Python Programming

Data types

- Python offers following built-in core data types
  - Numbers
  - String
  - List
  - Tuple
  - Sets
  - Dictionary

Numbers

- Number data types are used to store numeric values in Python
- Numbers in Python have the following core data types:
- Integers
  - Integer
  - Boolean
- Floating Point numbers
- Complex Numbers

Strings

- A Python string is a sequence of characters and each character can be individually accessed using its index.
- Strings in Python are stored as individual characters in contiguous location, with two-way index for each location
- Strings are immutable
- The length of a string variable can be determined using len() function

Lists and Tuples

- A list in Python represents a list of comma separated values of any data type between square brackets.
- Lists are mutable
- a=[105,'Neha',79.5]
- Tuples are represented a s group of comma separated values of any data types within parantheses
- h=(7,8,9,'a','e','i')
- Tuples are immutable

Sets and Dictionary

- Sets are comma separated unordered values enclosed in curly brackets
- a={1,2,3,4}
- Dictionary is an unordered set of comma separated key:value pairs within{}
- d={'a':1,'e':2,'i':3}

Mutable and Immutable Types

- Mutability means that in the same memory address, new value can be stored as and when required.
- The types that do not support this type are immutable types.
- In Python, the following are immutable types: integer, float, boolean, string, tuple.
- The mutable types are those whose values can be changed in place.
- Mutable types in Python are lists, dictionaries and sets

Operators

- Arithmetic Operators
  - Unary Operators
  - Binary Operators
    - Addition operator +
    - Subtraction operator –
    - Multiplication operator *
    - Division Operator /
    - Floor division operator //
    - Modulus operator %
    - Exponentiation operator **

Operators

- Augmented Assignment Operators
  - +=, -=,*=,/=,//=,**=,%=
- Relational Operators
  - <, <=, >, >= ,==, !=
- Identity Operators
  - is, is not
- Logical Operators
  - And,Or,Not

Statements in Python

- Python statements are of 3 types
  - Empty Statement
    - In Python an empty statement is pass statement.
    - It takes the form: pass
    - Whenever Python encounters a pass statement , Python does nothing and moves to the next statement in the flow of control
  - Simple Statement
    - Any single executable statement is a simple statement .
  - Compound Statement
    - A compound statement represents a group of statements executed as a unit.

Statement flow control

- In a program, statement may be executed sequentially, selectively or iteratively
  - Sequence
    - Statements are being executed sequentially
  - Selection
    - Execution of statements depend upon a condition.
  - Iteration(Loop)
    - Repetition of a set of statements depending upon a condition test.

if statement

- if condition:

  statement(s)

- if condition:

  statement(s)

  else:

  statements(s)

if statement(contd…)

- if condition:

  statement(s)

  elif condition:

  statement(s)

  …..

  else:

  statements(s)

Nested if statement

- A nested if is an if that has another if in its if's body or in elif's body or in its else's body

range() function

- The range() function of Python generates a list which is a special sequence type.
- General form of range() function is
  - range(<lowerlimit>,<upperlimit>)
  - This will produce a list having values starting from lowerlimit,lowerlimit+1,……upperlimit-1
  - Default step value is 1
  - range(<lowerlimit>,<upperlimit>,<step value>)
  - This will produce a list having values starting from lowerlimit,
    lowerlimit+<stepvalue>,……<=upperlimit-1

Membership operators

- in
  - Checks whether a value is in a sequence
  - Returns True if the value is present
  - Otherwise false
- not in
  - Checks whether a value is not in a sequence
  - Returns True if the value is not present
  - Otherwise false

Iteration / Looping statements

- Python provides 2 kinds of loops.
  - Counting loops
    - Loops that repeat a certain number of times
    - for loop is a counting loop
  - Conditional loops
    - Loops that repeat as long as some condition is true
    - while loop is a conditional loop

for Loop

- The general form of for loop is

  for <variable> in <sequence>:

  statements

- Sequence is a collection of items

  eg: list of numbers

  string of characters

  range

while Loop

- The general form of while loop is

```
while <condition>:
        statements
```

eg: printing numbers from 1 to 10

```
i=1
while i<=10:
        print(i)
        i+=1
```

Jump statements

- Python offers 2 jump statements to be used within loops .
  - break
    - break statement terminates the loop it lies within
    - General form is : break
  - continue
    - continue statement forces the next iteration of the loop to take place skipping any code in between.
    - General form is : continue

Loop else statement

- The else clause of a Python loop executes when the loop terminates normally.
- Syntax is :

```
for <variable> in <sequence>:
    statements
    …..
else:
    statement(s)
```

Loop else statement

- Syntax is :

```
while <test condition>:
    statements
    …..
else:
    statement(s)
```

Nested loops

- A loop given inside other loop is called a nested loop.

Thank you