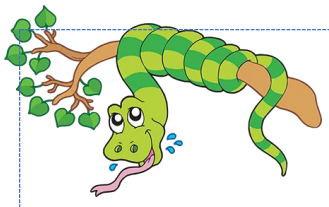


Python Basics

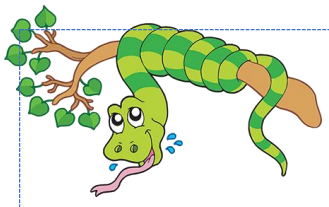
(Installation, Expressions)

Class viii

Lab 17

UNITED TRUST
ADD STRENGTHS TO THE WEAK

Powered By
bdeducation
Bangladesh Digital Education Research Limited
সময়ের পাঠশালা



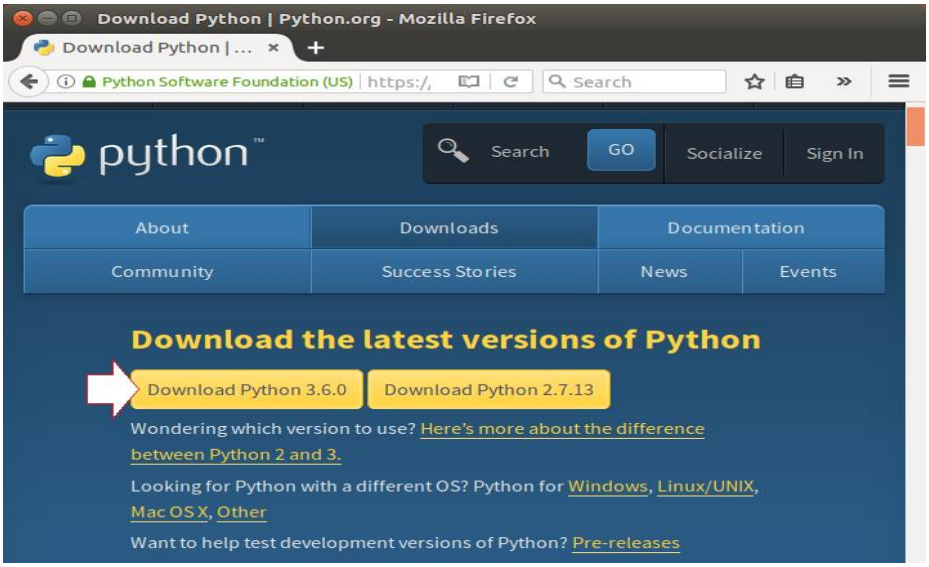
Lab Objectives:

- Installations
- Expressions
- Operator and Operations
- Basic Math

Installing python on windows operating system

The Python download requires about 30 Mb of disk space; keep it on your machine

- Step 1:
  - Go to : https://www.python.org/
  - Go to Downloads and click on Python 3.5+ version or latest one. The picture shows installing python 3.6.2



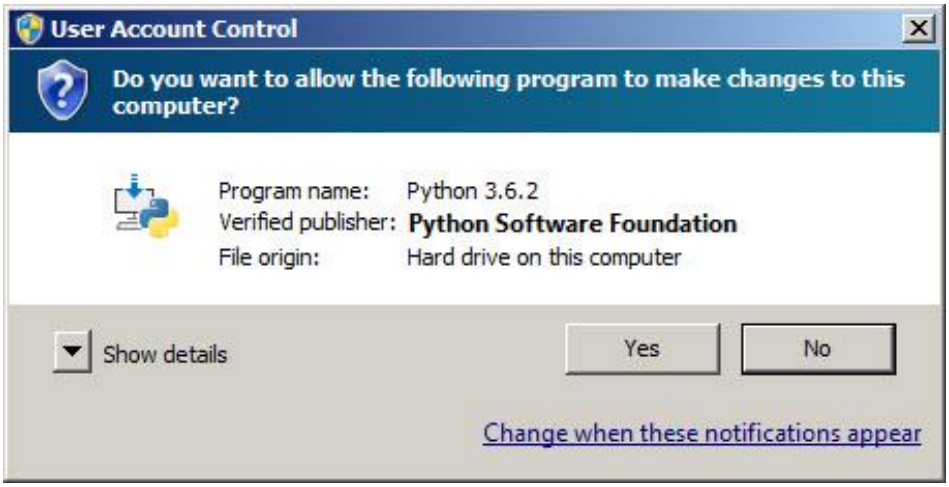
- Step two:
  - The file named **python-3.6.2.exe** should start downloading into your standard download folder. This file is about 30 Mb so it might take a while to download fully if you are on a slow internet connection (it took me about 10 seconds over a cable modem).
  - After finishing download click on downloaded file to install python on your machine
- Step three:
  - Double-click the icon labeling the file **python-3.6.2.exe**.

- An **Open File - Security Warning** pop-up window may

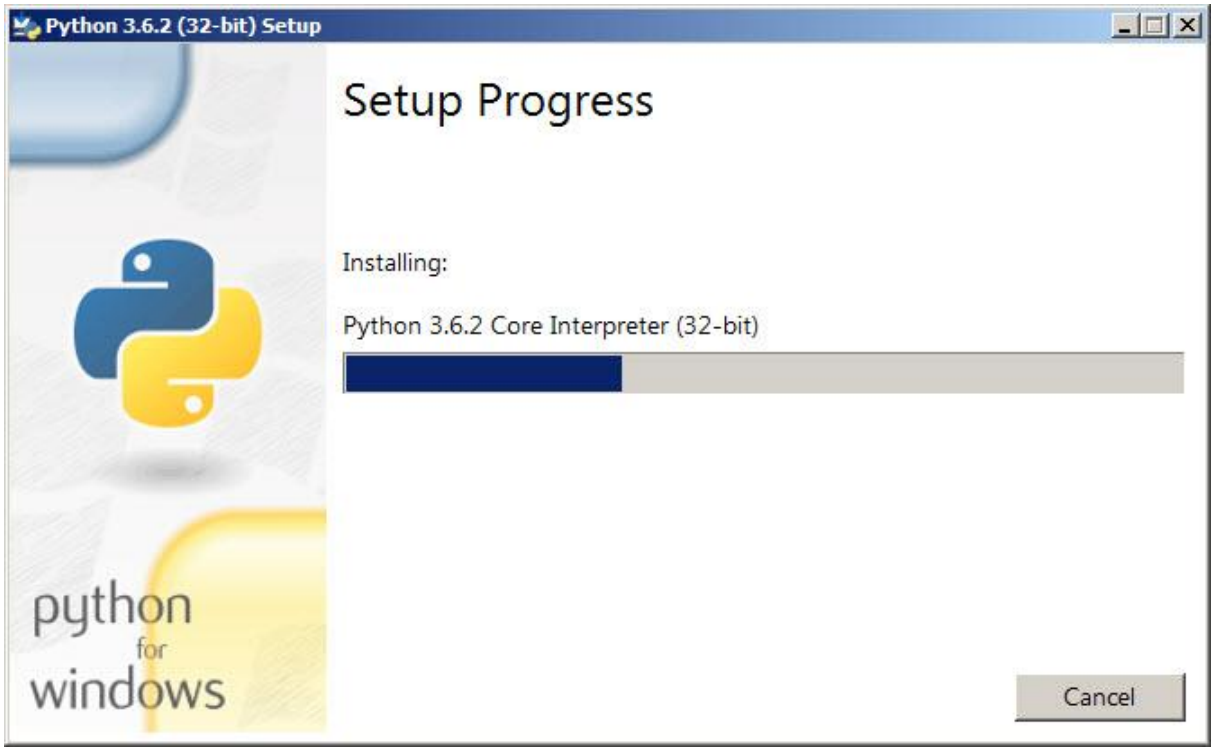
appear.

- Click **Run**.
- A **Python 3.6.2 (32-bit) Setup** pop-up window will appear.

- Ensure that the **Install launcher for all users (recommended)** and the **Add Python 3.6 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 will instead appear as **Upgrade Now** (and the checkboxes will not appear).
- Highlight the **Install Now** (or **Upgrade Now**) message, and then click it.
    - A **User Account Control** pop-up window will appear, posing the question **Do you want the allow the following program to make changes to this computer?**



- During installation, it will show the various components it is installing and move the progress bar towards completion. Soon, a new **Python 3.6.2 (32-bit) Setup** pop-up window will

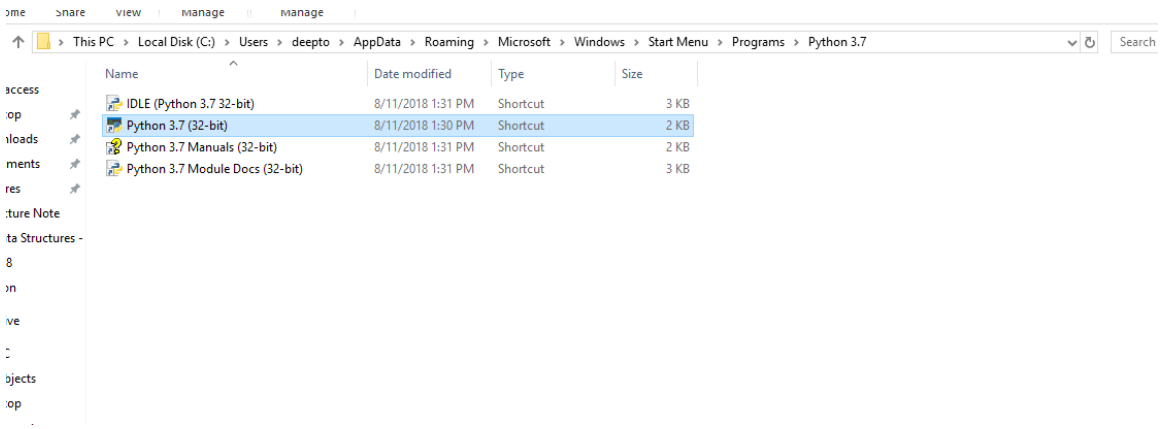


appear with a **Setup was successfully** message.

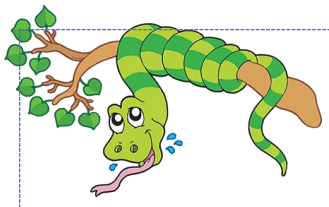
Verifying Installation

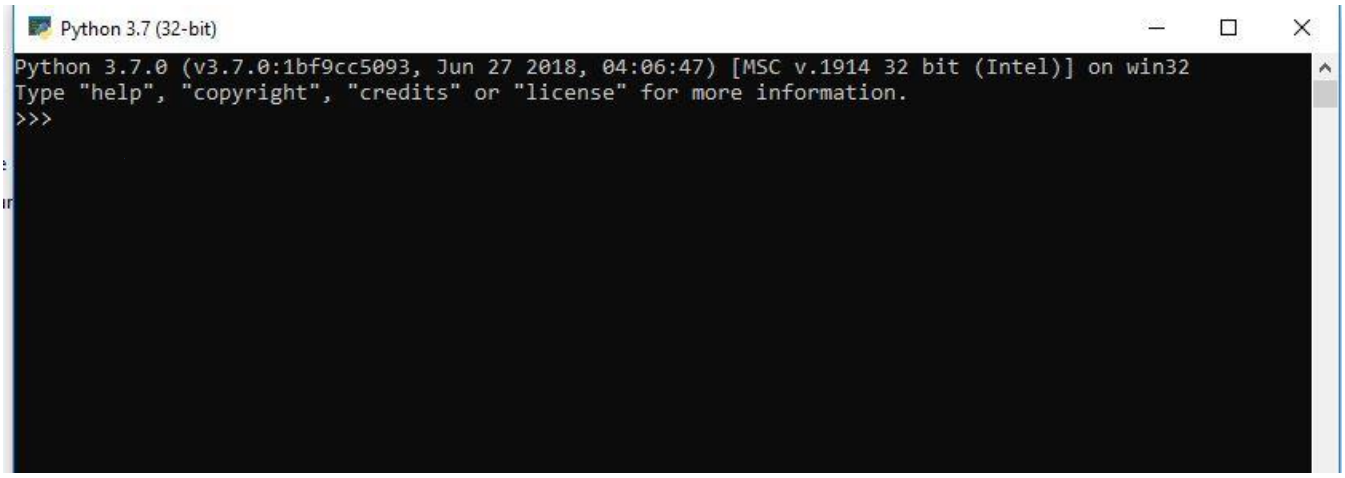
go to your pc where python is installed. Here my location is

C:\Users\deepto\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Python 3.7



Double click on Python 3.7(32-bit) or which version you have installed. You will find an interactive python shell for start coding. So let's jump into python.

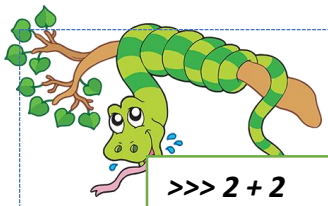




Entering Expressions into the Interactive Shell

You run the interactive shell by launching IDLE, which you installed with Python in the introduction. On Windows, open the Start menu, select All Programs ▸ Python 3.7, and then select IDLE (Python GUI). Or search by typing python from windows search bar. You'll find Python 3.7 IDLE. Click on that to start.

A window with the >>> prompt should appear; that's the interactive shell. Enter 2 + 2 at the prompt to have Python do some simple math.



>>> 2 + 2

4

The IDLE window should now show some text like this:

Python 3.3.2 (v3.3.2:d047928ae3f6, May 16 2013, 00:06:53) [MSC v.1600 64 bit (AMD64)] on win32

Type "copyright", "credits" or "license ()" for more information.

>>> 2 + 2

4

>>>

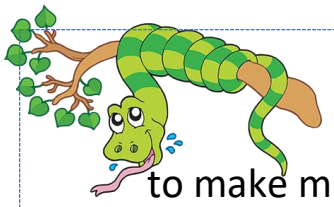
In Python, 2 + 2 is called an *expression*, which is the most basic kind of programming instruction in the language. Expressions consist of *values* (such as 2) and *operators* (such as +), and they can always *evaluate* (that is, reduce) down to a single value. That means you can use expressions anywhere in Python code that you could also use a value.

In the previous example, 2 + 2 is evaluated down to a single value, 4. A single value with no operators is also considered an expression, though it evaluates only to itself, as shown here:

```
>>> 2
2
```

Errors are Okay!

Programs will crash if they contain code the computer can't understand, which will cause Python to show an error message. An error message won't break your computer, though, so don't be afraid



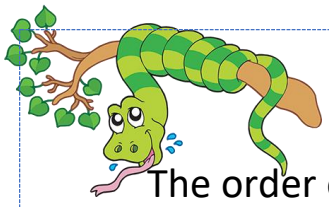
to make mistakes. A *crash* just means the program stopped running unexpectedly.

There are plenty of other operators you can use in Python expressions, too. For example, “Table 1” lists all the math operators in Python.

Table 1. Math Operators from Highest to Lowest Precedence

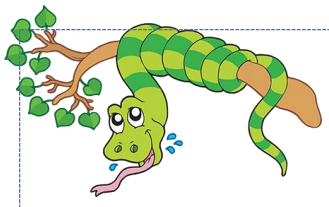
Operator	Operation	Example	Evaluates to...
**	Exponent	2 ** 3	8
%	Modulus/remainder	22 % 8	6
//	Integer division/floored quotient	22 // 8	2
/	Division	22 / 8	2.75
*	Multiplication	3 * 5	15
-	Subtraction	5 - 2	3
+	Addition	2 + 2	4

Table – 1



The order of operations (also called precedence) of Python math operators is similar to that of mathematics. The ** operator is evaluated first; the *, /, //, and %operators are evaluated next, from left to right; and the + and - operators are evaluated last (also from left to right). You can use parentheses to override the usual precedence if you need to. Enter the following expressions into the interactive shell:

```
>>> 2 + 3 * 6
20
>>> (2 + 3) * 6
30
>>> 48565878 * 578453
28093077826734
>>> 2 ** 8
256
>>> 23 / 7
3.2857142857142856
>>> 23 // 7
3
>>> 23 % 7
2
>>> 2    +         2
4
>>> (5 - 1) * ((7 + 1) / (3 - 1))
16.0
```



In each case, you as the programmer must enter the expression, but Python does the hard part of evaluating it down to a single value. Python will keep evaluating parts of the expression until it becomes a single value.

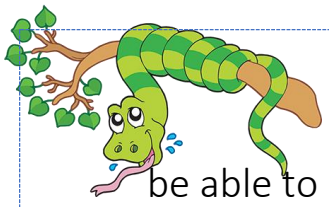
```
(5 - 1) * ((7 + 1) / (3 - 1))
        ↓
    4 * ((7 + 1) / (3 - 1))
        ↓
    4 * (( 8   ) / (3 - 1))
        ↓
    4 * (  8    /    2  )
        ↓
    4 * 4.0
        ↓
     16.0
```

These rules for putting operators and values together to form expressions are a fundamental part of Python as a programming language, just like the grammar rules that help us communicate. Here's an example:

This is a grammatically correct English sentence.

This grammatically is sentence not English correct a.

The second line is difficult to parse because it doesn't follow the rules of English. Similarly, if you type in a bad Python instruction, Python won't



be able to understand it and will display a SyntaxError error message, as shown here:

```
>>> 5 +
  File "<stdin>", line 1
    5 +
      ^
SyntaxError: invalid syntax
>>> 42 + 5 + * 2
  File "<stdin>", line 1
    42 + 5 + * 2
             ^
SyntaxError: invalid syntax
```

You can always test to see whether an instruction works by typing it into the interactive shell. Don't worry about breaking the computer: The worst thing that could happen is that Python responds with an error message. Professional software developers get error messages while writing code all the time.