Notes

Hello students and welcome to the notes section for this course.Rather than having huge sums of text which would be quite boring to read. I have taken care to keep the notes short, simple and up to the point. I have ignored the non-useful things and included important Python facts.

Notes & Section Summary for section 1:

What is Python:
Python is a widely used high-level, general-purpose, interpreted,dynamic programming language Python is founded by Guido van Rossum is a Dutch programmer who is best known as the author of the Python programming language.

Hello world program in Python:
The print function in Python allows us to print/display some text as the output.The simple line of code to display hello world is: print('Hello World').

You can use the general-purpose Python console to write Python code or you can use any IDE, which stands for Integrated Development Environment. PyCharm is the IDE which we have used in the course but you can use any IDE which supports Python.

Mathematical operations in Python.

We can perform mathematical operations like addition, subtraction, multiplication and division in Python. We can also perform some other operations like calculating exponent of any number. For example, to calculate square of number '4' we could type in 4**2 i.e 4 raised to power 2. Which is essentially calculating square of 4. In this manner we could calculate the exponent of any number. On similar terms, we could also calculate the square root of any number example 49**(1/2) which will give you the square root of number 49.

Strings in Python
Strings are among the most popular types in Python. We can create them simply by enclosing characters in quotes. Python treats single quotes the same as double quotes. Creating strings is as simple as assigning a value to a variable.

For example

```
var1 = 'Hello World!'
var2 = "Python Programming"
```

Accepting user input in Python 2
There are hardly any programs without any input.in most cases the input stems from the keyboard. For this purpose, Python provides the function input(). input has an optional parameter, which is the prompt string.
If the input function is called, the program flow will be stopped until the user has given an input and has ended the input with the return key. The text of the optional parameter, i.e. the prompt, will be printed on the screen.

The input of the user will be interpreted. If the user e.g. puts in an integer value, the input function returns this integer value. If the user on the other hand inputs a list,
the function will return a list.

Accepting user input in Python 3

If the input function is called, the program flow will be stopped until the user has given an input and has ended the input with the return key. The text of the optional parameter, i.e. the prompt, will be printed on the screen.

The input of the user will be returned as a string without any changes.If this raw input has to be transformed into another data type needed by the algorithm, we can use casting function.

Variables in Python

Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory.

Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory. Therefore, by assigning different data types to variables, you can store integers, decimals or characters in these variables. Python variables do not need explicit declaration to reserve memory space.

The declaration happens automatically when you assign a value to a variable. The equal sign (=) is used to assign values to variables.

The operand to the left of the = operator is the name of the variable and the operand to the right of the = operator is the value stored in the variable. Python has five standard data types −

- Numbers
- String
- List
- Tuple
- Dictionary
- **If statements in Python:**
- In order to write useful programs, we almost always need the ability to check conditions to change the behavior of the program accordingly. Conditional statements allow us to do so. The simplest form is the if statement, which has the general form:
- if BOOLEAN EXPRESSION:
  STATEMENTS
- **A few important things to note about if statements:**
- The colon (:) is significant and required. It separates the header of the compound statement from the body.
- The line after the colon must be indented. It is standard in Python to use four spaces for indenting.
  All lines indented the same amount after the colon will be executed whenever the BOOLEAN_EXPRESSION is true.
- The boolean expression after the if statement is called the condition. If it is true, then all the indented statements get executed. What happens if the condition is false, In a simple if statement like this,

nothing happens, and the program continues on to the next statement.

- **Elif statement in Python:**
- Sometimes there are more than two conditions and we need more than two branches. In such cases we use the elif statement.elif is an abbreviation of else if. Again, exactly one branch will be executed. There is no limit of the number of elif statements but only a single (and optional) final else statement is allowed and it must be the last branch in the statement:
-

```
if choice == 'a':
    print("You chose 'a'.")
elif choice == 'b':
    print("You chose 'b'.")
elif choice == 'c':
    print("You chose 'c'.")
else:
    print("Invalid choice.")
```

Each condition is checked in order. If the first is false, the next is checked, and so on. If one of them is true, the corresponding branch executes, and the statement ends.

Lists in Python:
Python has six built-in types of sequences, but the most common ones are lists and tuples.

- There are certain things you can do with all sequence types. These operations include indexing, slicing, adding, multiplying, and checking for membership. In addition, Python has built-in functions for finding the length of a sequence and for finding its largest and smallest elements. The list is a most versatile datatype available in Python which can be written as a list of comma-separated values (items) between square brackets. Important thing about a list is that items in a list need not be of the same type.

Creating a list is as simple as putting different comma-separated values between square brackets

- For example
- listone = ['physics', 'chemistry', 1997, 2000];
-
- **List functions in Python**
-

  append() : to append an item to the list.
  to append to a list named fruits, type in fruits.append("Banana")
- **function to calculate the length of the list:**
  print(len((fruits))
  This line of code will print out the length of the list named fruits
- **Insert function:**
  This function allows you to insert some item to the list, this function is similar to append but
  it allows you to insert item at a particular position.
  example:
  fruits.insert(1,"banana")
  This line of code places the item banana at position 1 in the list fruits.
- **The index function.**
  it returns the index value / position of particular item in the list.
  Example:
  print(fruits.index("Peach"))
  This line of code will return the index position of item Peach.
- **Range function in Python**
  The built-in range function in Python is very useful to generate sequences of
  numbers in the form of a list.
- The given end point is never part of the generated list;
- range(10) generates a list of 10 values, the legal indices for items of a sequence of length 10.
- It is possible to let the range start at another number, or to specify a different increment (even negative;
- Sometimes this is called the ‘step’):
- example:
- >>> range(1,10)
  [1, 2, 3, 4, 5, 6, 7, 8, 9]

- # You can use range() wherever you would use a list.

-
  ```
  a = range(1, 10)
  for i in a:
      print i
  ```