**XIST – Xanthron Industrial Software Training**

**Module 3**

**Revision**

• Using list as stack

• Strings in python

• Python does not support character type

• Strings are created by enclosing a sequence of characters within a pair of single or double or triple quotes

• String functions like len(), str()

• Python considers string as an object and there are so many methods like str.capitalize(), str.split(), str.center(), str.coount(), str.find(), str.lstrip(), str.lower, str.upper etc.

• A tuple is a sequence of immutable(which means you cannot update or change the values of tuple elements)

• Tuple is faster than list

**Python Dictionary**

Dictionary is an unordered collection of items. Dictionaries map **keys** to **values** and these key- value pairs provide a useful way to store data in Python. We can access value using these keys.

Dictionaries are constructed with curly braces,{}.

For example:

emp = {'name': 'MA Hadi', 'age': 33}

Here name, age are keys and MA Hadi and 33 are it’s corresponding values and keys are made up of strings.

We can access the data MA Hadi and 33 by calling

print (emp[‘name’])

print emp[‘age’]

print (emp.**keys()**) will display all keys as

['age', 'name']

print (emp.**values()**) will display all values as

[33, 'MA Hadi']

emp.**items()** returns items in a list format of (key, value) tuple pairs as

[('age', 33), ('name', 'MA Hadi')].

To **Add** a key-value pair to a dictionary

**emp['sex']='M'**

The new dictionary looks like

{'age': 33, 'name': 'MA Hadi', **'sex': 'M'**}

Same way we can **change** already existing value in the dictionary

emp['age'] = **34**

The new dictionary looks like

{'age': **34**, 'name': 'MA Hadi', 'sex': 'M'}

To **remove** a key-value pair from a dictionary, we’ll use the following

**del** emp[‘age’]

The new dictionary looks like

{'name': 'Madhu', 'sex': 'M'}

**emp.clear()** function will remove all the items of emp dictionary**.**

**del emp** will delete the whole dictionary.

Because dictionaries are mutable data types, they can be added to, modified, and have items removed and cleared.

**Local and Global variable**

Global variables are the one that are defined and declared outside a function and we need to use them inside a function. If a variable with same name is defined inside the scope of function as well then it will print the value given inside the function only and not the global value.

For Example.

# Global scope

n = "I love India"

def myName():

n = "My name is MA Hadi"

print n

myName()

print n

**Output :**

My name is MA Hadi

I love India

Here the first print statement prints the value of local variable n by calling the function myName() and the second one print the value of global variable n.

If we declare the variable n is global inside the function then the output will be different.

n = "I love India"

def myName():

**global n**

n = "My name is MA Hadi"

print n

myName()

print n

**Output:**

My name is MA Hadi

My name is MA Hadi

One of the most distinctive features of Python is its use of

indentation to mark blocks of code. Programs get structured through indentation,this means that code blocks are defined by their indentation.

Consider the if-statement from our simple password-checking program:

if pwd == 'apple':

print('Logging on ...')

else:

print('Incorrect password.')

print('All done!')

To indicate a block of code in Python, you must indent each line of the block by the same amount. The two blocks of code in our example if-statement are both indented four spaces, which is a typical amount of indentation for Python.

In most other programming languages, indentation is used only to help make the code look pretty. But in Python, it is required for indicating what block of code a statement belongs to.

For instance, the final print('All done!') is *not* indented, and so is *not* part of the

else-block.

For **commenting** a line use **#**

**Conditional Statements**

Please analyze the following python program

x,y = 100,900

if (x<y):

l = "y is greater than x"

elif (x==y):

l = "y is equal to x"

else:

l = "x is greater than y"

print l

The first line is for assigning values to variables x and y

Please see the syntax of if ...elif ... else statement and the

indentation of block of code.

We can use if statement in another way in python.

x,y = 100,900

l = "x is less than y" if (x<y) else "x is greater than or equal to y"

print l

It is more readable.

**Loops**

In python there are two statements to create loops

**1. while**

x = 0

while (x<5):

print x

x = x + 1

**Output:**

01234

**2. for**

Before going to discuss the for loop we have to learn a new **python function RANGE**

**syntax : range(begin,end,step)**

• begin: Starting number of the sequence.

• end: Generate numbers up to, but not including this number.

• step: Difference between each number in the sequence.

Examples:

a.range(5)

**Output:**

[0, 1, 2, 3, 4]

b. range(5,10)

O**utput:**

[5, 6, 7, 8, 9]

c. range (5,10,2)

**Output:**

[5, 7, 9]

d. range (10,5,-1)

**Output:**

[10, 9, 8, 7, 6]

for loops in python differ from the for loops of other languages.

Syntax of for loop:

for variable in sequence:

Statement1

Statement2

...

Statementn

else:

Else-Statement1

Else-Statement2

...

Else-Statementm

Examples:

1.

for i in range(5):

print (i)

**Output:**

01234

2.

for i in range(5,10):

print(i)

**Output:**

56789

3.

for i in range(4,10,2):

print(i)

**Output:**

468

4.

for i in range(0, -10, -2)

print(i)

**Output:**

0-2-4-6-8

5. Loops over a collection(list)

**days = [“Mon”,”Tue”,”Wed”,”Thu”,”Fri”,”Sat”,”Sun”]**

for i in days:

print (i)

**Output:**

Mon Tue Wed Thu Fri Sat Sun

**Break and continue statements**

**break** is used to exit from a specific loop under some conditions and

**continue** is used to continue the loop with out executing the code after the continue statement under some conditions.

for x in range(5,10):

if (x==7): break

print x

**Output:**

5,6

for x in range(5,10):

if (x%2==0): continue

print x

**Output:**

579

**skipped even numbers 6 and 8**

**Using enumerate function**

days = [“Mon”,”Tue”,”Wed”,”Thu”,”Fri”,”Sat”,”Sun”]

for i,d in enumerate(days):

print i,d

**Output:**

0 Mon

1 Tue

2 Wed

3 Thu

4 Fri

5 Sat

6 Sun

**Here we get each member of the list(d) and the it’s index value(i).**

**Functions in Python**

As you know functions are blocks of codes and are reusable.

Structure of functions in python

def function name (list of arguments):

statement1

statement2

.....

return

Example :

1. Function without any arguments

def func():

print “Hello”

2. Function with arguments

def func1(arg1,arg2):

print arg1,arg2

3. Function that returns a value

def cube(x):

return x\*x\*x

4. Function with default value for an argument

def power(num,x=1):

result = 1

for i in range(x):

result = result \* num

return result

5. Function with variable number of arguments

def multi\_add(\*args):

result = 0

for x in args:

result = result + x

return result

We can call this function with n number of arguments.

multi\_add(12,3,23,45)

**Excersises**

1.Write a program to calculate the sum of n numbers.

2.Write a program to find the biggest number from given three numbers.

3. Write a Python program which accepts the user's first and last name and print them in reverse order with a space between them

4.Write a Python program to test whether a number is within 100 or 1000 or 2000

5.Write a Python program which accepts the radius of a circle from the user and compute the area.

6. Write a Python program to calculate the sum of three given numbers, if the values are equal then return thrice of their sum

7.Write a Python program to get the volume of a sphere with radius 6.

8.Write a Python program that accepts an integer (n) and computes the value of n+nn+nnn.

9.Write a program which will find all such numbers which are divisible by 7 but are not a multiple of 5, between 2000 and 3200 (both included).

10.Write a program which can compute the factorial of a given numbers.

11. X = [[1,17,13],[4 ,15,26],[3 ,7,9]] and Y = [[25,81,12],[16,9,3], [14,51,8]]

Write a program to add these two matrices.

12. How to convert a dictionary to list in python. Explain with examples.

13. A dictionary named student exists.

Explain the following

1. student.clear()

2. del student