PAGE NO.
DATE:

Python notes

* python - It is a high-level, Interpreted & object-oriented programming Language.

* application of python -
① Google's App Engine web development framework uses python as an application language.
② Maya, a powerful integrated 3D modeling & Animation System.
③ Linux weekly news, published by using a web application.

* features of python -
① Simple & Easy-to learn.
② Interpreted & Interactive.
③ object-oriented Language
④ platform independent
⑤ Dynamic, Scalable, Extendable.
⑥ free & open source.

* Application of python -

① Web Application
② GUI based Desktop Application
③ Scientific & Numeric Application
④ Software Development.
⑤ Business Application.
⑥ 3D CAD Application.
⑦ Education purpose

PAGE NO.
DATE:

* python program contains following sections.

① Documentation section includes the comments that specify the purpose of the program.
② import section is used includes different build in or user defined modules.
③ Global declaration section - It is used to declare global variable.
④ class section describes the information about the user defined classes in the program.
⑤ Sub program section includes use defined function.
⑥ Pay ground section is the main section of program

* standard data types -

① ~~Number~~ Boolean -
ⓐ The simplest built-in type in python is the bool data type, it represents the two value, True or False.
ⓑ Ex: >>> size = 1
>>> size < 0
False

② None Data type -
ⓐ python defines a special variable None denoting a "null object".
ⓑ Ex: answer = none
if answer is none:
quit = True
elif answer == 'quit':
quit = True

PAGE NO.
DATE:

```
else :
    quite = False
```

③ Number Data type -

ⓐ Number data type stores numeric values.

ⓑ Ex:

```
>>> a = 10
>>> a
10
>>> type(a)
<class 'int'>
```

④ string -

ⓐ string is a collection of group of characters.

ⓑ Ex:

```
>>> S1 = "Hello"
>>> S2 = "Hi"
>>> S1
'Hello'
```

⑤ List data type -

ⓐ Lists are the most varsatile of python's compound data types.

ⓑ Ex:

```
>>> S = [10, 20, 30]
>>> S
[10, 20, 30]
```

⑥ Tuple -

ⓐ It is an ordered sequence of items same as List.

PAGE NO.
DATE:

⑥ Ex.

```
T = (70, 2, 50, "tybcs")
>>> T
(70, 2.50, "tybcs")
```

* Variable - A variable is like a container that stores values that we can access or change.

* constants - A constant is a type of variable whose value cannot be changed during the execution of program.

* Literals - A literal refers to the fixed value.

* identifiers - A Identifier is a name given to a function class, variable, module or other object that is used in python program.

* Reserved words - python keywords are reserved words with that have special meaning & functions

* Indentation - It helps to convey a better structure of a program to the readers

* Comments -
① Single Line comment (#)
② Multiple Line comment (''')

PAGE NO.
DATE:

* Dry run-
It is the process of a programmer manually working through their code to trace the value of variables.

* python operator precedence-
when an expression has two or more operators, we need to identify the correct sequence to evalute these operators.

* Associativity -
when two operators have the same precedence associativity helps to determine which the order of operations.

* Type Conversion -
① the process of converting the value of one data type to another data type is called ~~dat~~ type conversion.
② Implicit Type conversion- In this python automatically converts one data type to another data type.
③ Explicit Type conversion - In this the user convert the data type of an object to required data type.

* Conditional statement -
① if statement - Executes a statement if a condition is True.

PAGE NO.
DATE:

Ex. i = 10
```
if (i<15):
    print ("i is less than 15")
    print("this statement is not in if")
```

② if else statement - It evaluates test expression & will execute the body of if only when the test condition is True. If the condition is False, the body of else is executed.

Ex. i = 20
```
if (i<15):
    print("i is less than 15")
else:
    print("i is greater than 15")
```

③ if elif else statement - the elif is short for else if. It allow us to check for multiple conditions.

Ex. i = 20
```
if (i==10):
    print("i is 10")
elif (i==15):
    print("i is 15")
elif (i==20):
    print("i is 20")
else:
    print("i is not present")
```

PAGE NO.
DATE:

④ Nested if statement - when a programmer writes one if statement inside another if statement then it is called nested if statement.

Ex.
```
a = 30
b = 20
c = 10
if (a>b);
    if (a>c):
        print("a is greater")
else:
    print ("a is less than")
print ("End of Nested if")
```

* Looping -

① while loop - It repeatedly executes a target statement as long as a given condition is true.

② for loop - It is used to iterate the statement or a part of the program several times.

* strings -

① A python string is a sequence of characters.

② python strings are immutable sequences of Unicode points.

③ Ex.
```
my_string = 'Hello'
print (my_string)
```
o/p - Hello

PAGE NO.
DATE:

String Representation -

(a) In python, strings are stored as individual characters in a contiguous memory location.

(b) We can access the individual characters in string by using the string variable & index.

(c) The benefit of using string is that it can be accessed from both the direction (forward & backward)

* Updating / Manipulating String -

(1) The string can be update by (re) assigning a variable to another string.

(2) The new value can be related to its previous value or to a completely diffrent string altogether.

(3) Ex. msg1 = 'Hello world!'
print ("Updated string: ", msg1[:6] + 'python')

* Raw String & Unicode String -

(1) python raw string is created by prefixing a string literal with 'r'.

(2) Raw strings do not treat the backslash as a special character at all.

(3) Ex. print (r'python :\\ Programming')

(4) python strings are stored internally as 8-bit ASCII, while Unicode strings are stored as 16-bit Unicode.

(5) Ex. print (u'Hello, World!')

* List -

① A python List is a mutable sequence of data values called items or elements.

② A List is a collection of items or elements

③ Creating element in a list:

@ A list in python is created by placing all the items inside a square bracket [], Seperated by commas.

Syntax -

<list_name> = [value 1, value 2, ..., value N]

Ex. Emp = [20, "Amar", 'M', 50]

* Accessing Elements of a List -

① List Index

② Negative indexing

* Updating List -

① append() & - The append() method adds an element to the end of a list. we can insert a single item in the list data time with the append().

② extend() * - This method extends a list by appending items. we can add several items using extend() method.

③ insert() - we can insert one single item at a desired location by using the method insert()

④ + operator - the concatenation in python program is used to combine the elements or two lists.

⑤ * operator - The * operator repeats a list for the given number of times.

* Deleting List -

① pop() method - It is used to remove a particular item / element from the given index in the list.

② Using del keyword - we can delete one or more items from a list using del keyword.

③ remove() method - It is used to remove a particular element from the list.

* functions - A function is a block of organized, reusable code that is used to perform a single related action / operation.

* flow of Execution -

① when we are working with functions it is really important to know the order in which statements are executed called as flow of execution.

② function definitions do not alter the flow of execution of the program, but remember that statements inside the function are not execute

until the function is called.

③ function calls like a detour in the flow of execution.

④ Ex.

```
def my_function():
    print("Hello from a function")
my_function()
```

Local variable	Global variable
① local variables are declared inside a function.	Global variable are declared outside any function.
② Accessed only by the statements, inside a function in which they are declared.	Accessed by any statement in the entire program.
③ Local variables are alive only for a function.	Global variable are alive till the end of the program.

* Void function -
  In python, it is possible to compose a function without a return statement.

* Anonymous function - this are the function that are not bond to name.

* Recursion - It is method of programming or Coding a problem, in which a function calls itself one or more times in its body.

* functional programming Tools.

① filter() - The filter() function in python takes in a function & a list as arguments.

② map() - the map() function in python takes in a function & a list as argument.

③ reduce() - The function is called with lambda function & a list & a new reduced result is returned.

* Tuple -
A Tuple is a collection of object which ordered & immutable.

* Dictionaries -
the dictionary data structure is used to store key value pairs indexed by keys.

* Sets -
A set is an unordered collection of objects.

* Basic sets operations -

① Union of sets -
the union operation on two sets produces a new set containing all the distinct element from both the sets.

Page :
Date : / /

③ Intersection of sets -
The intersection operation on two sets produces a new set containing only the common elements from both the sets.

④ Difference of sets -
The difference operation on two sets produces a new set containing only the elements from the first set & none from the second set.

* modules -
modules are primarily (.py) files which contain python programming code defining functions, class, variables, etc.

* importing module -
① The import statement is used to imports a specific module by using its name.
② We can import the definitions inside a module to another module.
Import the definition inside a module.

```
import P1
print (P1.add(10,20))
print (P1.sub(20,10))
```

* importing objects from module -
① From x import a.
② From x import a,b,c.
③ From x import *.

* Aliasing modules-
① It is possible to modify the names of module & their functions within python by using the 'as' keyword.
② Syntax - import module as another_name.

* predefined modules-

① Numeric & mathmatical modules-
(a) math & cmath modules-
① python provide two mathmatical models math & cmath.
(b) Decimal modules-
Decimal numbers are just the floating-point numbers with fixed decimal points.
(c) Fraction modules-
It is a number which represent a whole number being divided into multiple parts.
(d) Random module- It provides functions to perform various types of operations.

② functional programming modules-

(a) itertools module.
(b) functools module.
(c) operator module.
(d) Time module.

* packages -
① A package is a collection of python modules i.e a package is a directory of python modules containing an additional __init__.py file.
② Creating a package is quite easy, since it makes use of the operating system's inherent hierarchical file structure.

* standard pacakages -
① NumPy & SciPy are the standard packages used by python.

Built-in packages -
① NumPy - It is the functional package for Scientific Computing with python.
② SciPy - It is a library that uses NumPy for more mathematical functions.
③ Matplotlib - It is a plotting library for 2D graphic in python.
④ pandas - It is an open source python library providing high-performance data manipulation & analysis tool.

* file -
① file is a named location on disk to store related information.
② To Create a new file in python, use the open() method, with one of the following parameters - "x" : ~~Create~~ Create, "a" = append.
"w" : write.

* operations on file -

① open() function.
② close() function.
③ Reading & writing files.

* Directories -
① A directory or folder is nothing more than a location on a disk used for storing information about files.
Syntax - os.mkdir("newdir")

* Regular Expression -

① A regular Expression or RegEx or REs is a sequence of characters that forms a search pattern.
② It can be used to check if a string contain a specified search pattern.
③ The module 're' provides the support to use RegEx in python program.
④ To avoid any confusion while dealing with regular expressions, we would use Raw strings as r'expression.
⑤ Types of regular Expression.
(a) Basic regular expression.
(b) Extended regular expression.

* Exception handling - An exception is also called as runtime error that can halt the execution of the program.