**Python Programs**

1. **To print the message**

**>>> print ("Hello, Python!")**

**Hello, Python!**

1. **Script Mode Programming**

**Create a source file with the name test.py**

**print ("Hello, Python!")**

**and save it in Scripts folder in**

**C:\Users\srsenthi\AppData\Local\Programs\Python\Python36\Scripts**

**>>>python test.py**

1. **Quotation in Python**

**>>> word='this is a word'**

**>>> sentence="this is a sentence"**

**>>> para="""this is a paragraph**

**... which is given in 3 lines"""**

**>>> print (word)**

**this is a word**

**>>> print (sentence)**

**this is a sentence**

**>>> print (para)**

**this is a paragraph**

**which is given in 3 lines**

1. **Comments :**

**>>> # this is a comment**

**... print ("hello") # this is another comment**

**hello**

**>>>**

1. **Variable Assignment**

**>>> counter = 100 # An integer assignment**

**>>> miles = 1000.0 # A floating point**

**>>> name = "John" # A string**

**>>> print (counter)**

**100**

**>>> print (name)**

**John**

**>>> print (miles)**

**1000.0**

1. **Multiple Assignment**

**>>> a,b,c=1,2,"joe"**

**>>> print (a, b, c)**

**1 2 joe**

1. **Numbers**

**>>> 2 + 2**

**4**

**>>> 50 - 5\*6**

**20**

**>>> (50 - 5\*6) / 4**

**5.0**

**>>> 8 / 5 # division always returns a floating point number**

**1.6**

**8. division**

**=========**

**>>> 17 / 3 # classic division returns a float**

**5.666666666666667**

**>>>**

**>>> 17 // 3 # floor division discards the fractional part**

**5**

**>>> 17 % 3 # the % operator returns the remainder of the division**

**2**

**>>> 5 \* 3 + 2 # result \* divisor + remainder**

**17**

**9 .\*\* operators**

**>>> 5 \*\* 2 # 5 squared**

**25**

**>>> 2 \*\* 7 # 2 to the power of 7**

**128**

**10. Calculations**

**>>> 5 \*\* 2 # 5 squared**

**25**

**>>> 2 \*\* 7 # 2 to the power of 7**

**128**

**11. Desk Calculator program**

**tax = 12.5 / 100**

**>>> price = 100.50**

**>>> price \* tax**

**12.5625**

**>>> price + \_**

**113.0625**

**>>> round(\_, 2)**

**113.06**

**Strings :**

**>>> 'oracle university'**

**'oracle university'**

**>>> 'doesn\'t'**

**"doesn't"**

**>>> "doesn\'t"**

**"doesn't"**

**>>> "\"Yes,\" he said."**

**'"Yes," he said.'**

**>>> s='firstline\nsecondline.'**

**>>> s**

**'firstline\nsecondline.'**

**>>> print(s)**

**firstline**

**secondline.**

**>>> print('name \n age\n dateofBirth')**

**name**

**age**

**dateofBirth**

**\n- new line**

**String literal span multiple lines**

**>>> print('C:\some\name')**

**C:\some**

**ame**

**>>> print("""\**

**... Usage: thingy [OPTIONS]**

**... -h Display this usage message**

**... -H hostname Hostname to connect to**

**... """)**

**Usage: thingy [OPTIONS]**

**-h Display this usage message**

**-H hostname Hostname to connect to**

**Strings concatenated using + operator and repeated with \***

**>>> 3 \* 'un' + 'ium'**

**'unununium'**

**>>> 'Py' 'thon'**

**'Python'**

**Concatenate variables**

**>>> prefix='py'**

**>>> prefix + 'thon'**

**'python'**

**>>> text**

**'Put several strings within parentheses to have them joined together.'**

**Strings – Indexing and slicing**

**=========================**

**>>> word= 'Python'**

**>>> word[0]**

**'P'**

**>>> word[1]**

**'y'**

**>>> word[2]**

**'t'**

**>>> word[-1]**

**'n'**

**>>> word[-2]**

**'o'**

**>>> word[:2] + word[2:]**

**'Python'**

**>>> word[0:3]**

**'Pyt'**

**>>> word[3:7]**

**'hon'**

**Python Strings – immutable**

**>>> word[0]='l'**

**Traceback (most recent call last):**

**File "<stdin>", line 1, in <module>**

**TypeError: 'str' object does not support item assignment**

**Python – Lists – contains heterogeneous datatypes – IIIr to arrays**

**list = [ 'abcd', 786 , 2.23, 'john', 70.2 ]**

**tinylist = [123, 'john']**

**print ( list) # Prints complete list**

**print (list[0]) # Prints first element of the list**

**print (list[1:3]) # Prints elements starting from 2nd till 3rd**

**print (list[2:]) # Prints elements starting from 3rd element**

**print (tinylist \* 2) # Prints list two times**

**print (list + tinylist )# Prints concatenated lists**

**eg:2**

**cubes = [1, 8, 27, 65, 125] # something's wrong here**

**>>> 4 \*\* 3 # the cube of 4 is 64, not 65!**

**64**

**>>> cubes[3] = 64 # replace the wrong value**

**>>> cubes**

**[1, 8, 27, 64, 125]**

**>>> cubes.append(216) # add the cube of 6**

**>>> cubes.append(7 \*\* 3) # and the cube of 7**

**>>> cubes**

**[1, 8, 27, 64, 125, 216, 343]**

**Assignment to slices**

**==================**

**>>> letters = ['a', 'b', 'c', 'd', 'e', 'f', 'g']**

**>>> letters**

**['a', 'b', 'c', 'd', 'e', 'f', 'g']**

**>>> # replace some values**

**>>> letters[2:5] = ['C', 'D', 'E']**

**>>> letters**

**['a', 'b', 'C', 'D', 'E', 'f', 'g']**

**>>> # now remove them**

**>>> letters[2:5] = []**

**>>> letters**

**['a', 'b', 'f', 'g']**

**>>> # clear the list by replacing all the elements with an empty list**

**>>> letters[:] = []**

**>>> letters**

**[]**

**>>> letters = ['a', 'b', 'c', 'd']**

**>>> len(letters)**

**4**

**Nest ists**

**>>> a = ['a', 'b', 'c']**

**>>> n = [1, 2, 3]**

**>>> x = [a, n]**

**>>> x**

**[['a', 'b', 'c'], [1, 2, 3]]**

**>>> x[0]**

**['a', 'b', 'c']**

**>>> x[0][1]**

**'b'**

**Tuples – read only lists – cannot be updated**

**tuple = ( 'abcd', 786 , 2.23, 'john', 70.2 )**

**tinytuple = (123, 'john')**

**print (tuple) # Prints complete list**

**print (tuple[0]) # Prints first element of the list**

**print (tuple[1:3]) # Prints elements starting from 2nd till 3rd**

**print (tuple[2:]) # Prints elements starting from 3rd element**

**print (tinytuple \* 2) # Prints list two times**

**print (tuple + tinytuple) # Prints concatenated lists**

**Dictionary**

**dict = {}**

**dict['one'] = "This is one"**

**dict[2] = "This is two"**

**tinydict = {'name': 'john','code':6734, 'dept': 'sales'}**

**print (dict['one']) # Prints value for 'one' key**

**print (dict[2]) # Prints value for 2 key**

**print (tinydict) # Prints complete dictionary**

**print (tinydict.keys()) # Prints all the keys**

**print (tinydict.values()) # Prints all the values**

**Fibanocci series**

**1.a, b = 0, 1**

**>>> while b < 10:**

**... print(b)**

**... a, b = b, a+b**

**2. >>> i = 256\*256**

**>>> print('The value of i is', i)**

**The value of i is 65536**

**3. >>> a, b = 0, 1**

**>>> while b < 1000:**

**... print(b, end=',')**

**... a, b = b, a+b**

**...**

**1,1,2,3,5,8,13,21,34,55,89,144,233,377,610,987,**

**How to clear the screen in command line**

**Import os**

**>>>os.system(‘cls’)**

**In Unix/Linux**

**>> import os**

**>>> os.system(‘clear’)**

**Conditional Statements**

**1.>>> x = int(input("Please enter an integer: "))**

**Please enter an integer: 42**

**>>> if x < 0:**

**... x = 0**

**... print('Negative changed to zero')**

**... elif x == 0:**

**... print('Zero')**

**... elif x == 1:**

**... print('Single')**

**... else:**

**... print('More')**

**2. a=10**

**if a==10:**

**print "Hello User"**

**3. Leap Year**

**==========**

**>>> year=2000**

**>>> if year%4==0:**

**... print ("year is leap")**

**... else:**

**... print("year is not leap")**

**...**

**year is leap**

**3. >>> a=10**

**>>> if a>=20:**

**... print ("condition is true")**

**... else:**

**... if a>=15:**

**... print("checking second value")**

**... else:**

**... print("all conditions are false")**

**...**

**all conditions are false**

**4.to display the table of numbers**

**>>> num=2**

**>>> for a in range (1,6):**

**... print (num \* a)**

**...**

**5.** **for letter in 'Python': # First Example**

**print ('Current Letter :', letter)**

**fruits = ['banana', 'apple', 'mango']**

**for fruit in fruits: # Second Example**

**print 'Current fruit :', fruit**

**print "Good bye!"**

**6. fruits = ['banana', 'apple', 'mango']**

**for index in range(len(fruits)):**

**print ('Current fruit :', fruits[index])**

**print ("Good bye!")**

**7 . Search for prime numbers 10 through 20 .**

**for num in range(10,20): #to iterate between 10 to 20**

**for i in range(2,num): #to iterate on the factors of the number**

**if num%i == 0: #to determine the first factor**

**j=num/i #to calculate the second factor**

**print ('%d equals %d \* %d' % (num,i,j))**

**break #to move to the next number, the #first FOR**

**else: # else part of the loop**

**print (num, 'is a prime number')**

**While loop**

**count = 0**

**while (count < 9):**

**print ('The count is:', count)**

**count = count + 1**

**print "Good bye!"**

**Infinite loop**

**var = 1**

**while var == 1 : # This constructs an infinite loop**

**num = raw\_input("Enter a number :")**

**print ("You entered: ", num)**

**print ("Good bye!")**

**Loops with else statements**

**count = 0**

**while count < 5:**

**print (count, " is less than 5")**

**count = count + 1**

**else:**

**print (count, " is not less than 5")**

**Single Statement Suite**

**flag = 1**

**while (flag): print 'Given flag is really true!'**

**print ("Good bye!")**

**Find the sum of Natural numbers from 1 to 10.**

**sum=0**

**for n in range(1,11):**

**sum+=n**

**print (sum)**

**Nested Loops**

**>>> for i in range (1,6):**

**... for j in range(1,i+1):**

**... print (i,end=" ")**

**... print()**

**...**

**Print the pyramid**

**>>> for i in range (1,6):**

**... for j in range (5,i-1,-1):**

**... print("\*",end="")**

**... print()**

**While loop**

**a=10**

**while a>0:**

**print "Value of a is",a**

**a=a-2**

**Program to add digits of a number**

**n=153**

**sum=0**

**while n>0:**

**r=n%10**

**sum+=r**

**n=n/10**

**print (sum)**

**Break statement**

**for i in [1,2,3,4,5]:**

**if i==4:**

**print ("Element found")**

**break**

**print (i)**

**Continue Statement**

**a=0**

**while a<=5:**

**a=a+1**

**if a%2==0:**

**continue**

**print (a)**

**print ("End of Loop")**

**Pass Statement**

**for i in [1,2,3,4,5]:**

**if i==3:**

**pass**

**print ("Pass when value is",i)**

**print (i)**

**Getting current time**

**import time;**

**localtime = time.localtime(time.time())**

**print "Local current time :", localtime**

**local current time: time.struct\_time(tm\_year=2017, tm\_mon=2, tm\_mday=16, tm\_hour=22, tm\_min=11, tm\_sec=46, tm\_wday=3, tm\_yday=47, tm\_isdst=0)**

**>>>**

**Getting Formatted time**

**import time;**

**localtime = time.asctime( time.localtime(time.time()) )**

**print "Local current time :", localtime**

**import calendar**

**cal = calendar.month(2008, 1)**

**print "Here is the calendar:"**

**print cal**