**XIST – Xanthron Industrial Software Training**

**Module 2**

**Revision**

**• We can execute python programs 3 ways**

**• Declaration of variables is not required in Python**

**• A valid variable name is a non-empty sequence of characters of any length**

**• +, -, \*, /, %, \*\*, // are arithmetic operators**

**• ==, !=, <>, >, <, <= ,>= are logical operators**

**• int(), long(), float(), complex() are conversion functions**

**• ceil(),f loor(), round(), sqrt(), min(), max(), cmp(), pow(), modf() are some mathematical functions**

**• A list is a data structure that holds an ordered collection of items i.e. you can store a sequence of items in a list.**

**Example:**

**shop\_list = [‘tomato’, ’sugar’, ’rice’, ’chicken’]**

**• The empty list is written as []**

**• append(), extend(), insert(), remove(), pop(), sort(), reverse(), count(), index() are some methods of list data structure.**

**Using List as Stacks**

**Using append and pop methods we can create stacks using lists.**

**For Example:**

**st = [10,20,30,40]**

**st.append(50)**

**st.append(60)**

**print(st) # output : [10,20,30,40,50,60]**

**st.pop()**

**print(st) # output : [10,20,30,40,50]**

**st.pop()**

**print(st) # output : [10,20,30,40]**

**Using Lists as Queues**

**It is also possible to use a list as a queue, where the first element added is the first element retrieved (“first-in, first-out”); however, lists are not efficient for this purpose. How to use lists as queues, we will discuss later.**

**String**

**In python strings are lists of characters. Python does not support character type; these are treated as strings of length one. Strings are created by enclosing a sequence of characters within a pair of single or double quotes.**

**You can specify multi-line strings using**

**triple quotes - (""" or '''). You can use single quotes and double quotes freely within the triple quotes.**

**var1 = 'Hello World!'**

**var2 = "Python Programming"**

**print (“var1[0]:” , var1[0])**

**print ("var2[1:5]: ", var2[1:5])**

**Output**

**var1[0]: H**

**var2[1:5]: ytho**

**String functions**

**1. len()**

**To find length of a string**

**Example:**

**name = ’Xanthron e-solutions’**

**print len(name)**

**index: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19**

**name: x a n t h r o n e - s o l u t i o n s**

**-ve index -19 -18 -17 -16 -15 -14 -13 -12 -11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1**

**name[4] is ‘h’, name[-19] is ‘s’ , name[0] is ‘X’ and name[0:3] is 'Xan'**

**2. str()**

**To convert a number to string**

**x = 100**

**y = 1**

**print (x+y)**

**print (str(x)+str(y))**

**Output: 101**

**1001**

**3. str.capitalize()**

**Return a copy of the string with its first character capitalized and the rest lowercased.**

**X = “ first character Capitalized”**

**print (X.capitalize())**

**Output: First character capitalized**

**4. str.center(width[, fillchar])**

**Return centered in a string of length width. Padding is done using the specified fillchar (default is an ASCII space). The original string is returned if width is less than or equal to len(s).**

**str1 = ‘Xanthron e-solutions’**

**print (str1.center(30,’-’)\_**

**print (str1.center(30) )# default is space**

**Output:**

**------Xanthron e-olutions------**

**Xanthron e-olutions**

**5. str.count(sub[, start[, end]])**

**Return the number of non-overlapping occurrences of substring sub in the range [start, end]. Start and end are optional.**

**X = “first character Capitalized”**

**print (X.count('a'))**

**print (X.count('a',0,17))**

**print (X.count('first’))**

**Output:**

**4**

**2**

**1**

**6. str.endswith(suffix[, start[, end]])**

**Return True if the string ends with the specified suffix, otherwise return False. Start and end are optional.**

**X = “first character Capitalized”**

**print (X.endwith(‘q’))**

**print (X.endwith(‘d’))**

**print (X.endswith(‘zed’))**

**print (X.endswith(‘t’,0,5))**

**Output:**

**False**

**True**

**True**

**True**

**7. str.find(*sub*[,*start*[,*end*]])**

**Return the lowest index in the string where substring is found within the slices[start:end]. Return-1 if substring is not found.**

**X = “first character Capitalized”**

**print (X.find(‘Cap’))**

**print (X.find(‘c’))**

**print (X.find(‘z’,1,10))**

**Output:**

**16**

**6**

**-1**

**8. str.isalnum()**

**Return true if all characters in the string are alphanumeric and false if there is at least one character is other character like space, /,? etc.**

**X = “first character Capitalized”**

**print (X.isalnum())**

**X = “firstcharacterCapitalized”**

**print (X.isalnum())**

**X = “firstcharacterCapitalized?”**

**print (X.isalnum())**

**Output:**

**False**

**True**

**False**

**9. str.isalpha()**

**Return true if all characters in the string are alphabetic and false if there is at least one non alphabetic character.**

**X = “first character Capitalized”**

**Print(X.isalpha()) # returns false due to non alphabetic character SPACE**

**X = “firstcharacterCapitalized”**

**print (X.isalpha()) #returns True**

**10. str.isdigit()**

**Return true if all characters in the string are digits and false if there is at least one non digit character.**

**11. str.islower()**

**Return true if all cased characters in the string are lowercase.**

**12. str.isnumeric()**

**Return true if all characters in the string are numeric characters, and there is at least one character, false otherwise.**

**13. str.isupper()**

**Return true if all cased characters in the string are uppercase and there is at least one cased character, false otherwise.**

**14. str.ljust(width[,fillchar])**

**Return the string left justified in a string of length width. Padding is done using the specified fillchar(default is an ASCII space). The original string is returned if width is less than or equal tolen(s).**

**15. str.lower()**

**Return a copy of the string with all the cased characters converted to lowercase.**

**16. str.lstrip([chars])**

**Return a copy of the string with leading characters removed.**

**x=" first character capitalized"**

**y="\*\*\*first character capitalized"**

**z="first character capitalized"**

**print (x.lstrip()) # default space**

**print (y.lstrip(‘\*’))**

**print (z.lstrip(‘fir’))**

**Output**

**'first character capitalized'**

**'first character capitalized'**

**'st character capitalized'**

**17. str.replace(old,new[,count])**

**Return a copy of the string with all occurrences of substring old replaced by new. If the optional argument count is given, only the first count occurrences are replaced.**

**18. str.rjust(width[,fillchar])**

**Return the string right justified in a string of length width. Padding is done using the specified fillchar(default is an ASCII space). The original string is returned if width is less than or equal to len(s).**

**19. str.rstrip([chars])**

**Return a copy of the string with trailing characters in the right side of string, removed.**

**20 .str.swapcase()**

**Return a copy of the string with uppercase characters converted to lowercase and vice versa.**

**21. str.upper()**

**Return a copy of the string with all the cased characters converted to uppercase.**

**Tuple**

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

**• Tuples are sequences, similar to lists.**

**• The difference between the tuple and list is that we cannot change the elements of a tuple once it is assigned whereas in a list, elements can be changed.**

**• If you have data that doesn't change, implementing it as tuple will guarantee that it remains write-protected.**

**• Tuple is faster than list.**

**• Tuples use parentheses, whereas lists use square brackets.**

**• Removing individual tuple elements is not possible.**

**• To delete entire tuple use del statement. Eg: del x**

**• To write a tuple containing a single value you have to include a comma, even though there is only one value Eg: t1=(5,)**

**• Nested tuple is possible.**

**For Example:**

**ntuple = ("mouse", [8, 4, 6], (1, 2, 3))**

**t1 = (1,2,3,4,5,6)**

**t2 = (‘malu’,’chandu’)**

**print (t2[1])**

**print (t1[2:4])**

**print (t1+t2)**

**Output:**

**‘chandu’**

**(3, 4)**

**(1,2,3,4,5,6,’malu’,’chandu’)**

**Tuple functions**

**1. len(): to find out total number of elements of a tuple. Eg: len(x)**

**2. cmp(tuple1,tuple2): compares two tuples**

**3. max() and min(): to find maximum and minimum values**

**4. tuple(x): Converts a list into tuple.**

**Excersises**

**1. tuple = ("university", [5, 4, 6], (3, 2, 1))**

**print(n\_tuple[0][3])**

**print(tuple[2][3])**

**What will be the output?**

**2. tuple = ("university", [5, 4, 6], (3, 2, 1))**

**tuple[1][1]=20**

**tuple[2][2]=0**

**print (tuple)**

**What will be the result and why?**

**3. s='University of Calicut'**

**print(s.swapcase())**

**print( s.swapcase().swapcase())**

**What will be the output?**

**4. What would the following code produce if myList=[0,6,2,7,8,9]. If there is an error, explain why.**

**1. myList[1]**

**2. myList[0]**

**3. myList[7]**

**4. myList[5]**

**5. myList[4.0]**

**6. myList[-3]**

**7. myList[-1]**

**8. myList[2:6]**

**9. myList[4:5]**

**10.myList[6:1000]**

**11.myList[2:4][1]**

**12.myList[0:3:2]**

**13.myList[1::3]**

**14.myList[::]**

**15.myList[2::2]**

**16.myList[::3]**

**17.myList[1::2]**

**18.myList[::4]**

**19.myList[::2]**

**20.myList[:]**

**21.myList[::-1][3:5]**

**22.myList[::-1]**

**23.myList[3:5][4]**

**24.myList[1:5][2:3][1]**

**25.myList[::1]**

**26.myList[2::-2][-3]**

**5. mystr = ‘My Malayalam’**

**do the following**

**1. print the upper case of mystr**

**2. from mystr create a list with ‘My’ and ‘Malayalam’ as it’s elements using a string method**

**3. Replace the word My from mystr with Our.**

**4. Convert the lowercase characters to upper and vice versa and store to S and display the content of S**

**5. Convert mystr to tuple and display the new tuple.**

**6. country = ‘India,Pakistan,Nepal,Srilanka’**

**a,b,c,d=country.split(',')**

**s = d.upper + ‘ VS ’ + a.upper**

**print(s)**

**What will be the output?**

**7. country = ‘India,Pakistan,Nepal,Srilanka’**

**c = country\*5**

**print(c)**

**What will be the result?**

**Questions**

**1. Write a short note on split method.**

**2. Tuple is faster than list. Why?**

**3. Write a short note on the python function input.**

**4. What is the difference between list and tuple?**

-----------------------------------------------------------------------------------------------