

## Practical-1

**Aim: To study various Python implementation and their setup.**

### What is Python?

Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

### Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-oriented way or a functional way.

### Python Implementations

Python currently has four production-quality implementations (CPython, Jython, IronPython, PyPy) and new high-performance implementation in early development, Pyston, which we do not cover further. We also mention a number of other, more experimental, implementations in the following section.

This book primarily addresses CPython, the most widely used implementation, which we usually refer to as just “Python” for simplicity. However, the distinction between a language and its implementations is an important one.

#### (1.) CPython

Classic Python (AKA CPython, often just called Python) is the most up-to-date, solid, and complete production-quality implementation of Python. It can be considered the “reference implementation” of the language. CPython is a compiler, interpreter, and set of built-in and optional extension modules, all coded in standard C.

CPython can be used on any platform where the C compiler complies with the ISO/IEC 9899:1990 standard (i.e., all modern, popular platforms). In “Installation”

on page 17, we explain how to download and install CPython. All of this book, except a few sections explicitly marked otherwise, applies to CPython. CPython supports both v3 (version 3.5 or better) and v2 (version 2.7).

## **(2.) Jython:**

Jython is a Python implementation for any Java Virtual Machine (JVM) compliant with Java 7 or better. Such JVMs are available for all popular, modern platforms. As

of this writing, Jython supports v2, but not yet v3. With Jython, you can use all Java

libraries and frameworks. For optimal use of Jython, you need some familiarity with

fundamental Java classes. You do not have to code in Java, but documentation and examples for Java libraries are couched in Java terms, so you need a nodding acquaintance with Java to read and understand them. You also need to use Javasupporting tools for tasks such as manipulating .jar files and signing applets. This

book deals with Python, not with Java. To use Jython, you should complement this

book with Jython Essentials, by Noel Rappin and Samuele Pedroni (O’Reilly), Java in

a Nutshell, by David Flanagan (O’Reilly), and/or The Definitive Guide to Jython (Open Source Version) by Josh Juneau, Jim Baker, et al. (Apress, available on O’Reil-

ly’s Safari and on Jython.org), as well as some of the many other Java resources available.

## **(3.) IronPython:**

IronPython is a Python implementation (as of this writing, only v2, not yet v3) for the Microsoft-designed Common Language Runtime (CLR), most commonly known as .NET, which is now open source and ported to Linux and macOS. With IronPython, you can use all CLR libraries and frameworks. In addition to Micro-

soft's own implementation, a cross-platform implementation of the CLR known as Mono works with other, non-Microsoft operating systems, as well as with Windows.

For optimal use of IronPython, you need some familiarity with fundamental CLR libraries. You do not have to code in C#, but documentation and examples for exist-

ing CLR libraries are often couched in C# terms, so you need a nodding acquaintance with C# to read and understand them. You also need to use CLR supporting tools for tasks such as making CLR assemblies. This book deals with Python, not with the CLR. For IronPython usage, you should complement this book with IronPython's own online documentation, and, if needed, some of the many other resour-

ces available about .NET, the CLR, C#, Mono, and so on.

#### **(4.) PyPy:**

PyPy is a fast and flexible implementation of Python, coded in a subset of Python itself, able to target several lower-level languages and virtual machines using advanced techniques such as type inferencing. PyPy's greatest strength is its ability

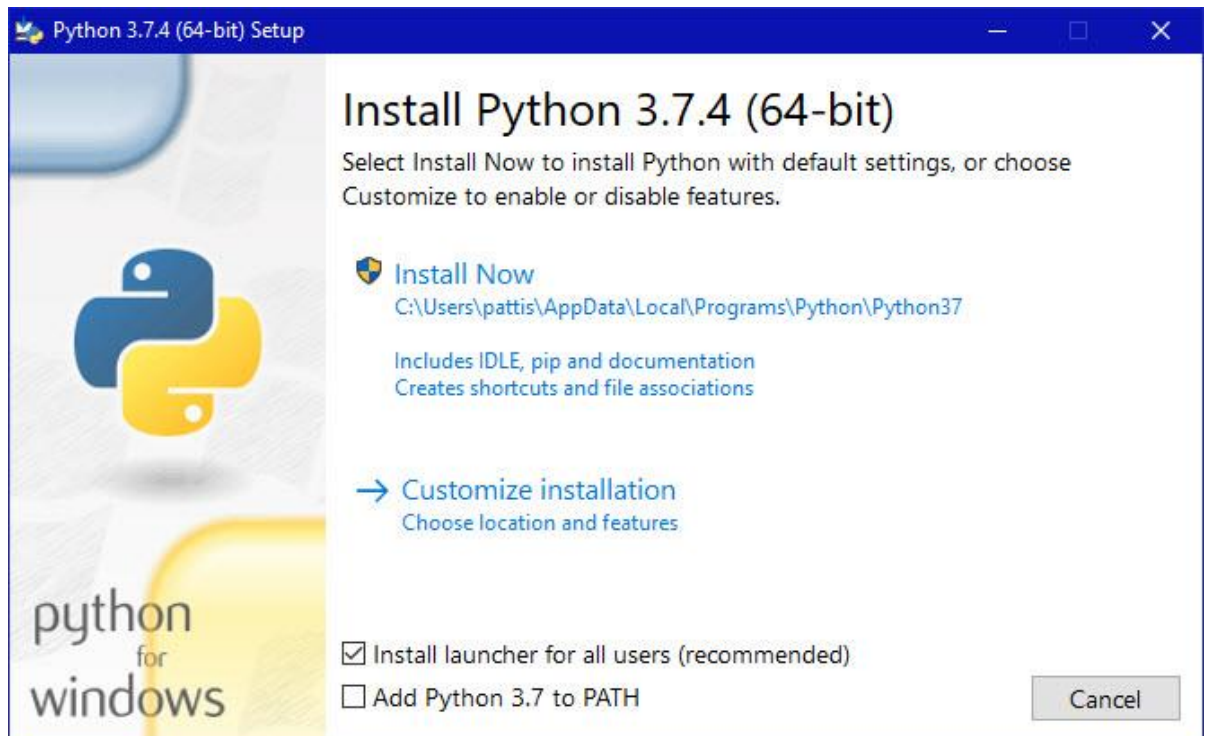
to generate native machine code "just in time" as it runs your Python program.

PyPy implements v2 and, at this writing, Python 3.5 support is being released in alpha. PyPy has substantial advantages in speed and memory management, and is seeing production-level use.

### **Installation of CPython:**

1. Double-click the icon labeling the file python-3.7.4-amd64.exe.

A Python 3.7.4 (64-bit) Setup pop-up window will appear.



Ensure that the Install launcher for all users (recommended) and the Add Python 3.7 to PATH checkboxes at the bottom are checked.

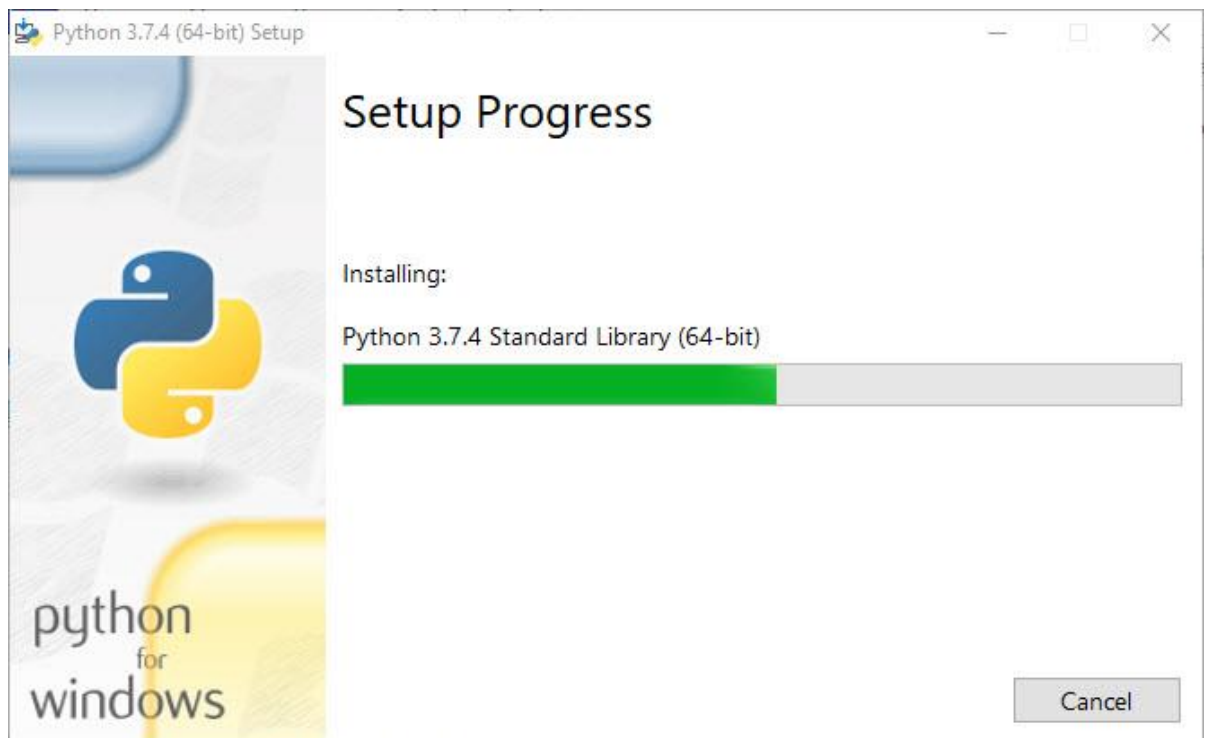
If the Python Installer finds an earlier version of Python installed on your computer, the Install Now message may instead appear as Upgrade Now (and the checkboxes will not appear).

2. Highlight the Install Now (or Upgrade Now) message, and then click it.

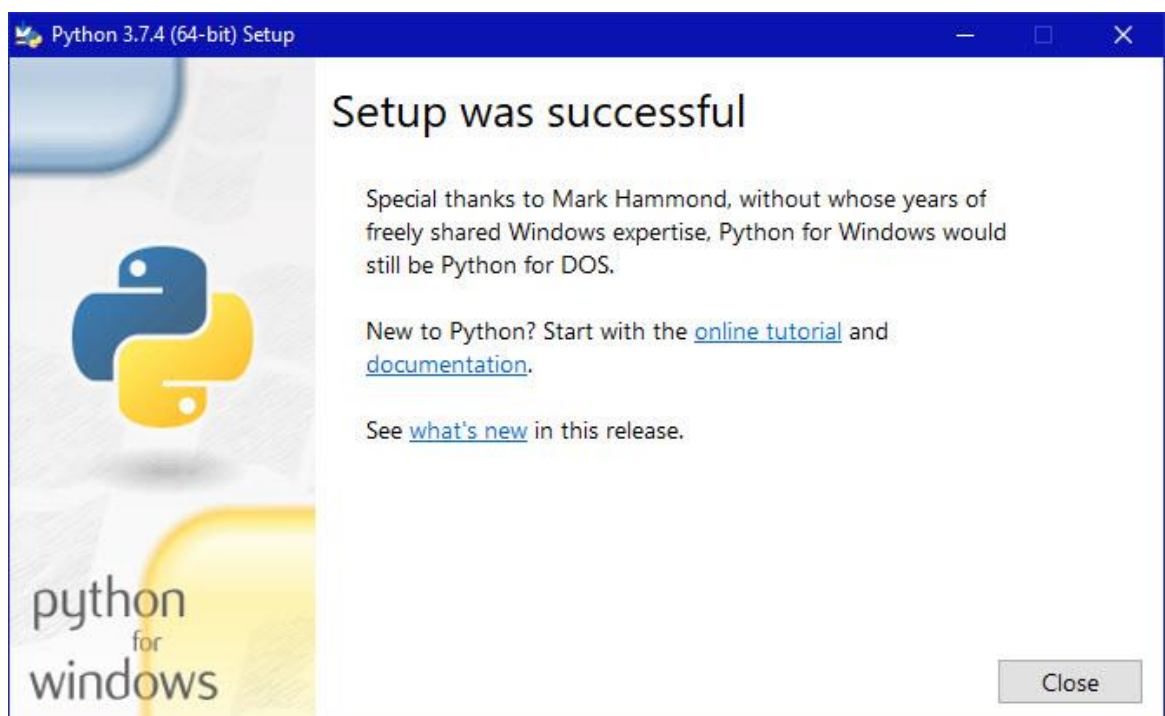
When run, a User Account Control pop-up window may appear on your screen. I could not capture its image, but it asks, Do you want to allow this app to make changes to your device.

3. Click the Yes button.

A new Python 3.7.4 (64-bit) Setup pop-up window will appear with a Setup Progress message and a progress bar.



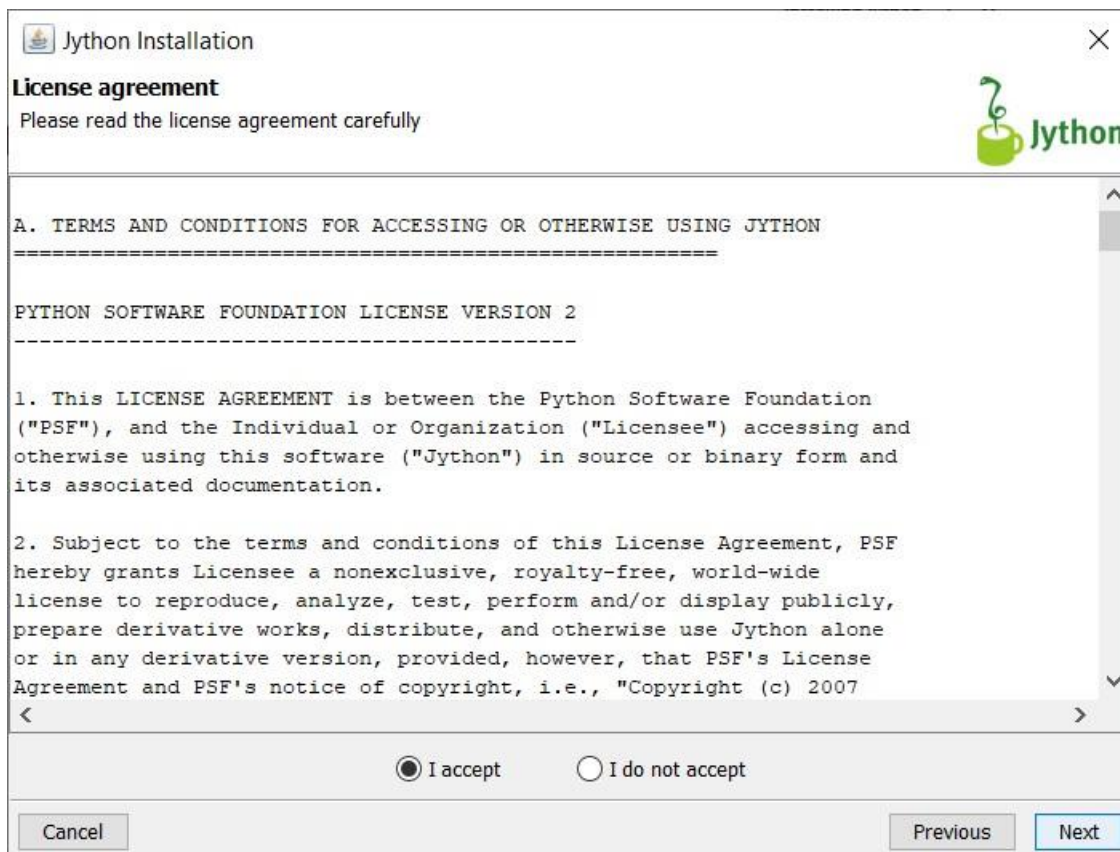
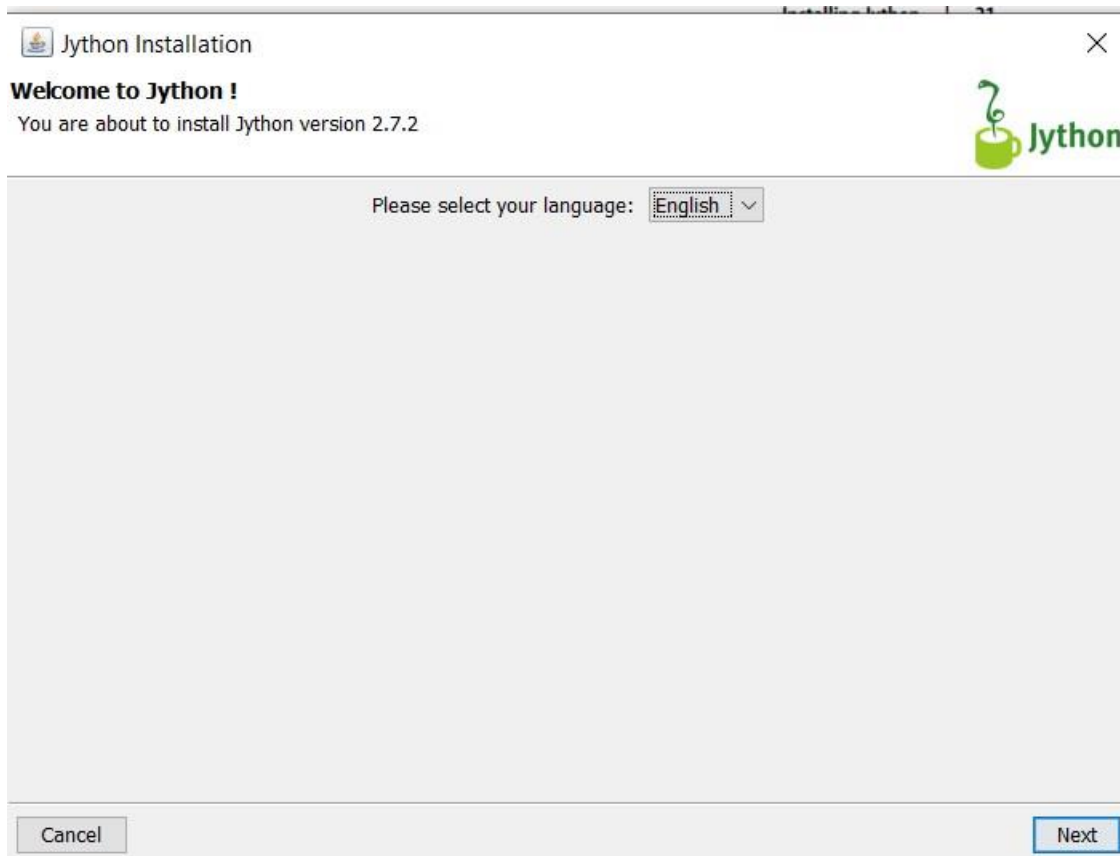
During installation, it will show the various components it is installing and move the progress bar towards completion. Soon, a new Python 3.7.4 (64-bit) Setup pop-up window will appear with a Setup was successfully message.




4. Click the Close button.

Python should now be installed.

## Installation of Jython:




 Jython Installation ✕

**Installation type**  
The following installation types are available

Please select the installation type:


☒ All (everything, including sources)  
☐ Standard (core, library modules, demos, examples, documentation)  
☐ Minimum (core)  
☐ Standalone (a callable .jar file)  
☐ Custom

☒ Core  
☒ Library modules  
☒ Demos and examples  
☒ Documentation  
☒ Sources  
☒ Install pip and setuptools


 Jython Installation ✕

**Target directory**  
Choose the location where you want Jython to be installed to

Please select the target directory:


 Jython Installation ✕

**Overview (summary of options)**  
The installation will be done using the following options




Target directory:	<input type="text" value="C:\jython2.7.2"/>		
Installation type:	<input type="text" value="All (everything, including sources)"/>		
OS name / version:	<input type="text" value="Windows 10"/>	<input type="text" value="10.0"/>	<input checked="" type="checkbox"/> Ok
Java vendor / version:	<input type="text" value="Oracle Corporation"/>	<input type="text" value="1.8.0_271"/>	<input checked="" type="checkbox"/> Ok

Please press Next to start the installation

 Jython Installation ✕

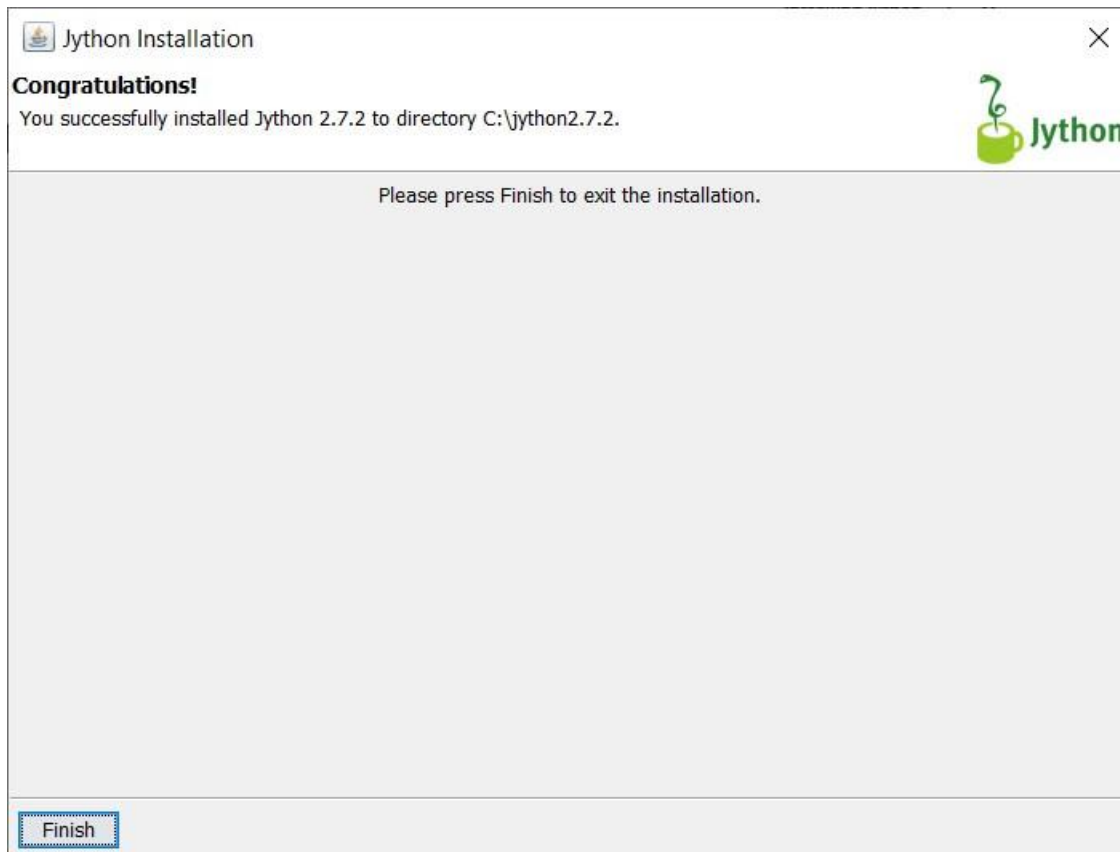
**The installation is now in progress**  
Please stand by, this may take a few seconds ...



Progress:  62%

Inflating C:\jython2.7.2\extlibs\guava-28.0-android.jar

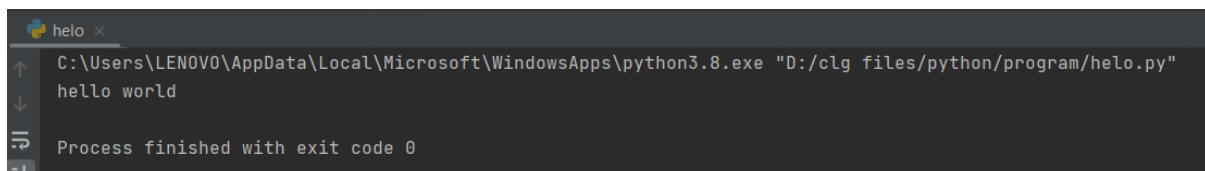




### Code:

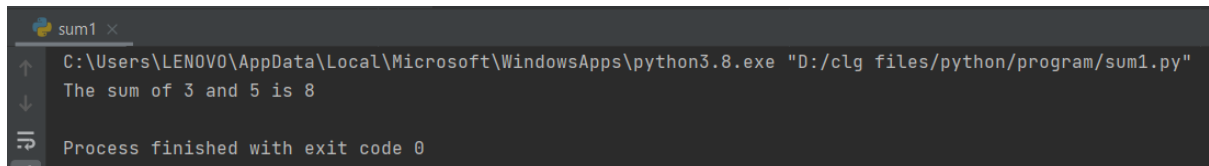
```
print('hello world')
```

### Output:



**Code:**

```
x=3  
y=5  
z=x+y  
print('The sum of {0} and {1} is {2}'.format(x, y, z))
```

**Output:**A screenshot of a Windows terminal window titled 'sum1'. The command prompt shows the execution of a Python script: 'C:\Users\LENOVO\AppData\Local\Microsoft\WindowsApps\python3.8.exe "D:/c/g files/python/program/sum1.py"'. The output of the script is 'The sum of 3 and 5 is 8'. At the bottom, it states 'Process finished with exit code 0'.

```
sum1 x  
C:\Users\LENOVO\AppData\Local\Microsoft\WindowsApps\python3.8.exe "D:/c/g files/python/program/sum1.py"  
The sum of 3 and 5 is 8  
Process finished with exit code 0
```