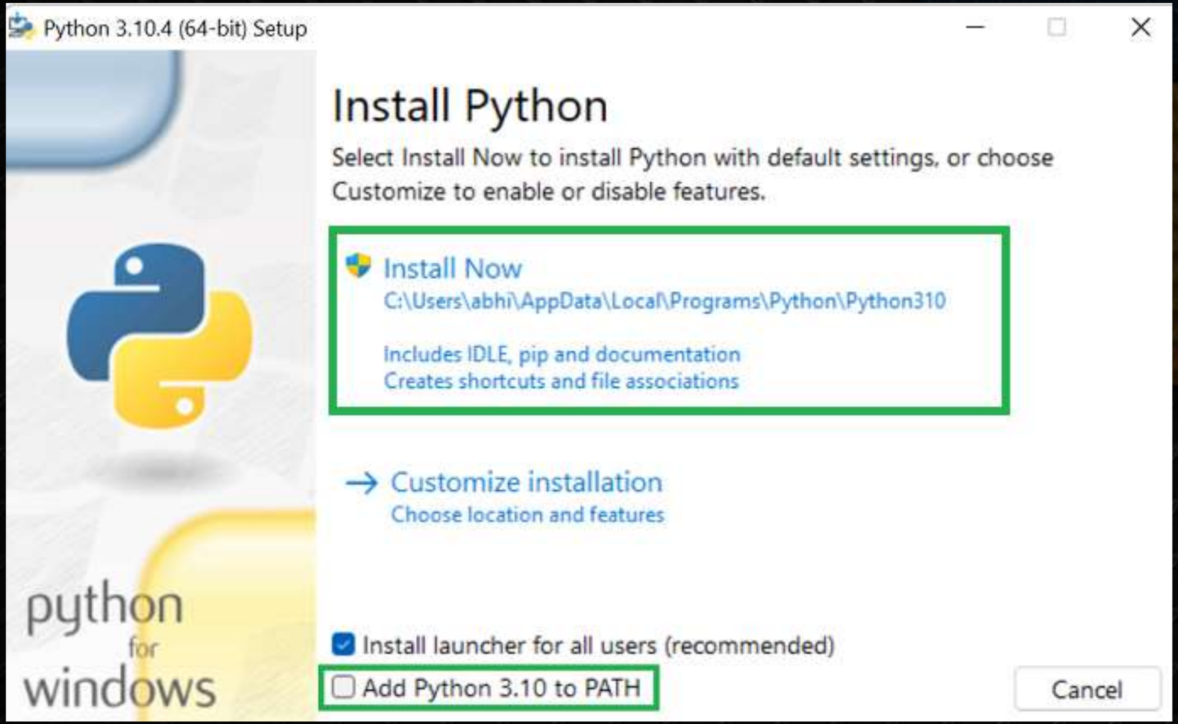
- Python is Easy to Use, Simple and Fast to Develop
- So it is the most accessible for programmers and non-programmers like scientists, researchers etc
- It is Open Source with a very Active Community.
- Vast Number of Libraries virtually for any Area of Science. Weather its ML or Physics or Biology…. etc

Python Fundamentals

Setting Up the Environment – VS CODE

Step 1: Installing Python Interpreter

- **Install Python from python.org**
- **Click the Download Python to download the latest version.**
- **Make sure to add the python to system path**


Python 3.10.4 (64-bit) Setup

Install Python

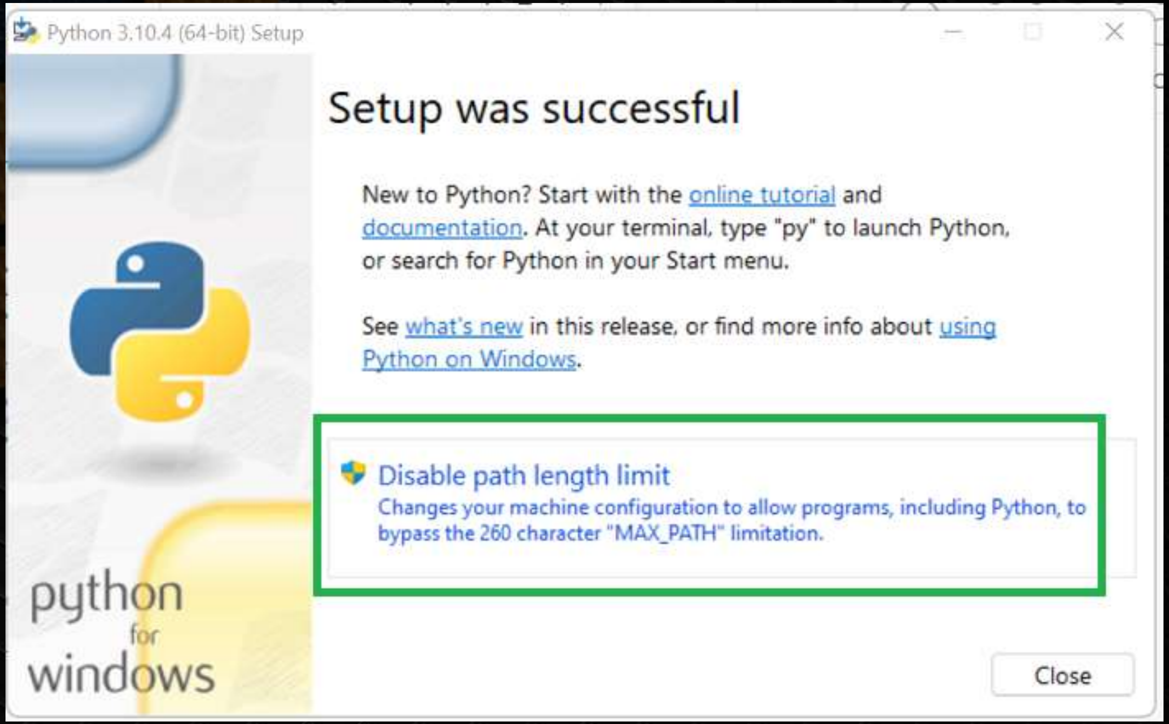
Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\abhi\AppData\Local\Programs\Python\Python310

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

[✓] Install launcher for all users (recommended)
[ ] Add Python 3.10 to PATH


Python 3.10.4 (64-bit) Setup

Setup was successful

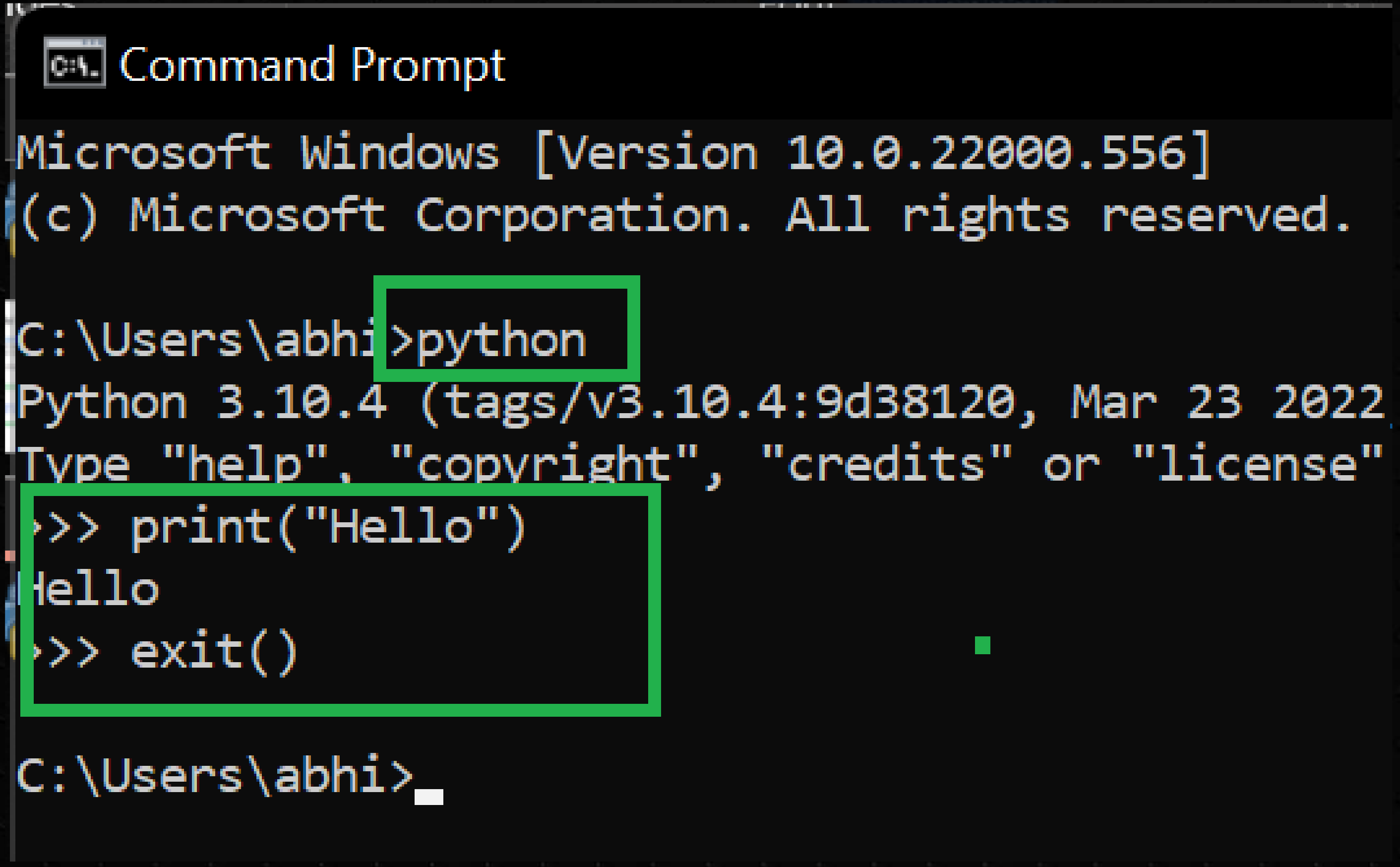
New to Python? Start with the online tutorial and documentation. At your terminal, type "py" to launch Python, or search for Python in your Start menu.

See what's new in this release, or find more info about using Python on Windows.

Disable path length limit
Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX_PATH" limitation.

Step 1: Test Installation

- **Open Command prompt and type 'python'**



Command Prompt

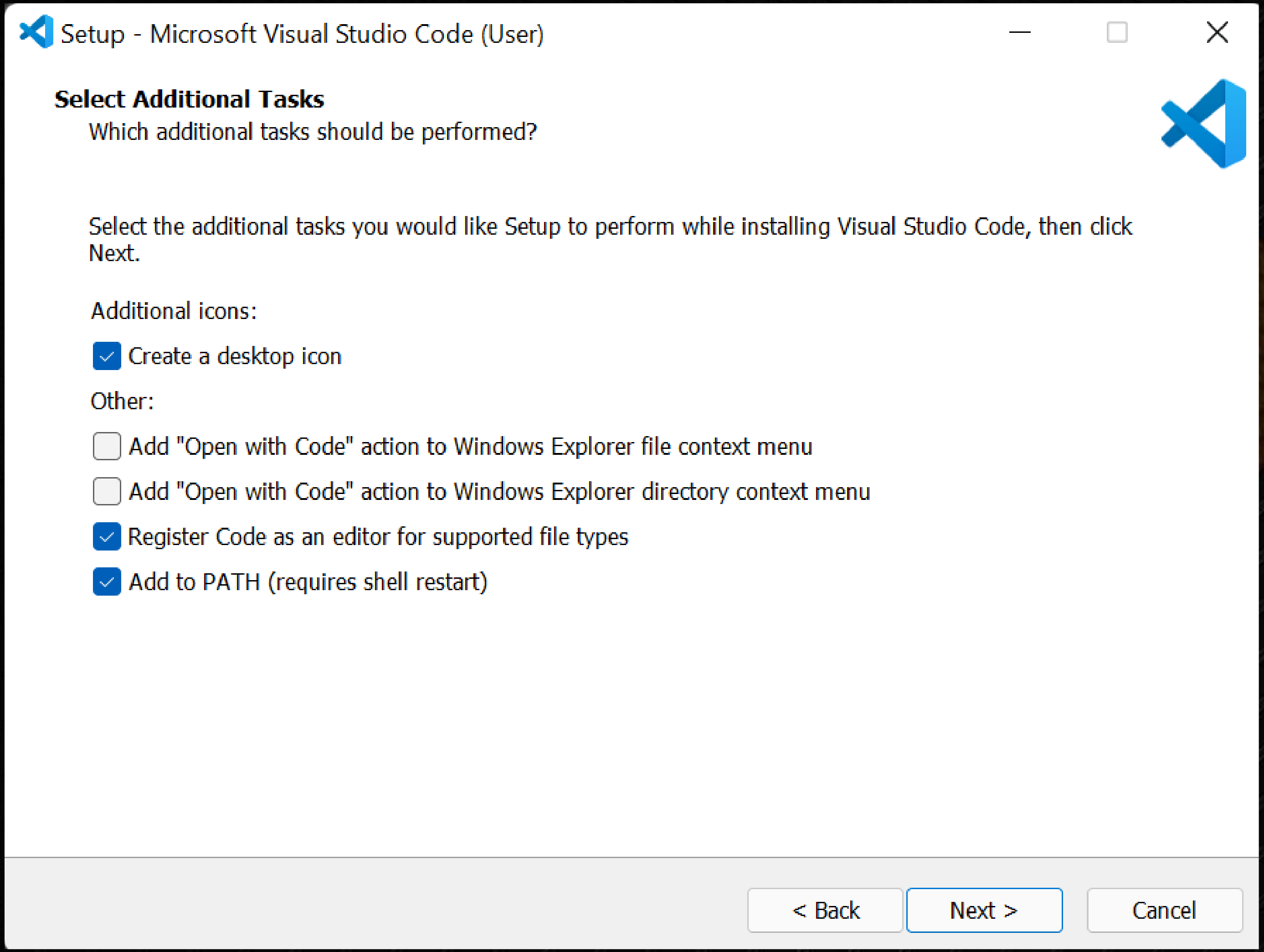
```
Microsoft Windows [Version 10.0.22000.556]
(c) Microsoft Corporation. All rights reserved.

C:\Users\abhi>python
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022
Type "help", "copyright", "credits" or "license"
>>> print("Hello")
Hello
>>> exit()

C:\Users\abhi>_
```

Step 2: Installing Visual Studio Code

Visual Studio Code

- Visual Studio Code is a lightweight but powerful source code editor from Microsoft.
- Download for free from https://code.visualstudio.com/
- It will run on your desktop and is available for Windows, macOS and Linux.
- Built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).

Step 2: Installing Visual Studio Code



Setup - Microsoft Visual Studio Code (User)

Select Additional Tasks

Which additional tasks should be performed?

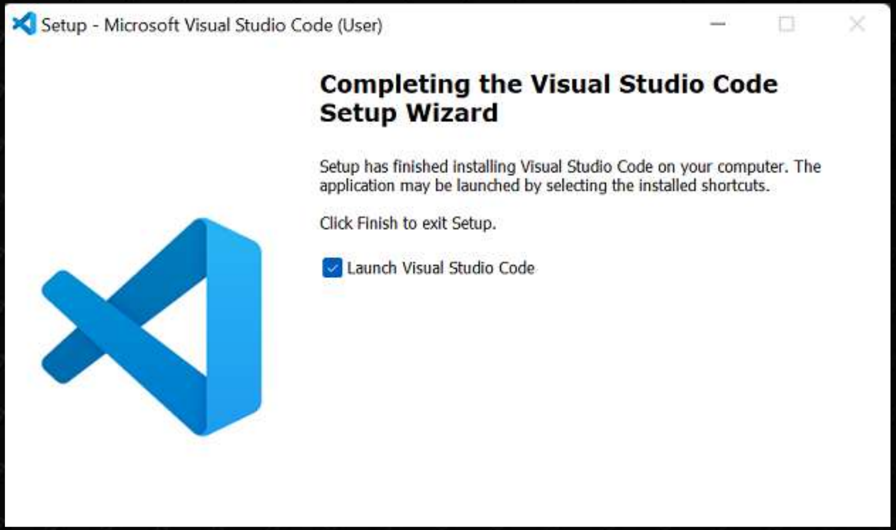
Select the additional tasks you would like Setup to perform while installing Visual Studio Code, then click Next.

Additional icons:

- [✓] Create a desktop icon

Other:

- [ ] Add "Open with Code" action to Windows Explorer file context menu
- [ ] Add "Open with Code" action to Windows Explorer directory context menu
- [✓] Register Code as an editor for supported file types
- [✓] Add to PATH (requires shell restart)



Setup - Microsoft Visual Studio Code (User)

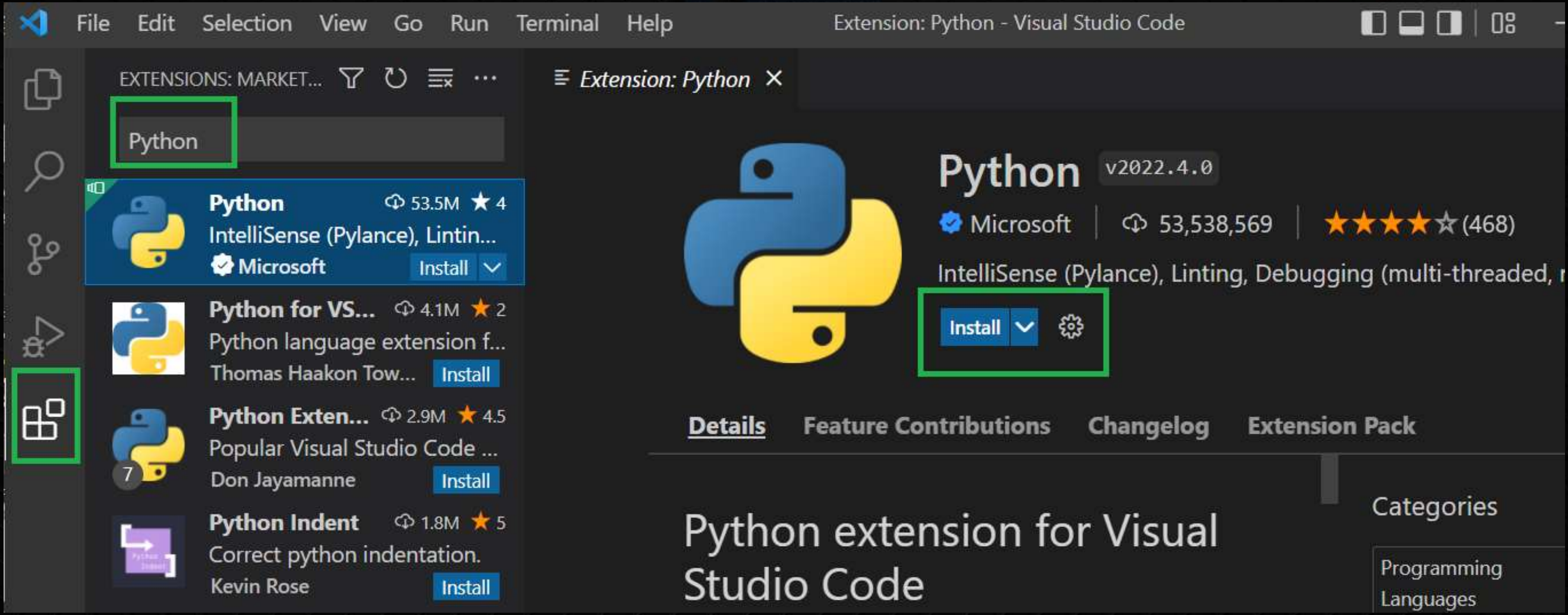
Completing the Visual Studio Code Setup Wizard

Setup has finished installing Visual Studio Code on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

- [✓] Launch Visual Studio Code

Step 3: Installing Python Extension



Python 53.5M ★ 4
IntelliSense (Pylance), Lintin...
Microsoft Install

Python for VS... 4.1M ★ 2
Python language extension f...
Thomas Haakon Tow... Install

Python Exten... 2.9M ★ 4.5
Popular Visual Studio Code ...
Don Jayamanne Install

Python Indent 1.8M ★ 5
Correct python indentation.
Kevin Rose Install

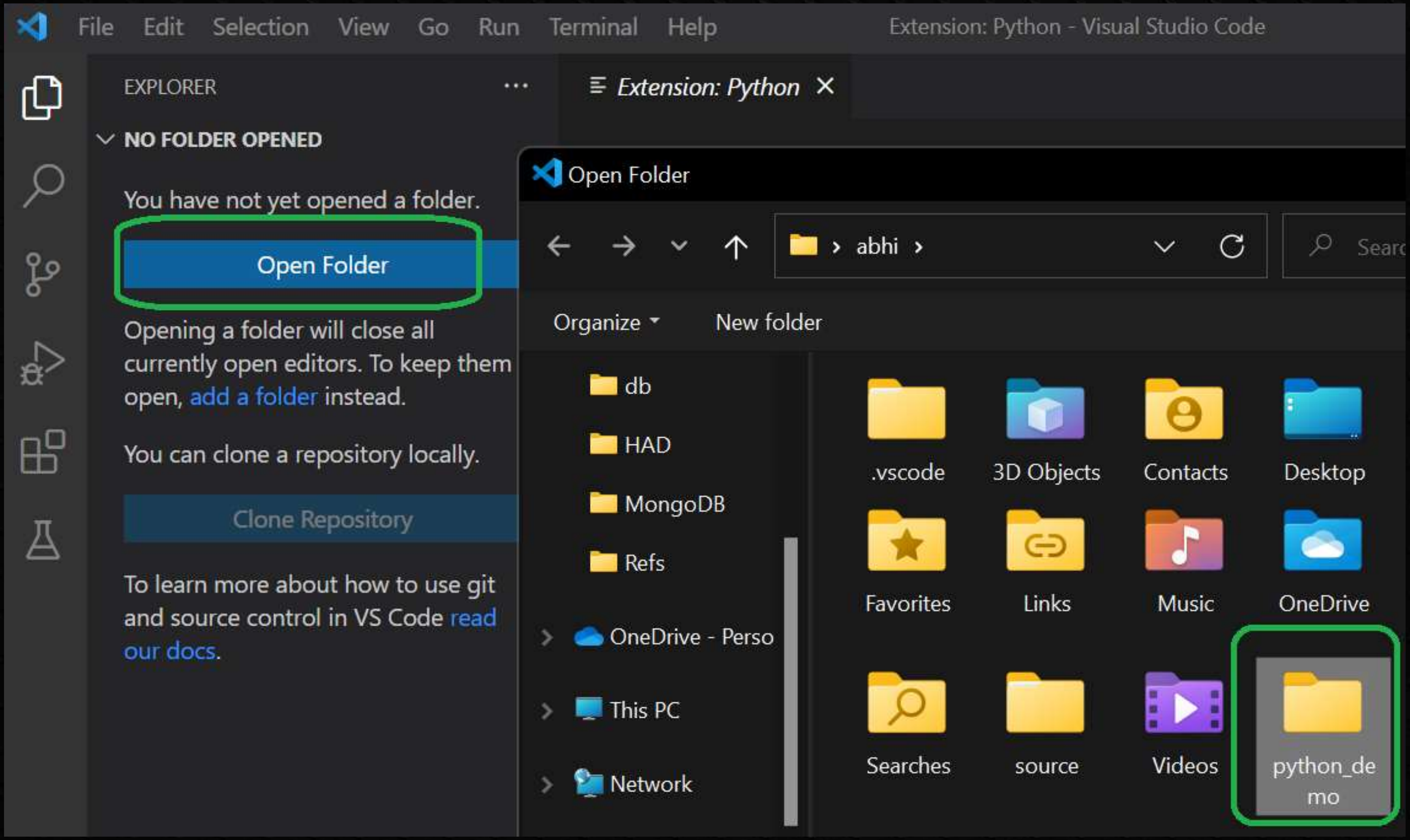
Python v2022.4.0
Microsoft | 53,538,569 | ★★★★☆ (468)
IntelliSense (Pylance), Linting, Debugging (multi-threaded,
Install

Details Feature Contributions Changelog Extension Pack

Python extension for Visual Studio Code

Categories
Programming Languages

Step 4: Open a New File and Test



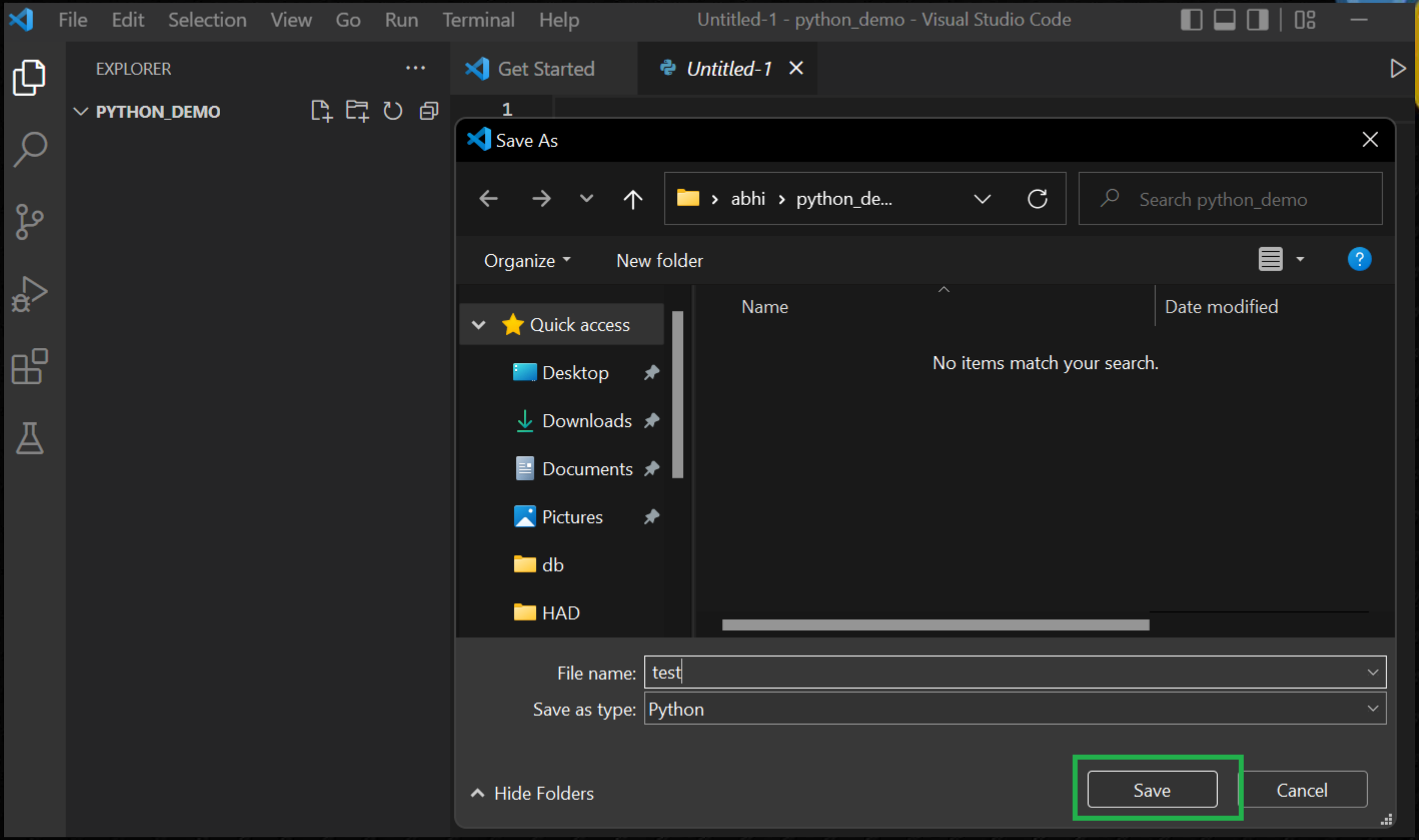
You have not yet opened a folder.

Opening a folder will close all currently open editors. To keep them open, add a folder instead.

You can clone a repository locally.

To learn more about how to use git and source control in VS Code read our docs.

Step 4: Open a New File and Test

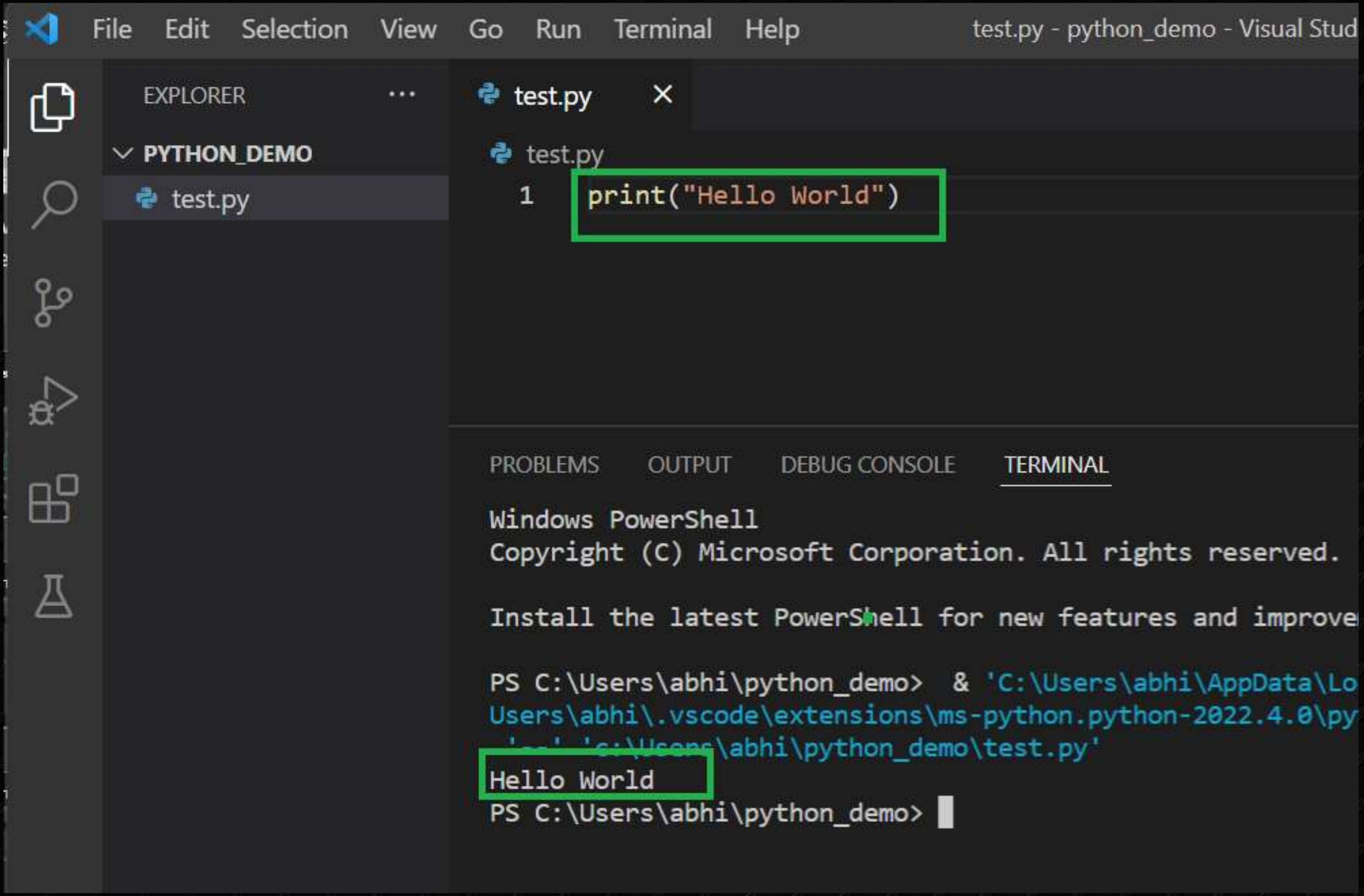


No items match your search.

File name: test

Save as type: Python

Step 4: Open a New File and Test



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

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improve

PS C:\Users\abhi\python_demo>  & 'C:\Users\abhi\AppData\Lo
Users\abhi\.vscode\extensions\ms-python.python-2022.4.0\py
 '--' 'c:\Users\abhi\python_demo\test.py'
Hello World
PS C:\Users\abhi\python_demo> 
```

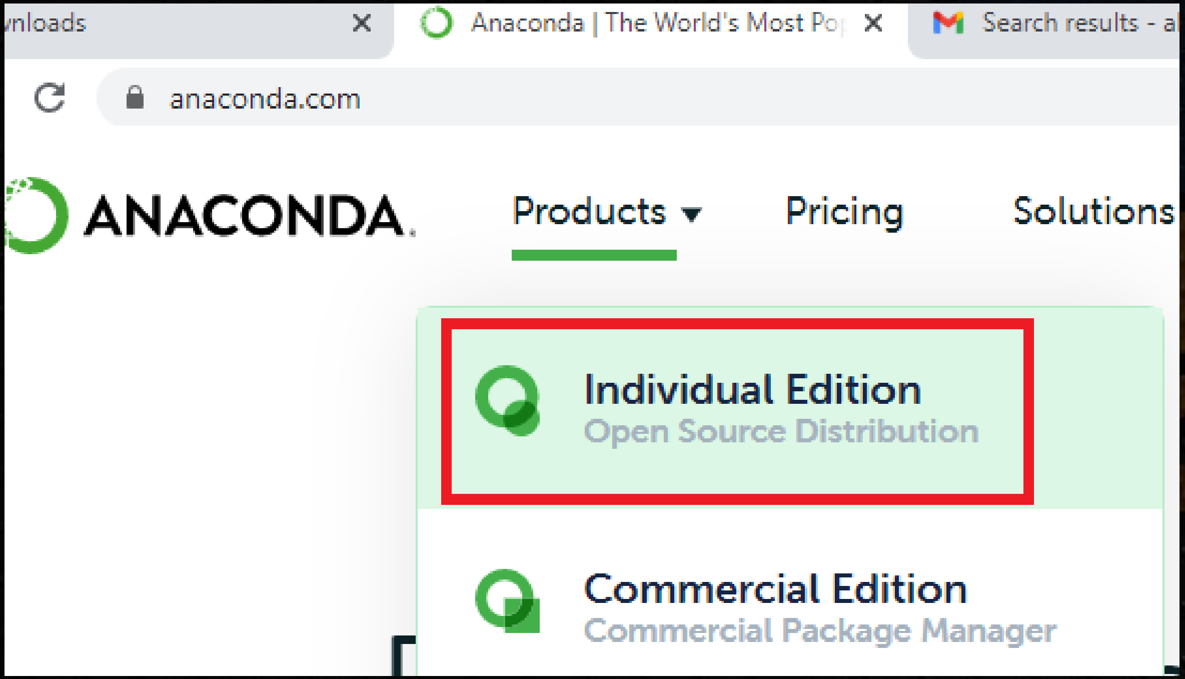
Python Fundamentals

Setting Up the Environment - Anaconda

Anaconda Distribution

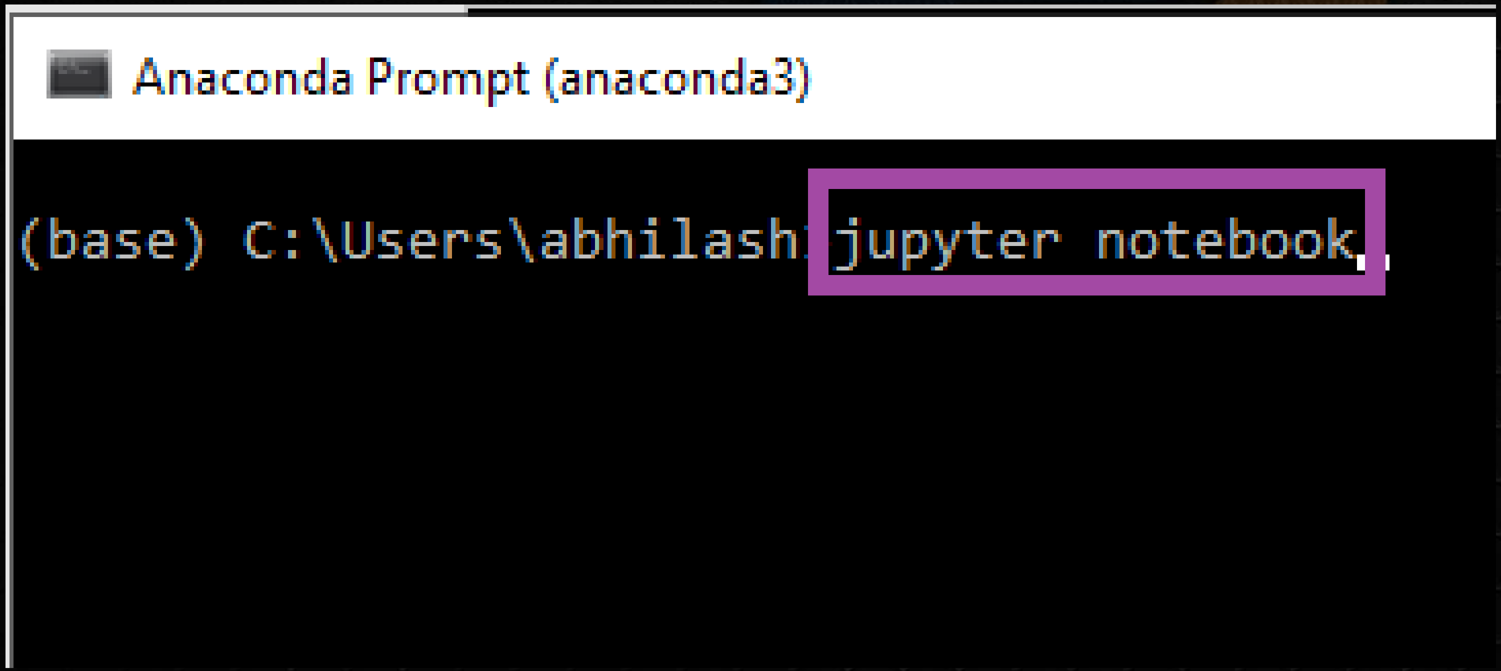
- The open-source Anaconda Distribution is the easiest way to install Python Environment on Linux, Windows, and Mac OS X.
- Anaconda has over 15 million users worldwide.
- It is the industry standard for developing Python based scientific as well as Machine Learning applications

Downloading and Installing Anaconda



Launching Jupyter Notebook

- An open-source web application that is used to create and share documents that contain live code, equations, visualizations and narrative text.

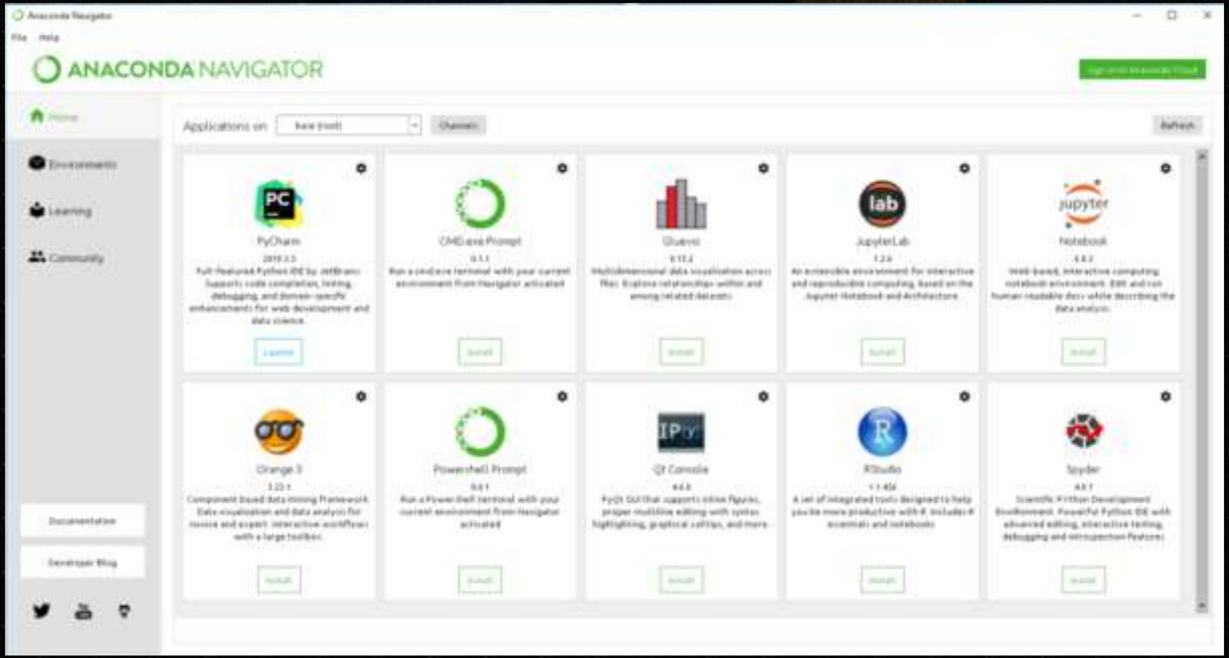


Anaconda Prompt (anaconda3)

```
(base) C:\Users\abhilash>jupyter notebook
```

Anaconda Navigator

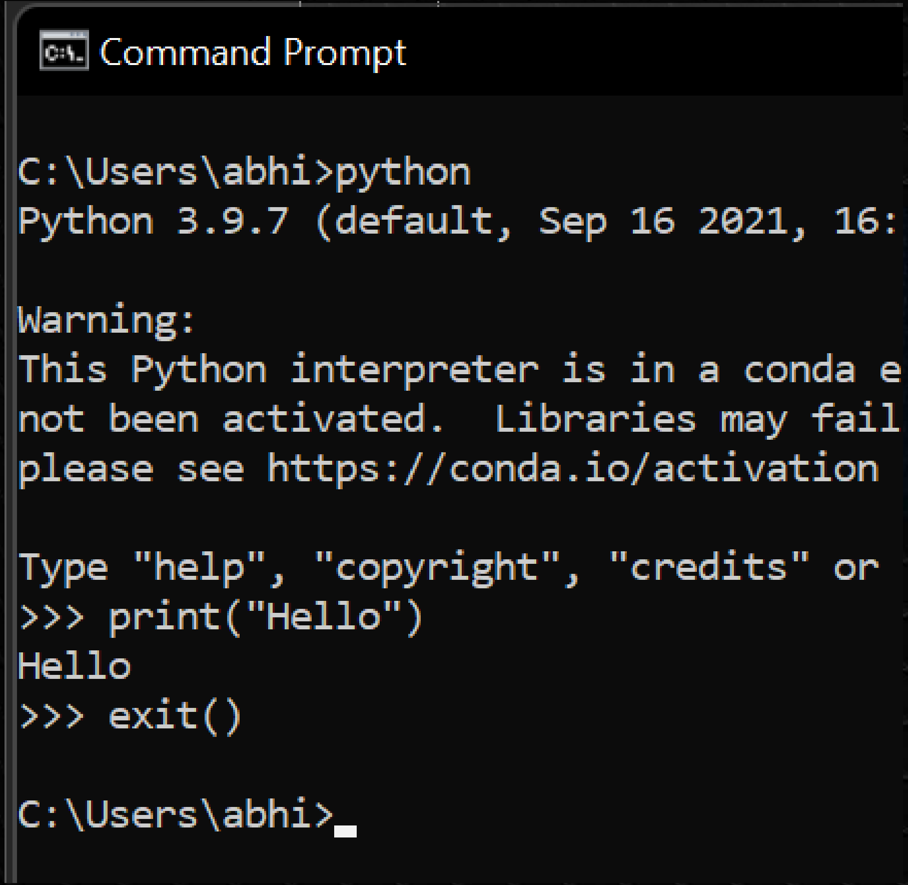
- Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda
- Helps to launch applications like Jupyter Notebook and easily manage conda packages



Test Python and Check Version

```
import sys


print("Hello World")
print(sys.version)
```

Test with command line



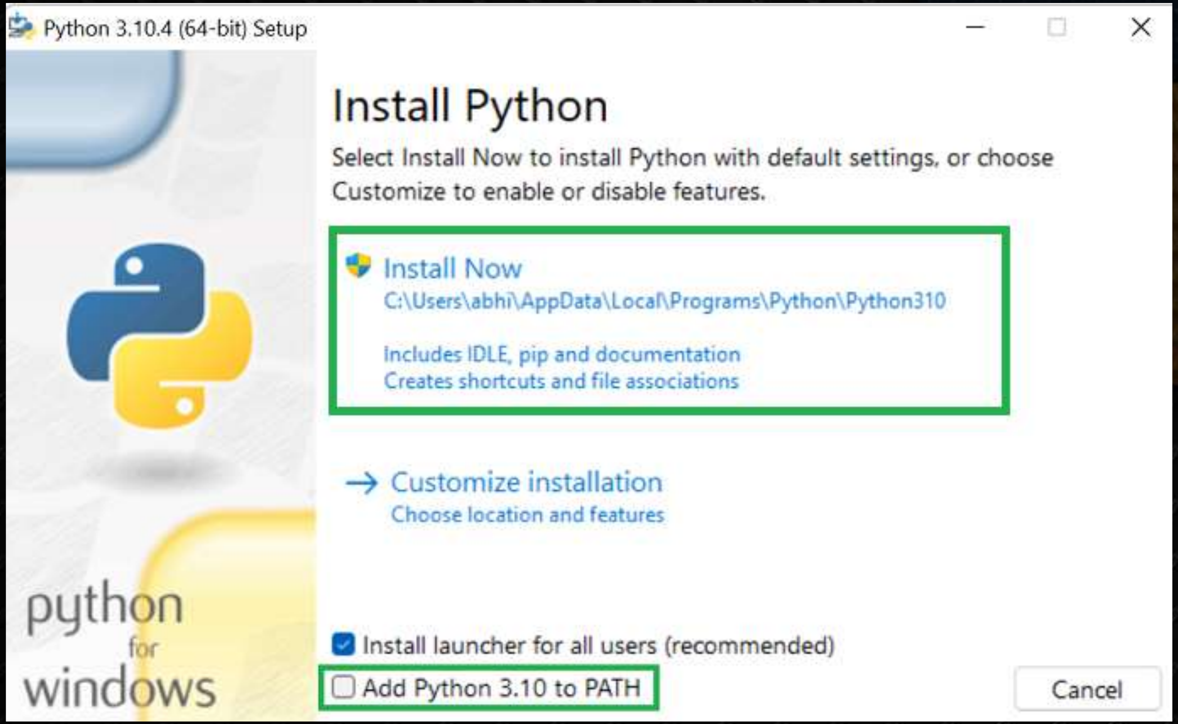
```
C:\Users\abhi>python
Python 3.9.7 (default, Sep 16 2021, 16:

Warning:
This Python interpreter is in a conda e
not been activated.  Libraries may fail
please see https://conda.io/activation

Type "help", "copyright", "credits" or
>>> print("Hello")
Hello
>>> exit()

C:\Users\abhi>_
```

Python Fundamentals

Setting Up the Environment - PyCharm

Step 1: Installing Python Interpreter

- **Install Python from python.org**
- **Click the Download Python to download the latest version.**
- **Make sure to add the python to system path**



Python 3.10.4 (64-bit) Setup

Install Python

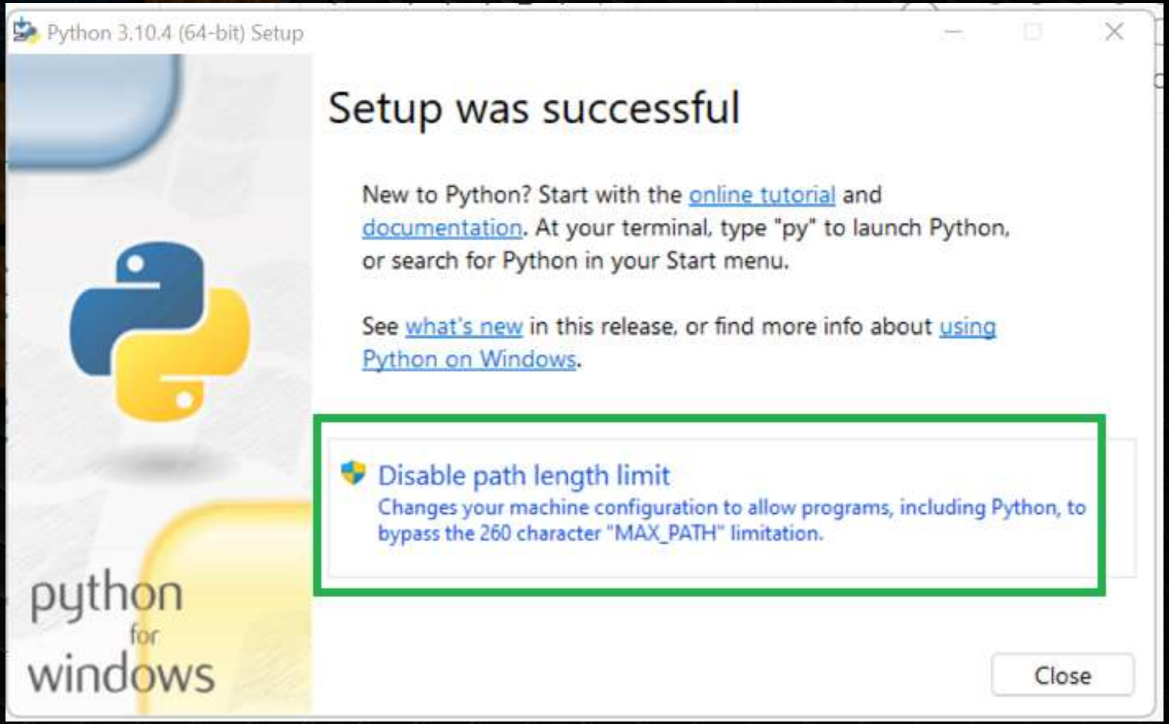
Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\abhi\AppData\Local\Programs\Python\Python310

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

[✓] Install launcher for all users (recommended)
[ ] Add Python 3.10 to PATH



Python 3.10.4 (64-bit) Setup

Setup was successful

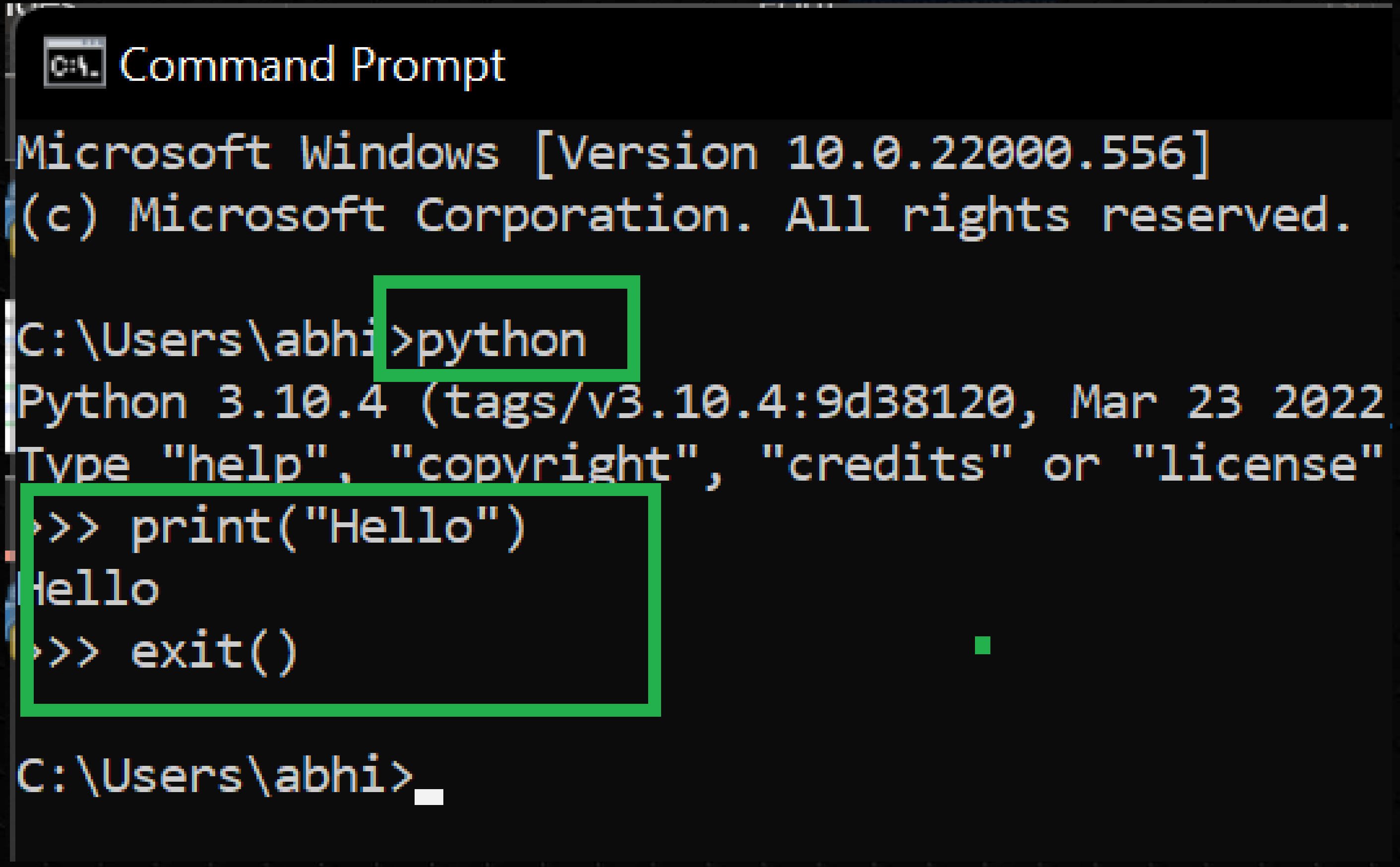
New to Python? Start with the online tutorial and documentation. At your terminal, type "py" to launch Python, or search for Python in your Start menu.

See what's new in this release, or find more info about using Python on Windows.

Disable path length limit
Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX_PATH" limitation.

Step 1: Test Installation

- **Open Command prompt and type 'python'**



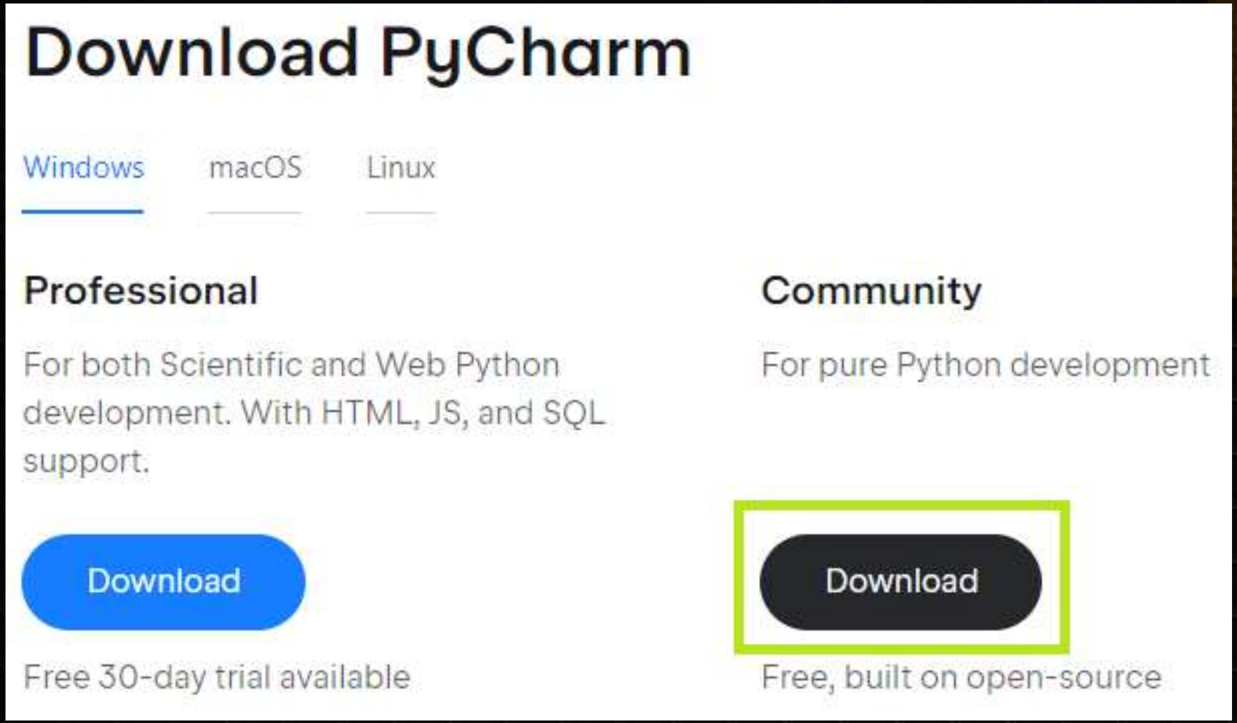
```
Command Prompt
Microsoft Windows [Version 10.0.22000.556]
(c) Microsoft Corporation. All rights reserved.

C:\Users\abhi>python
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022
Type "help", "copyright", "credits" or "license"
>>> print("Hello")
Hello
>>> exit()

C:\Users\abhi>_
```

Step 2 : Install PyCharm IDE

- PyCharm is an integrated development environment (IDE) used for Python programming.
- It is developed by the Czech company JetBrains..
- In addition to Python, PyCharm supports JavaScript, CoffeeScript, TypeScript, Cython, SQL, HTML/CSS, template languages, AngularJS, Node.js

Downloading PyCharm

https://www.jetbrains.com/pycharm/download



Download PyCharm

Windows macOS Linux

Professional

For both Scientific and Web Python development. With HTML, JS, and SQL support.

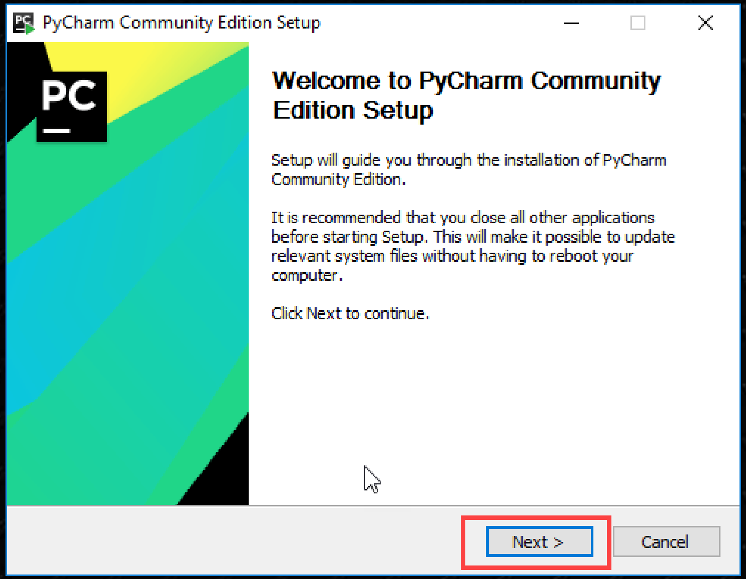
Free 30-day trial available

Community

For pure Python development

Free, built on open-source

Installing PyCharm



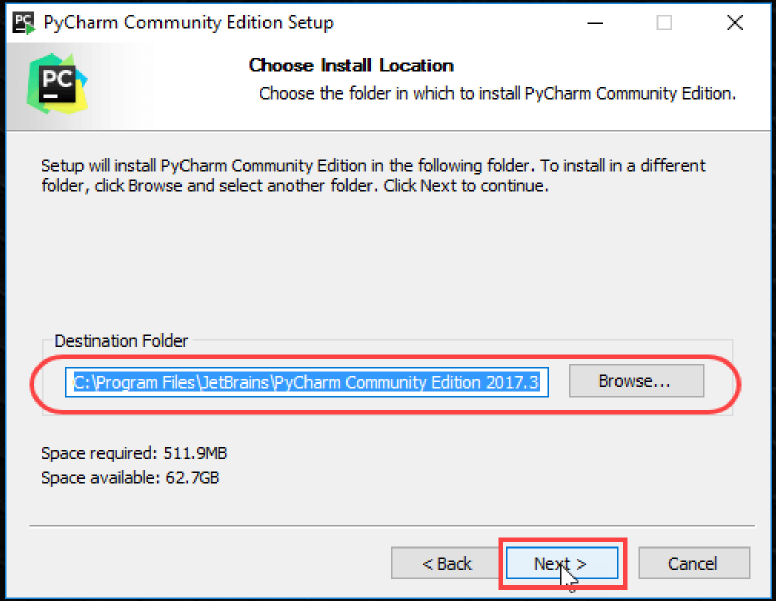
PyCharm Community Edition Setup

Welcome to PyCharm Community Edition Setup

Setup will guide you through the installation of PyCharm Community Edition.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.



PyCharm Community Edition Setup

Choose Install Location

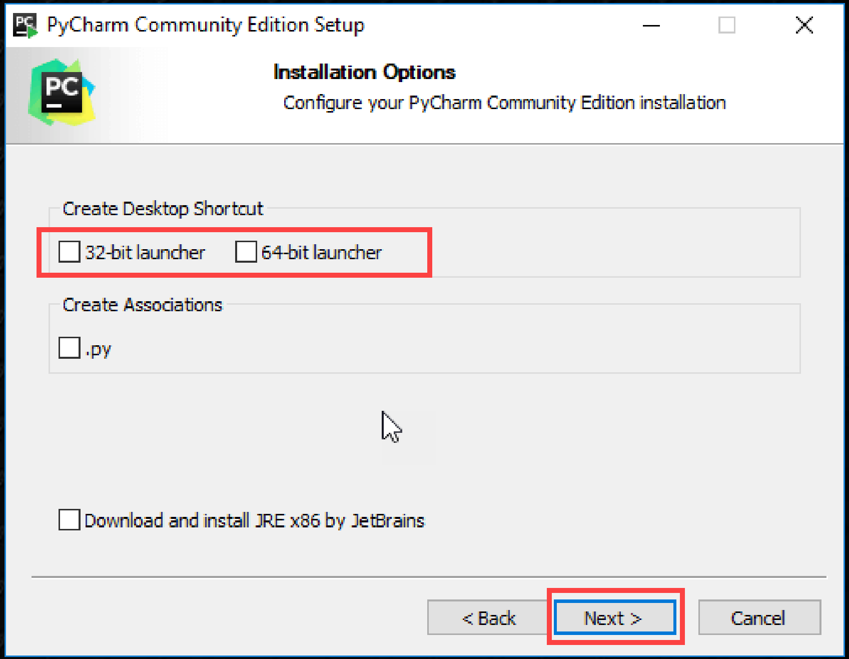
Choose the folder in which to install PyCharm Community Edition.

Setup will install PyCharm Community Edition in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder: C:\Program Files\JetBrains\PyCharm Community Edition 2017.3

Space required: 511.9MB

Space available: 62.7GB



PyCharm Community Edition Setup

Installation Options

Configure your PyCharm Community Edition installation

Create Desktop Shortcut

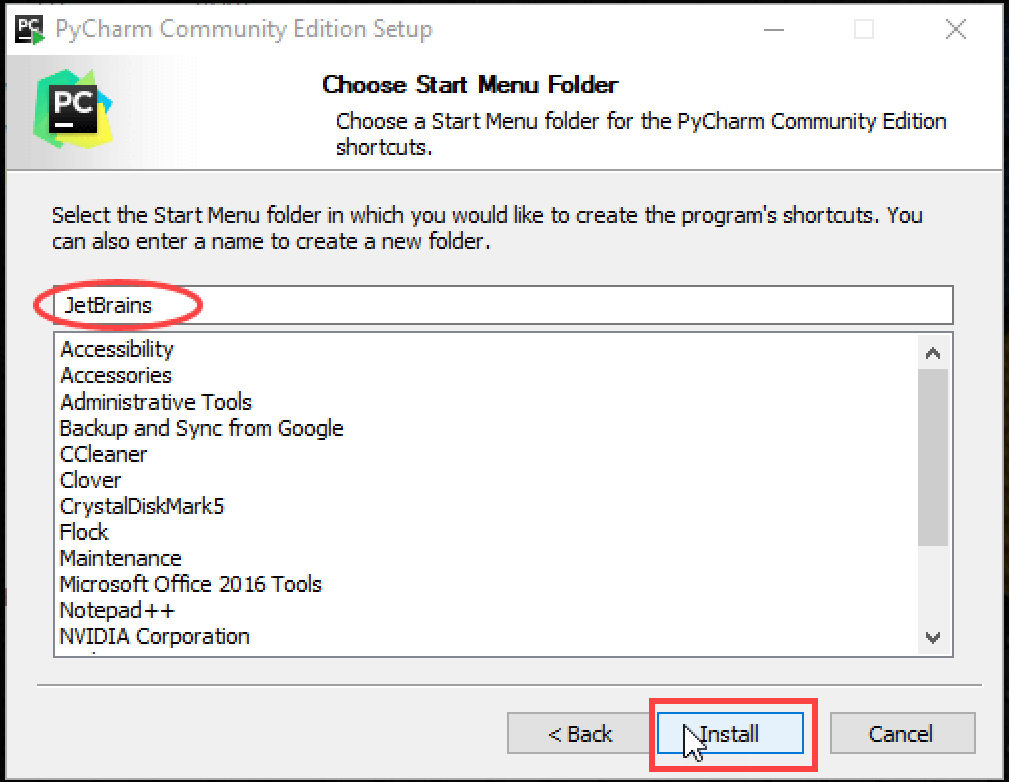
- [ ] 32-bit launcher
- [ ] 64-bit launcher

Create Associations

- [ ] .py

- [ ] Download and install JRE x86 by JetBrains

Installing PyCharm



PyCharm Community Edition Setup

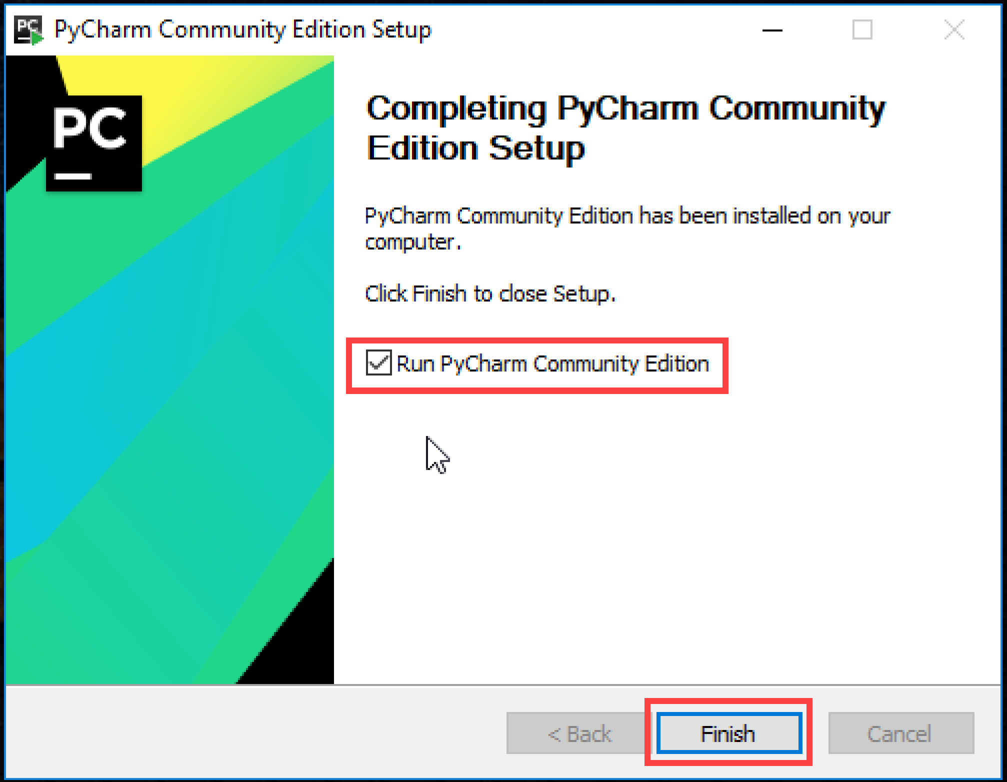
Choose Start Menu Folder

Choose a Start Menu folder for the PyCharm Community Edition shortcuts.

Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.

JetBrains

Accessibility
Accessories
Administrative Tools
Backup and Sync from Google
CCleaner
Clover
CrystalDiskMark5
Flock
Maintenance
Microsoft Office 2016 Tools
Notepad++
NVIDIA Corporation



PyCharm Community Edition Setup

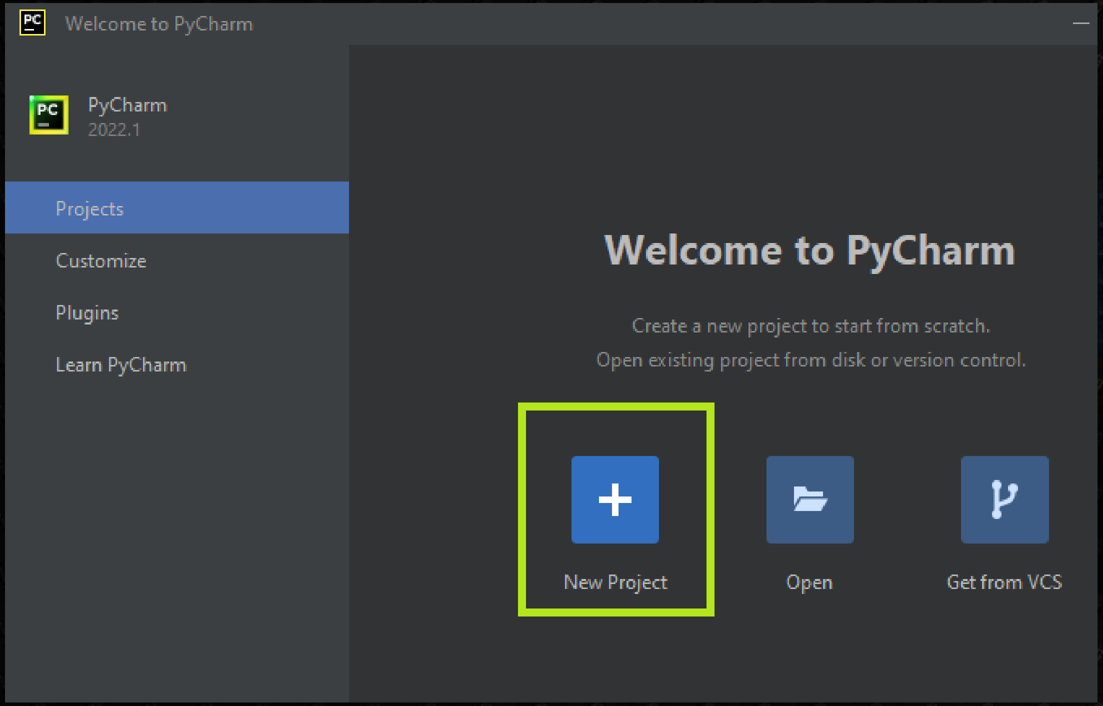
Completing PyCharm Community Edition Setup

PyCharm Community Edition has been installed on your computer.

Click Finish to close Setup.

[✓] Run PyCharm Community Edition

Installing PyCharm

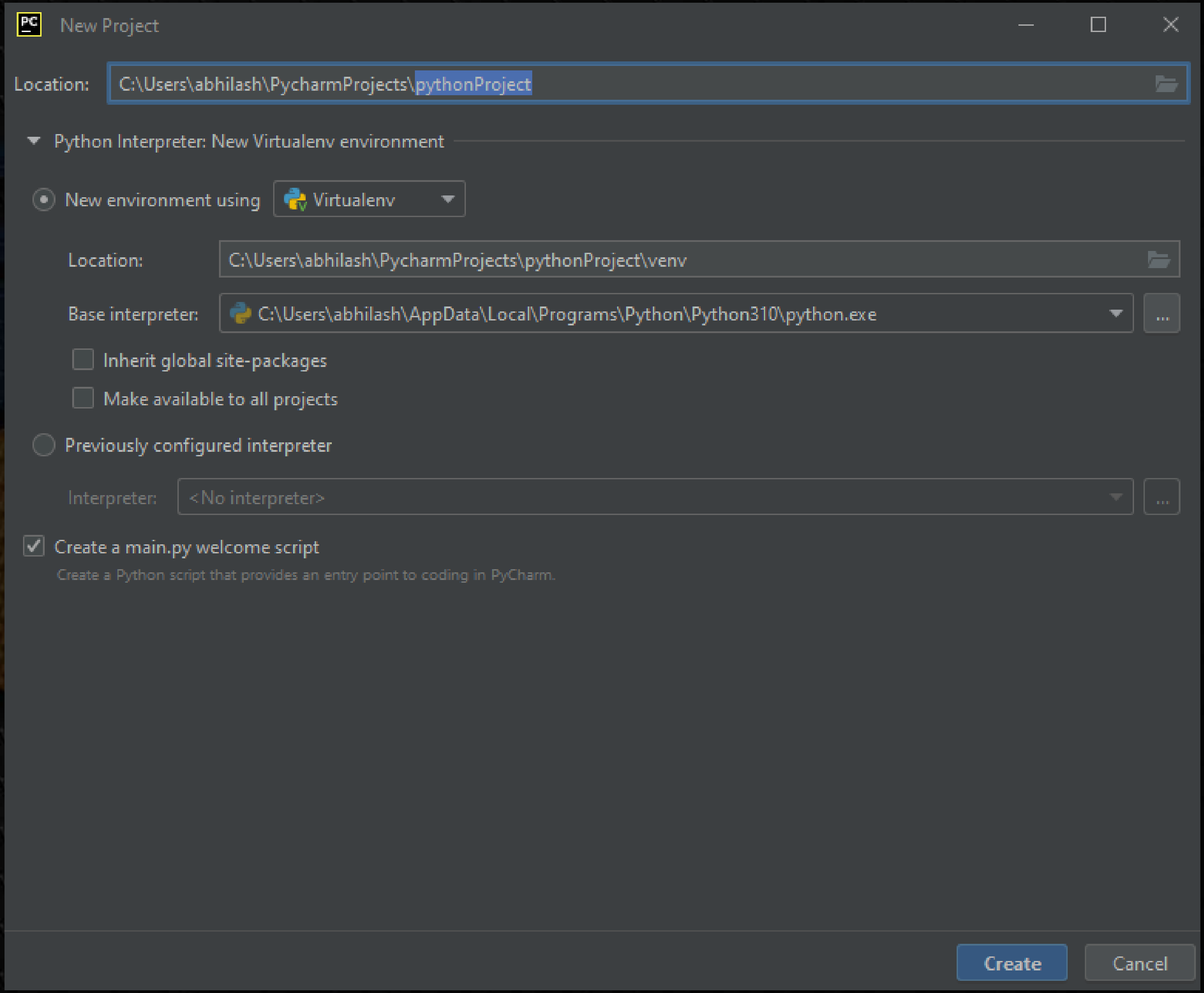

Welcome to PyCharm

PyCharm
2022.1

- Projects
- Customize
- Plugins
- Learn PyCharm

Welcome to PyCharm

Create a new project to start from scratch.
Open existing project from disk or version control.


New Project

Location: C:\Users\abhilash\PycharmProjects\pythonProject

Python Interpreter: New Virtualenv environment

(●) New environment using Virtualenv

Location: C:\Users\abhilash\PycharmProjects\pythonProject\venv

Base interpreter: C:\Users\abhilash\AppData\Local\Programs\Python\Python310\python.exe

☐ Inherit global site-packages

☐ Make available to all projects

( ) Previously configured interpreter

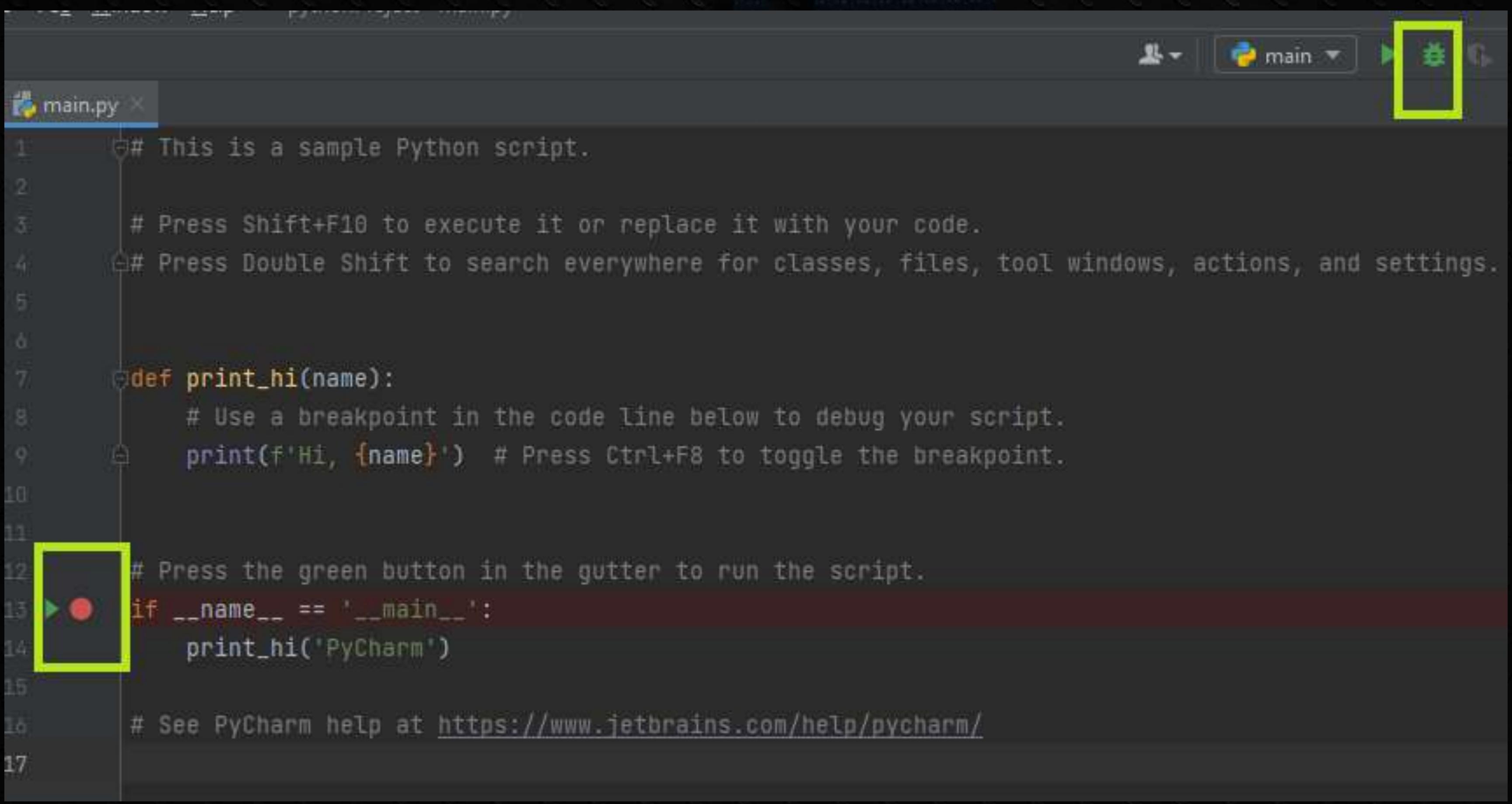
Interpreter: <No interpreter>

☑ Create a main.py welcome script

Create a Python script that provides an entry point to coding in PyCharm.

Anaconda Navigator

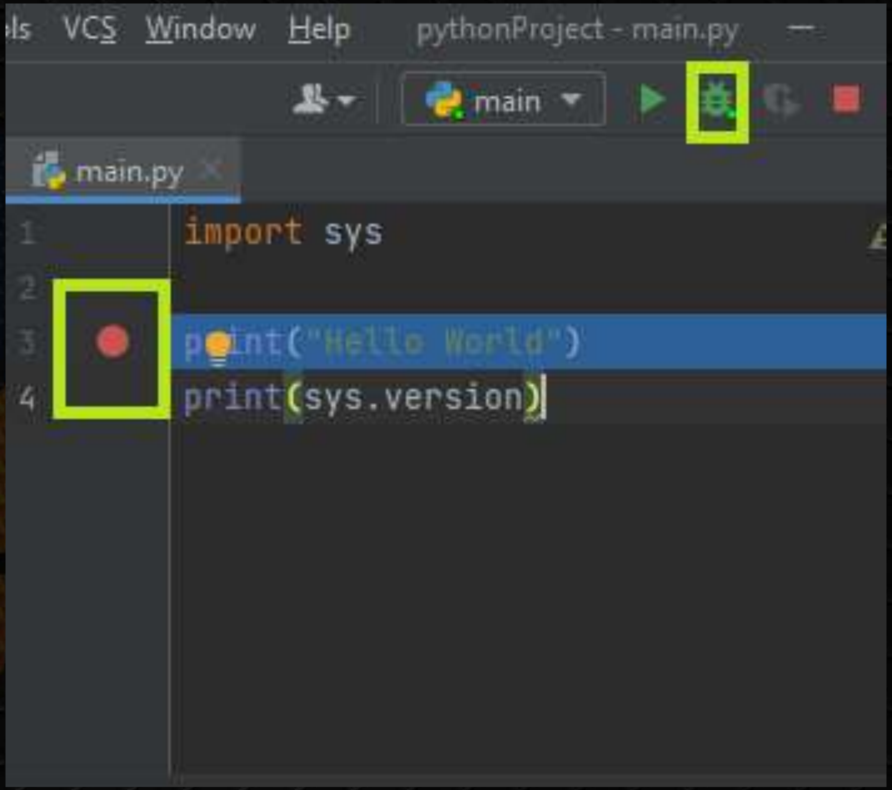
- Try running a test program and also try using the debugger



```
# This is a sample Python script.

# Press Shift+F10 to execute it or replace it with your code.
# Press Double Shift to search everywhere for classes, files, tool windows, actions, and settings.


def print_hi(name):
    # Use a breakpoint in the code line below to debug your script.
    print(f'Hi, {name}')  # Press Ctrl+F8 to toggle the breakpoint.


# Press the green button in the gutter to run the script.
if __name__ == '__main__':
    print_hi('PyCharm')

# See PyCharm help at https://www.jetbrains.com/help/pycharm/
```

Test Python and Check Version

```
import sys

print("Hello World")
print(sys.version)
```



```
import sys

print("Hello World")
print(sys.version)
```

Python Fundamentals

COMMENTS

Python Comments

- Single Line Comments Start with a pound sign.
- Multi Line Comments (docstrings) enclosed in triple quotation marks (Not actually comment, but python will ignore unassigned strings)

```
# This is a single line Python comment

""" This is a Multi line Python comment
      sometimes called a docstring """
```

Python Fundamentals

VARIABLES

Using Variables in Python

- Variable names can contain only letters, numbers, and underscores.
- Spaces are not allowed in variable names
- Avoid using Python keywords and function names as variable names
- Be careful when using the lowercase letter l and the uppercase letter because its case sensitive

Using Variables in Python : Assignment

```
message = "Hello  world!"
print(message)

my_money = 100                          # An integer
my_bal   = 1000.0                       # A float
my_name     = "Abhi"                    # A string
num1 = num2 = num3 = 100                # Multiple Assignment
num1,num2, my_name = 10,20,"abhi"

print (my_money)
print (my_bal)
print (my_name)
print (b)
```

Using Variables in Python : Assignment

```
x = x + 2
we can also write
x += 2


x = x - 2
we can also write
x -= 2
```

Variables in Python : Exercise

Create a simple program to store various details of a student and print them.

1) Name
2) Roll No
3) Semester
4) Department
5) Phone no
6) Email
7) Address

Python Fundamentals

STANDARD DATA TYPES

Python Standard Data Types

- Data Type define the type of data stored in the Memory.

- Python has five standard data types :

  1. Numbers
  2. Strings
  3. List
  4. Tuple
  5. Dictionary

Python Fundamentals

NUMBERS

Python Numbers Overview

- Number data types store numeric values.
- It is created when we assign a value to them.
- Using del, we can remove the reference to a variable

```
A = 10
B = 20

print(A)
del A,B
print(A)
```

Python Numerical Types

- Python supports three numerical types

1. int (signed integers)
2. float (floating point real values)
3. complex (complex numbers)

```
Int_eg = 100
float_eg = 3.14
comp_eg = 3.14j
```

Python Numerical Types Conversion

- Python converts an expression containing mixed types to a common type for evaluation.

1. int(x) convert x to a plain integer.

1. float(x) convert x to a floating-point number.

2. complex(x) convert x to a complex number with real part x and imaginary part zero.

3. complex(x, y) to convert x and y to a complex number with real part x and imaginary part y. x and y are numeric expressions

Python Types Conversion

```
int_eg = 100
float_eg = 3.14

print(int(float_eg))
print(float(int_eg))
Print(str(int_eg))
```

Python Basic Arithmetic Operators

```
x = 7, y = 2

Addition: x + y = 9
Subtraction: x - y = 5
Multiplication: x*y = 14
Division: x/y = 3.5

Floor Division: x//y = 3 (rounds to the nearest)
Modulus: x%y = 1 (gives the remainder when divide)
Exponent: x**y = 49 (7 to the power of 2)
```

Numbers and Arithmetic Operators Exercise

Create a simple program to solve the following mathematical problems

1) Assign A and B with values
2) Find the solution for $(A+B)^2$

$$(A+B)^2 = A^2 + 2*A*B + B^2$$

Python Fundamentals

STRINGS

Strings

- A string is a series of characters.
- Anything inside quotes is considered a string in Python
- We can use single or double quotes around your strings like this

```
"A sample string."
'Another sample string.'
'I informed him that, "Python is very easy" '
```

Strings – Changing Case

- We can change a string to title case or all uppercase or all lowercase letters like this:

```
name = "Abhilash Nelson"
print(name.title())
print(name.upper())
print(name.lower())
```

Strings – Using Variables

- We can use variables to represent a string:
- Using f-strings (f is format), Python formats the string by replacing the name of any variable in braces with its value.
- Place the letter f immediately before the opening quotation mark

```
first_name = "Abhilash"
last_name = "Nelson"

my_name = f"{first_name} {last_name}"

print(my_name)
print(f"Hello, {my_name.title()}!")
```

Strings – Adding Tab and New Line

- To add a tab to your text, use the characters \t
- To add a newline, use the characters \n

```
print("Apples\tOranges");
print("Apples\nOranges");
```

Strings – Removing white space

- Remove white space from right end using the rstrip() method
- Remove white space from left end using the lstrip() method
- Remove white space from both ends the strip() method

```
fruits = (" Apples Oranges ");
print(fruits.lstrip())
print(fruits.rstrip())
print(fruits.strip())
```

Strings – Slicing and Concatenation

```
greet = 'Good Morning!'

print (greet)
print (greet[0]) # first char
print (greet[2:5]) # chars from 3rd to 5th
print (greet[2:]) # from 3rd character
print (greet * 2) # Prints two times
print (greet + "WORLD") # concatenated string
```

Strings – Formatting using % operator

Strings can also be formatted using the % operator. This gives us control over how you want your string to be displayed and stored.

The syntax for using the % operator is:

"string to be formatted" %(values or variables to be inserted into string, separated by commas)

Strings – Formatting using % operator

Example:

```
make = 'Dell'
dollarRate = 70.256
myText = 'The amount for this %s computer is %d
USD and the exchange rate is %4.2f USD to 1
INR' %(make,1299, dollarRate)
print (myText)
```

The %s formatter is used to represent a string , %d formatter represents an integer and the %f formatter is used to format floats

(%4.2f where 4 refers to the total length of the string and 2 refers to 2 decimal places)

Strings – Formatting using format() method

Strings can also be formatted using the format() method.

The syntax is:

"string to be formatted".format(values or variables to be inserted into string, separated by commas)

Strings – Formatting using format() method

Example:

```
make = 'Dell'
dollarRate = 70.256
myText = 'The amount for this {0:s} computer is
{1:d} USD and the exchange rate is {2:4.2f} USD
to 1 INR' .format(make,1299, dollarRate)
print (myText)
```

({2:4.2f}, we are referring to the parameter in position 2, which is a float and we want it to be formatted with 2 decimal places and a total length of 4

Strings – Formatting using format() method

Example without any formatting:

```
make = 'Dell'

dollarRate = 70.256

myText = 'The amount for this {} computer is {}
USD and the exchange rate is {} USD to 1 INR'
.format(make,1299, dollarRate)

print (myText)
```

Strings – Formatting using format() method

Example without any formatting:

```
myText = '{0} is easier than
{1}'.format('Python','Java')

print (myText)
```

More String Methods – count()

Count method: Return the number of times the substring sub appears in the string.

count (substring, [start, [end]])

```
# count the entire string
print('Hello Good Morning'.count('d'))

#count from index 3 to end of string
print('Hello Good Morning'.count('d', 3))

# count from index 3 to 10-1
print('Hello Good Morning'.count('d', 3, 13 ))
```

More String Methods – endswith()

endswith method: Return True if the string ends with the specified suffix
endswith (suffix, [start, [end]])   (suffix can also be a tuple of suffixes)

```
'Superman'.endswith('man')   #true

#check from index 3 to end of string
'Superman'.endswith('man',3)   #true

#check from index 2 to 6-1
'Superman'.endswith('man', 2, 6)   # False

# Using a tuple of suffixes
'Postman'.endswith(('man', 'ma'), 2, 6) => True
```

More String Methods – find() or index()

Return the index in the string where the first occurrence of the substring is found.
find() returns -1 if sub is not found.
index() returns ValueError is sub is not found

```
'Hello Good Morning'.find('go')

# check from index 4 to end of string
'Hello Good Morning'.find('go',4)

# check from index 4 to 15
'Hello Good Morning'.find('go',4,15)

'This is a string'.find('kk')
=> -1
'This is a string'.index('kk')
=> ValueError
```

More String Methods – isalnum()

isalnum() return true if all characters in the string are alphanumeric (no spaces) and there is at least one character, false otherwise.

```
'hello1234'.isalnum()
⇒True

'h e l l o 1 2 3 4'.isalnum()
⇒False

'hello'.isalnum()
=> True
```

More String Methods – isalpha(), isdigit()

```
'hello'.isalpha()
=> True
'hello1234'.isalpha()
=> False
'1234'.isalpha()
=> False
'1234'.isdigit()
=> True
```

More String Methods – islower(), istitle(), isupper()

```
'hello'.islower()
⇒True

'Hello World'.istitle()
⇒True

'HELLO'.isupper()
=> True
```

More String Methods – join(), upper(), lower()

parameter provided is joined by a separator.

```
separator = '-'
myTuple = ('h', 'e', 'l' ', 'l' ', '0')
separator.join(myTuple)
=> 'h-e-l-l-o'

'Hello'.lower()
'Hello'.upper()
```

More String Methods – replace(), split()

```
# Replace all occurences
'Hello world'.replace('o', 'i')

# Replace first 2 occurences
'Hello world'.replace('O', 'i', 2)

# Split using whitespace as delimiter
'Hello world'.split(' ') => ['Hello', 'world']
```