# **Programming with Python**

# Module 01

In this module, you will learn about what Python is and how it is used.

## What is Python

Python Programming Language – Official Website

"Python is a programming language that lets you work more quickly and integrate your systems more effectively. You can learn to use Python and see almost immediate gains in productivity and lower maintenance costs.

* Python runs on Windows, Linux/Unix, Mac OS X
* Python is free to use, even for commercial products, because of its OSI-approved open source license
* There are two main versions of Python 2.x and 3.x. Both work, but 3.x has improved and advanced features.
* You can find out more about the difference from the Python website here: <http://wiki.python.org/moin/Python2orPython3> (opens external site)
* Our textbook uses version 3.x, but Mac already has 2.x installed. I recommend uninstalling and using 3.x <https://docs.python.org/3/using/mac.html> (opens external site)

## Installing Python

Installation is easy, but you can always search for video tutorials based on your chosen OS. <https://www.google.com/search?q=How+to+install+python> (opens external site)

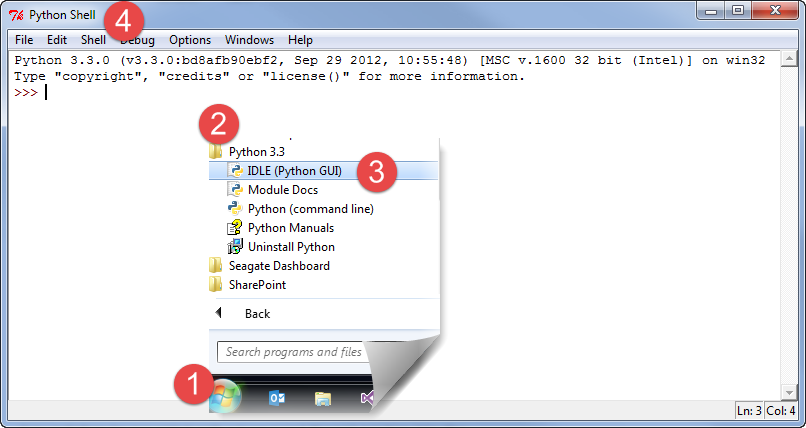
## Running Python

You can create and run python code using its built-in editor called IDLE.

<https://www.google.com/search?q=How+to+use+Python+idle> (opens external site)

One Windows, the basic steps are:

1. Open the Start menu
2. Open the Start menu locate or search for IDLE under Python
3. Launch IDLE from the link you found
4. Wait for the IDLE Python Shell to open

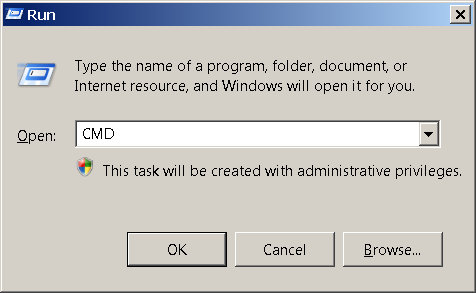


#### Figure 1. Steps to open the IDLE Python Shell Application

## Console Applications

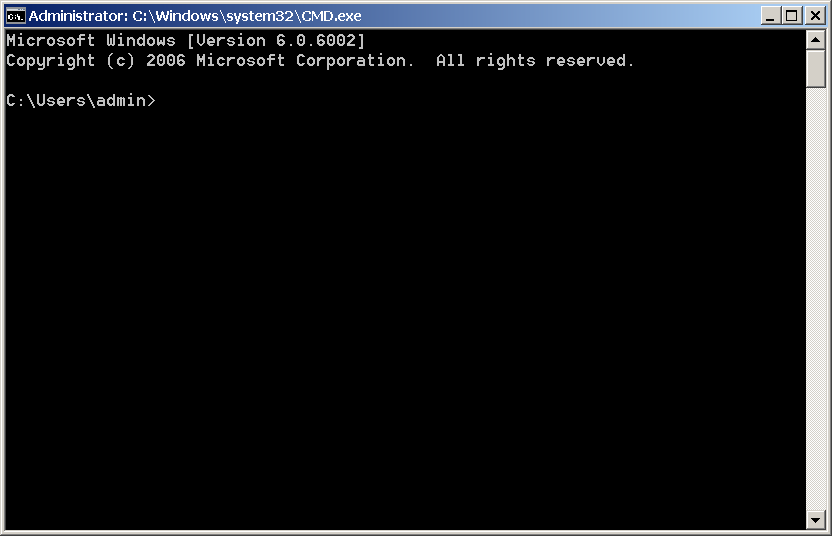
Once you have installed Python, you can create a program that runs as a Console application (think “Command Prompt”). The Windows program IPConfig.exe is an example of a console application.

To start, in Windows, you open a console window using Start Menu ➤ Run (windows key + r) and type in the following command “CMD”.



#### Figure 2. The Windows Run Dialog window

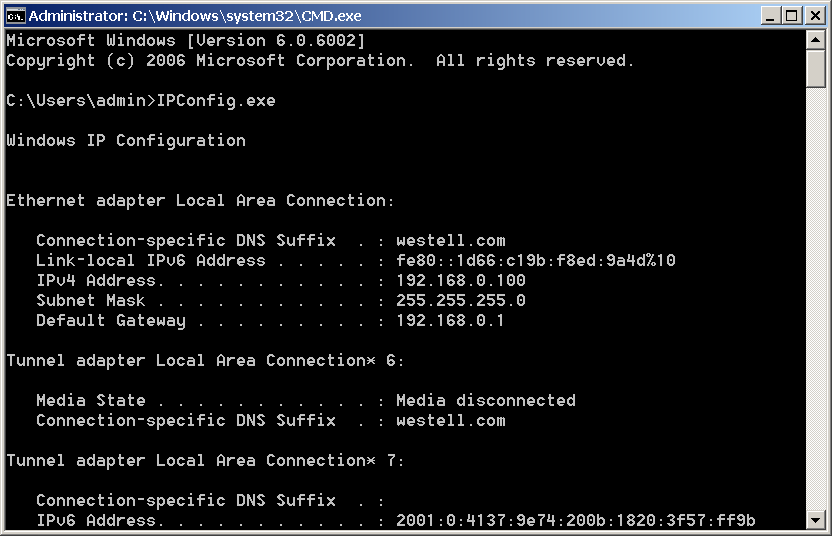
You will then be presented with a Command Prompt window like this…



#### Figure 3. A Windows Command prompt

If you are using a Mac, try searching for a similar feature for your OS. Here is an example: <https://www.google.com/search?q=How+to+open+a+command+shell+in+MAC> (opens external site)

Now, type in the command “IPConfig.exe” and hit the Enter key to see the IPConfig program run.



#### Figure 4. Results from running IpConfig.exe

IPConfig.exe is a Console application. These applications are not fancy, but they do allow you to accomplish useful tasks on a computer with the minimal fuss as to making your application look nice!

## Programming Basics

In a very general way, programs work with two things:

* ***Data:*** the information you want to work with, such as a person’s name and phone number.
* ***Operations:*** are the things you want to do with the data, such as printing data.

Of course, a program may also have other things as well, like *comments*, *namespaces*, or *directives*, but data and operations are the core of the program. As for those other things, here is a list of commonly ones:

* ***Comments:*** provide additional information to humans.
* ***Namespaces:*** provide an easy way to organize your code into named groups.
* ***Directives:*** provide additional information to the computer, but are not directly part of the program.
* ***Statements:*** the commands you add to a Python code file.
* A *statement* is one instruction to the computer
* Each of these statements will be made up of one or more keywords or symbols (sometimes called *tokens*)
* Since these can be more than one token per statement, you also need a way to indicate to the computer that you are done with a statement. In Python, you do so with a *carriage return*
* In Python you optionally use a semicolon(;) at the end of the statement (*But this is considered wrong by many of the python faithful!)*

x = 4 # This is one statement

y = 5 # this is another

z= x + y # and another as well

## Comments

In the previous code, we used comments to identify the purpose of each statement.

**#** This is a standard, *inline*, Python comment.

Commenting code is useful to add notes like these, but also useful when you want to see if disabling a set of statements solves a problem. Any code that follows a comment will not be processed.

If you *comment out* a section of code, and the problem disappears, then you know that the error is related to that set of statements.

print('test')

#print test opps this only works in 2.x

A comment only affects one line of code unless you use a "block" comment. Block, or *multi-line*, comments are not officially available in Python, but look like this in other C style languages.

**/\***

C Style languages use a slash-star and star-slash pair for a block comment.

Note to self: Both these statements are commented out for testing

int x = 5;

int y = 10;

**\*/**

Still, Block or *multi-line*, comments can be un-officially made in Python using 3 single quotes like this:

'''

Both these statements are commented out for testing

int x = 5;

int y = 10;

'''

## Case-Sensitivity

Python is a case-sensitive language, so you must be careful as you type.

x = 4 # This places the value of four into a variable called x

X = 13 # But, this places the value of four into a variable called X !

print(X) # displays the value 13 to the user

**PRINT**(X) # this command is not understood by Python

## Functions

Programmers have found that it a good practice to organize your code into groups. Statements are often grouped into *functions (*also known as *methods* or *sub-procedures)*. After you create a function, you can run its group of statements by *calling* the method.

#Create a function

def DemoFunction():

print("This is a statement in DemoMethod")

print("This is another statement in DemoMethod")

#End DemoFunction

#Call the function

DemoFunction()

### The print() Function

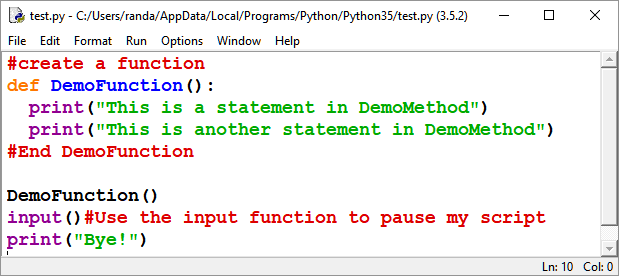
The print() function was created in Python to print out information to the command window. If you remember, IPConfig.exe wrote out its data to the command window for a human user to read. If you want to send output you would use the method to do so.

### The input() Function

The input() function gets data from the program’s user. It is also used to “pause” the program in the first chapter of the book, but we will see more practical uses for it later.

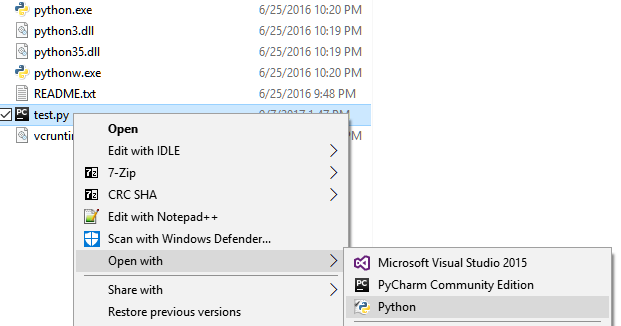
**Note**: In Python 2.x you use the raw\_input() function instead of the input() function to avoid and error!

The Input command can also be used to pause or stop the CMD (command) window from closing once the script finishes.



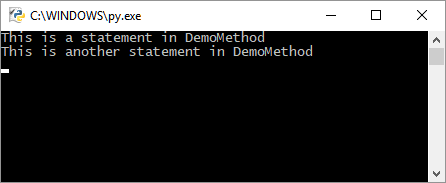
#### Figure 5. A Python script file opened in IDLE

You can see this behavior if you run a Python script from Windows Explorer.



#### Figure 6. Running a Python script file from Windows Explorer

When the script finishes, its programming statements it will close the Command window immediately, unless you add the input() function to force it to pause and wait for the user presses the Enter key.



#### Figure 7. The results of the script file while paused

## A Main() Method

Most applications run a “Main()” method as soon as a program or script is started. In Python this method is hidden, or implied. The entire body of the script file is considered the Main() method.

Within the script body, the implied Main() method, any code you type in will be processed one line after the other. If you call another method from the Main() method, it will jump to that method, run the statements inside of the called method, and return to the Main() method when it is done. The example below outlines the order in which your statements will be processed.

def DemoMethod(): # 3) jumps to here and run both statements…

print("This is a statement in DemoMethod")

print("This is another statement in DemoMethod")

#End DemoMethod

# 1) Start of Main

print("This is a statement in the invisible Main method")

DemoMethod() # 2) call the method DemoMethod()...

# 4) jumps back to here…

print("This is another statement in the invisible Main method")

# 5) End Main (the program ends! )

# Let’s review chapter 1 in you book now…

#### LAB 1-1: Create a Console Application

1. Create a new script using IDLE that prints out the text, “This is a test!” Do not use the **input**() function yet.
2. Run the script using Window Explorer. Note how the command window closes immediately after the script finishes.
3. Add the **input**() function to pause the script until the user presses the **Enter** key