***: PYTHON NOTES :***

* *Integer divide by integer always an integer in java on in python 2.1 version.*
* *In the version 3.6 integer by integer floating point number.*

***Rules to name an identifier:***

* *The identifier should not pre-defined words.*
* *Identifier should not start with numeric value.*
* *Identifiers can’t be more than 32 character.*
* *Except’\_’ no other special symbol are use while naming identifiers.*
* *Identifiers must start with ‘\_’or alphabet.*

*A=20 it will be having its own address.*

*/\*program\*/*

*>>> a=20*

*>>> print a*

*20*

***/\*Ex.2***

*>>> a=20.3*

*>>> a*

*20.3*

*>>> print a*

*20.3*

*>>> type(a)*

*<type 'float'>*

***/\*Ex.3***

*>>> a=2*

*>>> type(a)*

*<type 'int'>*

***/\*Ex.4***

*>>> a="yash"*

*>>> a*

*'yash'*

*>>> type(a)*

*<type 'str'>*

***/\*Ex.5***

*>>>type(20.3)*

*<class ‘float’>*

***Type casting:***

* ***Int :*** *used to covert any char into an int value.*

***Syntax:*** *int(value/var).*

***/\*sample typecasting\*/***

*>>> a=20.4*

*>>> a*

*20.4*

*>>> int(a) //typecasting*

*20*

*>>> a*

*20.4*

*>>> a=int(a)//typecasting value is initilazed in a.*

*>>> a*

*20*

* ***Float:*** *used to convert any char into an float value.*

***Syntax:*** *float(value/var).*

**/\*typecasting\*/**

*a=23*

*>>> float(a)*

*23.0*

*d=float(3)*

*>>> d*

*3.0*

* *String: String is collection of character enclosed with in “ “or’ ‘ or ‘’’ ‘’’ or””” “”” codes.*

*used to convert set of char’s into an string value.*

***Syntax:*** *str(var/value).*

*>>>"yashwanth"*

*'yashwanth'*

*>>> 'yashwanth'*

*'yashwanth'*

***>>> ""yashwanth""***

***SyntaxError: invalid syntax***

*>>> """ yashwanth """*

*' yashwanth '*

*>>> '''yashwanth'''*

*'yashwanth'*

*>>>a='yashwanth'*

*>>> a*

*'yashwanth'*

*>>> type(a)*

*<type 'str'>*

***List:*** *It is a collection of hemogenous or hetrogenous data items which should be enclosed within in a set of* ***[ ].***

***Syntax:*** *uname [ var,var2,var3………..varn].*

*>>> a=[2,3,4,5]*

*>>> type(a)*

*<type 'list'>*

*>>> a=(2,3,4,5)*

*>>> type(a)*

*<type 'tuple'>*

***Set:*** *used to convert set of char’s into an set.*

***Def’n:*** *Set is a collection of non repeated data items which are enclosed within pair of* ***{ }****.*

***Syntax:***

*Uname={ var,var2,var3………..varn }*

*A={2,3,4………..n}*

***Tuple:*** *It is a set a collection set of hemogenous or hetrogenous data items which are enclosed within circular bracisis and can’t be modified.*

***Syntax:*** *tuple=( group of value).*

***Dictionary:*** *It is a collection of values which are identified with the help of unique key. In dictionary the repeated key is prohibited.*

***Syntax:***

*dictionary name={key:value,key2:value2…………..key n:value n}.*

***Datatype:*** *String is a collection of char which are enclosed with in double qoutes “ “ or ‘ ‘ or “”” “”” or ‘’’ ‘’’.*

***Syntax:*** *varname =” character” or ‘ character‘ .*

*Each and every char in String identified with form variable name.*

*Backward indexing*

***Syntax:*** *vname[ indexvalue].*

*A=”Jspiders”*

*-8 -7 -6 -5 -4 -3 -2 -1*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *J* | *S* | *P* | *I* | *D* | *E* | *R* | *S* |

*0 1 2 3 4 5 6 7*

*a="Jspiders"*

*Forword indexing*

*>>> a[3]*

*'i'*

*>>> a[5]*

*'e'*

*>>> a[-2]*

*'r'*

***/\*to find substring\*/***

***Syntax:****vname=[ start point: end point]*

*>>>a[0:3]*

*'Jsp'*

*/\*Ex.2\*/*

*>>> a[-1:-4]*

*''*

*Vname=[:endpoint] //not to specifiy startpoint*

*/\*example\*/*

*>>>a[:4]*

*'Jspi'*

*/\*Ex:2\*/*

*>>>a[:-2]*

*'Jspide'*

*Vname= [startpoint:] //not to specifiy endpoint*

*/\*example\*/*

*>>> a[2:]*

*'piders'*

*/\*ex.2\*/*

*>>>a[-2:]*

*'rs’*

***To print the even position numbers***

***Syntax:*** *vname[startpoint:endpoint:inc/dec value of index].*

/\*ex:\*/

*>>> a[0::2]*

*'Jpdr'*

***To print the odd position numbers.***

***/\*ex\*/***

*>>> a[1::2]*

*'sies'*

***Slice:***

***Syntax:*** *Strname[startpoint:endpoint].*

/\*Ex.

*>>> a[::2]*

*'ysw'*

***Multiple of three:***

*/\*Ex.*

*>>> a="jspiders"*

*>>> a*

*'jspiders'*

*>>> a[0::3]*

*'jir'*

***Length of sring:***

***Syntax:*** *len(vname).*

***/\*Ex:***

***>>>****a=”Jspiders”*

*a*

*>>> len(a)*

*8*

***Reversed String:***

*/\*Ex:*

***>>>****a=”Jspiders”*

*a*

*>>> a[::-1]*

*'sredipsj'*

*/\*Ex:*

*>>> a[::-2]*

*'seis'*

* *When we are decrementing our index then the default the starting and will be consider as -1 if the user as not mention the starting point.*
* *If we want to get even position from last.*

*/\*Ex:*

*>>> a[-2::-2]*

*'rdpj'*

*Strname[startpoint:endpoint:updation]*

*Strname[::updation]*

*Strname[stpt::updation]*

***To find the number of character of string:***

*>>>a=”jspiders”*

*>>> len(a)*

*8*

***Import statement:***

*Import requirement path/requirement name.*

*>>> import math*

*>>> dir(math)*

*['\_\_doc\_\_', '\_\_name\_\_', '\_\_package\_\_', 'acos', 'acosh', 'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'copysign', 'cos', 'cosh', 'degrees', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs', 'factorial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'hypot', 'isinf', 'isnan', 'ldexp', 'lgamma', 'log', 'log10', 'log1p', 'modf', 'pi', 'pow', 'radians', 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'trunc']*

*/\*Ex:*

*>>> math.pi*

*3.141592653589793*

***To find square root function***

*>>> math.sqrt*

*<built-in function sqrt>*

*/\*Ex:*

*>>> math.sqrt (4443556)*

*2107.9743831460573*

*/\*Ex:*

*>>> math.sqrt(443556)*

*666.0*

*/\*Ex:*

*>>> math.sqrt(99980001)*

*9999.0*

***To find power value’s:***

***/\*Ex:***

*>>> math.pow(111,2)*

*12321.0*

*/\*Ex:*

*>>> math.pow(420,20)*

*2.9173316787576666e+52*

***To find degree to radians:***

***/\*Ex:***

*>>> math.radians(90)*

*1.5707963267948966*

***To find the values of trignometry like(sin,tan,cos):***

*>>> math.sin(90)*

*0.8939966636005579*

*/\*Ex:*

*>>> math.sin (math.radians(90))*

*1.0*

*/\*Ex:*

*>>> math.tan (math.radians(90))*

*1.633123935319537e+16*

*/\*Ex:*

*>>> math.cos(math.radians(90))*

*6.123233995736766e-17*

***Log operend:***

***/\*Ex:***

*>>> math.log(20)*

*2.995732273553991*

*/\*Ex:*

*>>> math.log(20,20)*

*1.0*

*/\*Ex:*

*>>> math.log(2,10)*

*0.30102999566398114*

*/\*Ex:*

*>>> math.log (5,10)*

*0.6989700043360187*

*/\*Ex:*

*>>> math.log(8)*

*2.0794415416798357*

***To find ceil operation:***

*>>> math.ceil (1.8)*

*2.0*

***To find floor operation:***

*>>> math.floor (1.5)*

*1.0*

***To find the factorial:***

***/\*Ex:***

*>>> math.factorial(5)*

*120*

*To find GCD:*

*/\*Ex:*

*>>>math.gcd(2,4)//this will be done in above version 3.6.2*

*2*

*>>>math.gcd(4,math . gcd(20,8)) //this will be done in above version 3.6.2*

*4*

*/\*Ex:*

*>>> from math import sin*

*>>> sin(0)*

*0.0*

*>>> sin*

*<built-in function sin>*

*/\*Ex:*

*>>> from math import tan,cos*

*>>> from math import log,pi*

*>>> log(10,2)*

*3.3219280948873626*

*>>> pi*

*3.141592653589793*

***To perform string operation:***

*>>> import string*

*>>> dir(string)*

*['Formatter', 'Template', '\_TemplateMetaclass', '\_\_builtins\_\_', '\_\_doc\_\_', '\_\_file\_\_', '\_\_name\_\_', '\_\_package\_\_', '\_float', '\_idmap', '\_idmapL', '\_int', '\_long', '\_multimap', '\_re', 'ascii\_letters', 'ascii\_lowercase', 'ascii\_uppercase', 'atof', 'atof\_error', 'atoi', 'atoi\_error', 'atol', 'atol\_error', 'capitalize', 'capwords', 'center', 'count', 'digits', 'expandtabs', 'find', 'hexdigits', 'index', 'index\_error', 'join', 'joinfields', 'letters', 'ljust', 'lower', 'lowercase', 'lstrip', 'maketrans', 'octdigits', 'printable', 'punctuation', 'replace', 'rfind', 'rindex', 'rjust', 'rsplit', 'rstrip', 'split', 'splitfields', 'strip', 'swapcase', 'translate', 'upper', 'uppercase', 'whitespace', 'zfill']*

*>>> s="jspiders"*

*>>> dir(s)*

*['\_\_add\_\_', '\_\_class\_\_', '\_\_contains\_\_', '\_\_delattr\_\_', '\_\_doc\_\_', '\_\_eq\_\_', '\_\_format\_\_', '\_\_ge\_\_', '\_\_getattribute\_\_', '\_\_getitem\_\_', '\_\_getnewargs\_\_', '\_\_getslice\_\_', '\_\_gt\_\_', '\_\_hash\_\_', '\_\_init\_\_', '\_\_le\_\_', '\_\_len\_\_', '\_\_lt\_\_', '\_\_mod\_\_', '\_\_mul\_\_', '\_\_ne\_\_', '\_\_new\_\_', '\_\_reduce\_\_', '\_\_reduce\_ex\_\_', '\_\_repr\_\_', '\_\_rmod\_\_', '\_\_rmul\_\_', '\_\_setattr\_\_', '\_\_sizeof\_\_', '\_\_str\_\_', '\_\_subclasshook\_\_', '\_formatter\_field\_name\_split', '\_formatter\_parser', 'capitalize', 'center', 'count', 'decode', 'encode', 'endswith', 'expandtabs', 'find', 'format', 'index', 'isalnum', 'isalpha', 'isdigit', 'islower', 'isspace', 'istitle', 'isupper', 'join', 'ljust', 'lower', 'lstrip', 'partition', 'replace', 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill']*

*/\*Ex:*

*>>> s.capitalize()*

*'Jspiders'*

*/\*Ex:*

*>>> s=s.capitalize()*

*>>> s*

*'Jspiders'*

***Capitalize:***

*>>> help(s.capitalize)*

*Help on built-in function capitalize:*

*capitalize(...)*

*S.capitalize() -> string*

*Return a copy of the string S with only its first character*

*capitalized.*

*/\*Ex:*

*>>> s2="jSPIDERS"*

*>>> s2*

*'jSPIDERS'*

*>>> s2.capitalize ()*

*'Jspiders'*

***Count:****To count the number of occurrence of a character.*

*>>> help(s.count)*

*Help on built-in function count:*

*count(...)*

*S.count(sub[, start[, end]]) -> int*

*Return the number of non-overlapping occurrences of substring sub in*

*string S[start:end]. Optional arguments start and end are interpreted*

*as in slice notation.*

*/\*Ex:*

*>>> a="yashwanthyash"*

*>>> a.count('y')*

*2*

***Types of string usage:***

*Str.count(substr)*

*Str.count(substr.stpt)*

*/\*Ex:*

*>>> a.count ('y',3)*

*2*

*Str.count(substr,stpt,endpt)*

*/\*Ex:*

*>>> a.count('y',0,3)*

*0*

*String variable can be a variable or value.*

***Find:***

*>>> help(s.find)*

*Help on built-in function find:*

*find(...)*

*S.find(sub [,start [,end]]) -> int*

*Return the lowest index in S where substring sub is found,*

*such that sub is contained within S[start:end]. Optional*

*arguments start and end are interpreted as in slice notation.*

*Return -1 on failure.*

*/\*Ex:*

*>>> a="Hello hai how are you"*

*>>> a.find('w')*

*12*

*/\*Ex:*

*>>> a="Hello hai how are you"*

*>>> a.find('h')*

*6*

*/\*Ex:*

*>>> a="Hello hai how are you"*

*>>> a.find('h',7)*

*10*

*/\*Ex:*

*>>> a="Hello hai how are you"*

*>>> a.find('h',3,6)*

*-1*

*/\*Ex:*

*>>> a="Hello hai how are you"*

*>>> a.find('hai')*

*6*

*Starting point of the substring which is present in the given string.*

***Index:***

*>>> help(s.index)*

*Help on built-in function index:*

*index(...)*

*S.index(sub [,start [,end]]) -> int*

*Return the lowest index in S where substring sub is found,*

*such that sub is contained within S[start:end]. Optional*

*arguments start and end are interpreted as in slice notation.*

*Like S.find() but raise ValueError when the substring is not found.*

*/\*Ex:*

*>>> a="Hello hai how are you"*

*>>> a.index("ho")*

*10*

*/\*Ex:*

*>>> a.index('h',4,7)*

*6*

***Isalpha:***

*It is going to check weather the enter character is alphabet or not.*

*/\*Ex:*

*>>> a='k'*

*>>> b='y'*

*>>> a.isalpha ()*

*True*

*/\*Ex:*

*>>> c=3*

*>>> c.isalpha()*

*Traceback (most recent call last):*

*File "<pyshell#50>", line 1, in <module>*

*c.isalpha()*

*AttributeError: 'int' object has no attribute 'isalpha'*

*/\*Ex:*

*>>> c='3'*

*>>> c.isalpha()*

*False*

*/\*Ex:*

*>>> b.isalpha ()*

*True*

*/\*Ex:*

*>>> a="abdce"*

*>>> a.isalpha()*

*True*

*/\*Ex:*

*>>> a="asdg34"*

*>>> a.isalpha()*

*False*

***Isdigit:***

*/\*Ex:*

*>>> '3'.isdigit()*

*True*

*/\*Ex:*

*>>> 3.isdigit()*

*SyntaxError: invalid syntax*

*/\*Ex:*

*>>> a="4"*

*>>> a.isdigit ()*

*True*

*/\*Ex:*

*>>> b*

*'y'*

*>>> b.isdigit ()*

*False*

***Isalnum:***

*/\*Ex:*

*>>> a="asdg34"*

*>>> a.isalnum()*

*True*

*/\*Ex:*

*>>>b=’y’*

*>>> b.isalnum ()*

*True*

*/\*Ex:*

*>>> '#$'.isalnum ()*

*False*

***Isupper:***

*>>> help(a.isupper)*

*Help on built-in function isupper:*

*isupper(...)*

*S.isupper() -> bool*

*Return True if all cased characters in S are uppercase and there is*

*at least one cased character in S, False otherwise.*

***/\*Ex:***

*>>> s*

*'Jspiders'*

*>>> s.upper ()*

*'JSPIDERS'*

*/\*Ex:*

*>>> a="yashwanth"*

*>>> a=a.upper()*

*>>> a*

*'YASHWANTH'*

***Islower:*** *this function is used to convert a character of string to lower case.*

***Syntax:****strval.lower()*

*/\*Ex:*

*>>> a="yashwanth"*

*>>> a=a.lower ()*

*>>> a*

*'yashwanth'*

*/\*Ex:*

*>>>s=”JSPIDERS”*

*>>> s*

*'JSPIDERS'*

*>>>s=s.lower()*

*>>> s*

*'jspiders'*

*/\*Ex:*

*>>> s.islower ()*

*True*

*Returns true only when all the character’s are in lowercase.*

*>>> a="yash yashwanth"*

*>>> a*

*'yash yashwanth'*

***Split:***

*>>> a.split(" ")*

*['yash', 'yashwanth']*

*>>> a.split(" ")*

*['yash', 'yashwanth']*

*>>>a="yash yashwanth"*

*>>> a*

*'yash yashwanth'*

*>>> a.split("a")*

*['y', 'sh', ' y', 'shw', 'nth']*

***Join:***

*>>> "".join (a)*

*'ysh yshwnth'*

*>>> "a".join(a)*

*'yash yashwanth'*

***DATA STRUCTURE IN PYTHON:***

***Def'n:***

***Liner And Non-liner:***

***In linar data structure:***

*string.*

***In non-liner data structure:***

*linked list*

*tree*

*hashmap .*

***List :*** *It is collection of hemogenous or hetrogenous data items which are represented inside the pair of [ ] braces and saperated with ,( comma).*

*now declare a variable called empty list.*

*Syntax: var=[ ]*

*/\*Ex:*

*>>> var=[ ]*

*>>> type(var)*

*<type 'list'>*

*>>> a=list()*

*>>> type(a)*

*<type 'list'>*

***Initilize a list:***

***Syntax:*** *var = [ 2,3,4,5,..................n]*

*/\*Ex:*

*>>> a= [2,3,4,5,6]*

*>>> a*

*[2, 3, 4, 5, 6]*

*>>> len(a)*

*5*

***Syntax:*** *strvar=[ stpt:endpt]*

*/\*Ex:*

*>>> a= [2,3,4,5,6]*

*>>> a*

*>>> a[2:4]*

*[4, 5]*

*/\*Ex://without startpoint*

*>>> a= [2,3,4,5,6]*

*>>> a*

*>>> a[:4]*

*[2, 3, 4, 5]*

*/\*Ex://without endpoint*

*>>> a= [2,3,4,5,6]*

*>>> a*

*>>> a[0:]*

*[2, 3, 4, 5, 6]*

*/\*Ex:*

*>>> a[3:3]*

*[ ]*

*/\*Ex:*

*>>> a[0:len(a):2]*

*[2, 4, 6]*

*/\*Ex:*

*>>>a="jspiders"*

*>>> a*

*'jspiders'*

*>>> list(a)*

*['j', 's', 'p', 'i', 'd', 'e', 'r', 's']*

*>>> dir(list)*

*['\_\_add\_\_', '\_\_class\_\_', '\_\_contains\_\_', '\_\_delattr\_\_', '\_\_delitem\_\_', '\_\_delslice\_\_', '\_\_doc\_\_', '\_\_eq\_\_', '\_\_format\_\_', '\_\_ge\_\_', '\_\_getattribute\_\_', '\_\_getitem\_\_', '\_\_getslice\_\_', '\_\_gt\_\_', '\_\_hash\_\_', '\_\_iadd\_\_', '\_\_imul\_\_', '\_\_init\_\_', '\_\_iter\_\_', '\_\_le\_\_', '\_\_len\_\_', '\_\_lt\_\_', '\_\_mul\_\_', '\_\_ne\_\_', '\_\_new\_\_', '\_\_reduce\_\_', '\_\_reduce\_ex\_\_', '\_\_repr\_\_', '\_\_reversed\_\_', '\_\_rmul\_\_', '\_\_setattr\_\_', '\_\_setitem\_\_', '\_\_setslice\_\_', '\_\_sizeof\_\_', '\_\_str\_\_', '\_\_subclasshook\_\_', 'append', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']*

***Append list:***

***Syntax=****listname.append(value)*

*/\*Ex:*

*>>> a=[2,3,4,5,6,7]*

*>>> a*

*[2, 3, 4, 5, 6, 7]*

*>>> a.append(8)*

*>>> a*

*[2, 3, 4, 5, 6, 7, 8]*

***Remove list:***

*>>> a.pop()*

*8*

*/\*Ex:*

*>>> a*

*[2, 3, 4, 5, 6, 7]*

*>>> a.append(8)//Insert the element*

*>>> a.append(9)//Insert the element*

*>>> a*

*[2, 3, 4, 5, 6, 7, 8, 9]*

*>>> a.pop()//delete element*

*9*

*>>> a.pop()*

*8*

***Insert list(Stack list):***

*Append is as similar to push operator.*

*>>>a.clear()//removes from the list*

*//done in above version.*

*/\*Ex:*

*>>> a*

*[2, 3, 4, 5, 6, 7]*

*>>> a.append(3)*

*>>> a*

*[2, 3, 4, 5, 6, 7, 3]*

*>>> a.count(3)*

*2*

*/\*Ex:*

*>>> a.count(3,4)*

*Traceback (most recent call last):*

*File "<pyshell#9>", line 1, in <module>*

*a.count(3,4)*

*TypeError: count() takes exactly one argument (2 given)*

*/\*Ex:*

*>>> a.count(7)/\*count the number of occurence present*

*1*

***Sorting:***

*>>> c=[2,43,6,7,8,23,45]*

*>>> c.sort()*

*>>> c*

*[2, 6, 7, 8, 23, 43, 45]*

*>>> sorted()*

*Traceback (most recent call last):*

*File "<pyshell#14>", line 1, in <module>*

*sorted()*

*TypeError:* ***Required argument 'iterable' (pos 1) not found***

*/\*Ex:* ***another type of sorting :***

*>>> sorted(c)*

*[2, 6, 7, 8, 23, 43, 45]//Accending order.*

*/\*****Ex:Decending order.***

*>>> c[::-1]*

*[45, 43, 23, 8, 7, 6, 2]*

***Collections in PYTHON:***

*>>> import collections*

*>>> dir(collections)*

*['Callable', 'Container', 'Counter', 'Hashable', 'ItemsView', 'Iterable', 'Iterator', 'KeysView', 'Mapping', 'MappingView', 'MutableMapping', 'MutableSequence', 'MutableSet', 'OrderedDict', 'Sequence', 'Set', 'Sized', 'ValuesView', '\_\_all\_\_', '\_\_builtins\_\_', '\_\_doc\_\_', '\_\_file\_\_', '\_\_name\_\_', '\_\_package\_\_', '\_abcoll', '\_chain', '\_class\_template', '\_eq', '\_field\_template', '\_get\_ident', '\_heapq', '\_imap', '\_iskeyword', '\_itemgetter', '\_repeat', '\_repr\_template', '\_starmap', '\_sys', 'defaultdict', 'deque', 'namedtuple'].*

***Remove function:***

* *Can remove one variable.*

*/\*Ex:*

*>>> a.remove (2)*

*>>> a*

*[2, 24, 34, 64, 646]*

* ***more than one variable means throw an error.***

*>>> a.remove ([2,34,64])*

*Traceback (most recent call last):*

*File "<pyshell#10>", line 1, in <module>*

*a.remove ([2,34,64])*

*ValueError: list.remove(x): x not in list*

***Reverse Function:***

***/\*Ex:***

*>>> a*

*[2, 24, 34, 64, 646]*

*>>> a.reverse ()*

*>>> a*

*[646, 64, 34, 24, 2]*

*/\*Ex:*

*>>> a[::-1]*

*[2, 24, 34, 64, 646]*

*>>> a=a[::-1]//Actual value is modified.*

*>>> a*

*[2, 24, 34, 64, 646]*

***Insert function:***

*Syntax: listname.insert(position,element).*

*/\*Ex:*

*>>> c=[2,3,4,5,7]*

*>>> c.insert (4,6)*

*>>> c*

*[2, 3, 4, 5, 6, 7]//element is inserted at 4th* ***Index****.*

*/\*Ex:*

*>>> c*

*[2, 3, 4, 5, 6, 7]*

*>>> c[2:5]*

*[4, 5, 6]*

*>>> c[::2]//even position of list*

*[2, 4, 6]*

*>>> c[1::2]//odd position of the list, must specifiy the start point.*

*[3, 5, 7]*

*/\*Ex:*

*>>> len(c)//to find the length*

*6*

***To find String into char list:***

***/\*Ex:***

*>>> list("yash")*

*[ 'y', 'a', 's', 'h']*

***To find char into ASCII value:***

*function* ***ord*** *is help to convert char to ASCII value*

*Syntax: ord(char)*

*/\*Ex:*

*>>> ord('a')*

*97*

***To find ASCII value to char:***

***/\*Ex:***

*>>> chr(97)//to pass only one value*

*'a'*

***Set:*** *It is a ordered Collection of hemogenous or hetrogenous non repeated data items.*

* *Always the elements should be enclosed with { }.*
* *To create an empty list we use set of ( ).*
* *setvar=set( ).*
* *As like list we cannot create an empty set using set variable = { }.*

*>>> dir(set)*

*['\_\_and\_\_', '\_\_class\_\_', '\_\_cmp\_\_', '\_\_contains\_\_', '\_\_delattr\_\_', '\_\_doc\_\_', '\_\_eq\_\_', '\_\_format\_\_', '\_\_ge\_\_', '\_\_getattribute\_\_', '\_\_gt\_\_', '\_\_hash\_\_', '\_\_iand\_\_', '\_\_init\_\_', '\_\_ior\_\_', '\_\_isub\_\_', '\_\_iter\_\_', '\_\_ixor\_\_', '\_\_le\_\_', '\_\_len\_\_', '\_\_lt\_\_', '\_\_ne\_\_', '\_\_new\_\_', '\_\_or\_\_', '\_\_rand\_\_', '\_\_reduce\_\_', '\_\_reduce\_ex\_\_', '\_\_repr\_\_', '\_\_ror\_\_', '\_\_rsub\_\_', '\_\_rxor\_\_', '\_\_setattr\_\_', '\_\_sizeof\_\_', '\_\_str\_\_', '\_\_sub\_\_', '\_\_subclasshook\_\_', '\_\_xor\_\_', 'add', 'clear', 'copy', 'difference', 'difference\_update', 'discard', 'intersection', 'intersection\_update', 'isdisjoint', 'issubset', 'issuperset', 'pop', 'remove', 'symmetric\_difference', 'symmetric\_difference\_update', 'union', 'update']*

*/\*Ex:*

*>>> a={2,3,5,4,3,6,7,3,6,8,6}*

*>>> a*

*set([2, 3, 4, 5, 6, 7, 8])*

***ADD:***

*>>> a.add(9)*

*>>> a*

*set([2, 3, 4, 5, 6, 7, 8, 9])*

*/\*Ex:*

*>>> a.add('\n')*

*>>> a*

*set([2, 3, 4, 5, 6, 7, 8, 9, '\n'])*

***CLEAR:***

***/\*Ex:***

*>>> a.clear ()//it will be removed.*

*>>> a*

*set([])*

***To find a subset:***

*/\*Ex:*

*>>> a={2,3,4,5,6,7,8,9,0}*

*>>> a*

*set([0, 2, 3, 4, 5, 6, 7, 8, 9])*

*>>> b={2,3,4,5}*

*>>> b.issubset (a)*

*True*

***Super Set:***

*/\*Ex:*

*>>> a={2,3,4,5,6,7,8,9,0}*

*>>> a*

*set([0, 2, 3, 4, 5, 6, 7, 8, 9])*

*>>> b={2,3,4,5}*

*>>> a.issuperset (b)*

*True*

*>>> b.issuperset (a)*

*False*

***Disjoint Set: Should not have any comman element.***

*>>> b.isdisjoint (a)*

*False*

***Difference Set :Difference element will be removed.***

*/\*Ex:*

*>>> a.difference (b)*

*set([0, 8, 9, 6, 7])*

*/\*Ex:*

*>>> b.difference (a)*

*set([])//empty set*

***UNION: to join 2 sets.***

*/\*Ex:*

*>>> a.union (b)*

*set([0, 2, 3, 4, 5, 6, 7, 8, 9])*

*/\*Ex:more than 2 elements*

*>>> (a.union (b)).union (c)*

*>>> a={2,3,4,5}*

*>>> b={2,4,5,6}*

*>>> c={4,5,6,7}*

*set([2, 3, 4, 5, 6, 7])*

***Intersection:tacking the common element b/w 2 sets.***

*/\*Ex:*

*>>> a.intersection (b)*

*set([2, 3, 4, 5])*

*/\*Ex:more than 2 sets*

*>>> a={2,3,4,5}*

*>>> b={2,4,5,6}*

*>>> c={4,5,6,7}*

*>>> (a.intersection(b)).intersection(c)*

*set([4, 5]).*

***POP:***

*/\*Ex:*

*>>> a.pop() //delete from first.*

*0*

*>>> a.pop()*

*2*

***Removed:***

*/\*Ex;*

*>>> a.remove (5)*

*>>> a*

*set([3, 4, 6, 7, 8, 9]) //5 element is removed.*

* *Set will not support for index variable.*

***TUPLE:***

***Def'n:*** *It is a Immutable (Non modifiable) set of hemogenous or hetrogenous data items which are enclosed within the pair of braces or paranthisis.*

*Tuple is intitilized as vname= ( elements............)*

*or we can say vname = values,..............*

*/\*Ex:*

*>>> vname=(2,3,45)*

*>>> vname = 2,3,4,5,6*

*>>> vname*

*(2, 3, 4, 5, 6)*

***To create an empty tuple:***

*>>> a=tuple ()*

*>>> a*

*()*

*(OR) another way*

*>>> c= ()*

*>>> c*

*()*

*insertion and deletion is prohibited in tuple.*

*>>> dir(tuple)*

*['\_\_add\_\_', '\_\_class\_\_', '\_\_contains\_\_', '\_\_delattr\_\_', '\_\_doc\_\_', '\_\_eq\_\_', '\_\_format\_\_', '\_\_ge\_\_', '\_\_getattribute\_\_', '\_\_getitem\_\_', '\_\_getnewargs\_\_', '\_\_getslice\_\_', '\_\_gt\_\_', '\_\_hash\_\_', '\_\_init\_\_', '\_\_iter\_\_', '\_\_le\_\_', '\_\_len\_\_', '\_\_lt\_\_', '\_\_mul\_\_', '\_\_ne\_\_', '\_\_new\_\_', '\_\_reduce\_\_', '\_\_reduce\_ex\_\_', '\_\_repr\_\_', '\_\_rmul\_\_', '\_\_setattr\_\_', '\_\_sizeof\_\_', '\_\_str\_\_', '\_\_subclasshook\_\_', 'count', 'index']*

***COUNT:***

*>>> vname*

*(2, 3, 4, 5, 6)*

*>>> vname = (2,3,4,5,6,7,2,3,4)*

*>>> vname*

*(2, 3, 4, 5, 6, 7, 2, 3, 4)*

*>>> vname.count(2)*

*2*

***INDEX:***

***/\*Ex:***

*>>> vname*

*(2, 3, 4, 5, 6, 7, 2, 3, 4)*

*>>> vname.index (3)*

*1*

*>>> help(vname.index)*

*Help on built-in function index:*

*index(...)*

*T.index(value, [start, [stop]]) -> integer -- return first index of value.*

*Raises ValueError if the value is not present.*

*/\*Ex:*

*>>> vname.index(2,4)*

*6*

*Syntax:vname.index(element,startpoint,endpoint)*

***To convert anything in a list:***

*vname=list(vname)*

*/\*Ex:*

*>>> vname='yashwanthyash'*

*>>> v=list(vname)*

*>>> v*

*['y', 'a', 's', 'h', 'w', 'a', 'n', 't', 'h', 'y', 'a', 's', 'h']*

***To convert anything in a set:***

***/\*Ex:***

*>>> vname = set(vname)*

*>>> vname*

*set(['a', 'h', 'n', 's', 't', 'w', 'y'])*

***To convert anything in a tuple:***

***/\*Ex:***

*>>> vname=tuple(vname)*

*>>> vname*

*('a', 'h', 'n', 's', 't', 'w', 'y')*

***Range Function:***

*A function which is used to get series of value is called as* ***range function.***

***Syntax:****range(stpt)*

*/\*Ex:*

*>>> range(0)*

*[]*

*range( stpt,endpt)*

*/\*Ex:*

*>>> range(0,10)*

*[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]*

*( OR )*

*>>> list(range (0,10))/\*In the form of list\*/*

*[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]*

*>>> set (range (0,9))/\*In the form of set\*/*

*set([0, 1, 2, 3, 4, 5, 6, 7, 8])*

*>>> tuple( range (0,9))/\*In the form of tuple\*/*

*(0, 1, 2, 3, 4, 5, 6, 7, 8)*

*In version 2.7 the range function will directly give me the group of values in form of list.*

*xrange(0,9) in 2.7version.*

*/\*Ex:*

*>>> xrange(0,9)*

*xrange(9)*

*>>> list (xrange (0,9))*

*[0, 1, 2, 3, 4, 5, 6, 7, 8]*

*/\*Ex:*

*>>> set (xrange (0,9))*

*set([0, 1, 2, 3, 4, 5, 6, 7, 8])*

*/\*Ex;*

*>>> tuple(xrange (0,9))*

*(0, 1, 2, 3, 4, 5, 6, 7, 8)*

*/\*Ex:*

*>>> list(range(20))*

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

*By default it is taking as ZERO.*

*/\*Ex:*

*>>> list (xrange (0,20,2))/\*Even numbers*

*[0, 2, 4, 6, 8, 10, 12, 14, 16, 18]*

***Parameters:***

*range(endpt)*

*range( stpt,endpt)*

*prints value>=stpt*

*value<endpt*

*range( stpt,endpt,update) //i,e update may be +ve or -ve.*

*range is very imp for looping function.*

*/\*Ex:*

*>>> list (xrange (20,0,-2))*

*[20, 18, 16, 14, 12, 10, 8, 6, 4, 2] //reverse order*

*/\*Ex:*

*>>> list (xrange (0,20,.2))*

*Traceback (most recent call last):*

*File "<pyshell#12>", line 1, in <module>*

*list (xrange (0,20,.2))*

*TypeError: integer argument expected, got float*

***Dictionary:***

*Dictionary is a collection of non repitative is an values where each value will be identified with the help of key but not with help of index and the dictionary value should be enclosed with the pair of* ***{ }.***

*Always we need key interns of char or string.*

*How do we declare a dictionary*

***Synatx:*** *var = { key2:value2 , key3:value2 , .......................}*

*Every value must have key.*

*Repeated keys are not allowed.*

*To create a empty dictionary:*

*var = { } .*

*/\*Ex:*

*>>> aplha = {'a':"apple",'b':"bad",'c':"cow"}*

*>>> type(alpha)*

*>>> type(aplha)*

*<type 'dict'>1*

*>>> aplha["b"]*

*'bad'*

*>>> aplha["b"]="ball" //replaced with bad to ball*

*>>> aplha*

*{'a': 'apple', 'c': 'cow', 'b': 'ball'}*

*>>> type(aplha)*

*<type 'dict'>*

*>>> aplha["b"]*

*'bad'*

*>>> aplha["b"]="ball"*

*>>> aplha*

*{'a': 'apple', 'c': 'cow', 'b': 'ball'}*

*>>>aplha["b"]=["ball","bag","bat"] //can keep multiple values.*

*>>> aplha*

*{'a': 'apple', 'c': 'cow', 'b': ['ball', 'bag', 'bat']}*

*>>> aplha["b"]*

*['ball', 'bag', 'bat']*

*>>> aplha["b"][1]*

*'bag'*

*/\*Ex: //nested list.*

*>>> a=[[2,3,4,5],[5,6,7,8]]*

*>>> a[1][2]*

*7*

*>>> dir(aplha)*

*['\_\_class\_\_', '\_\_cmp\_\_', '\_\_contains\_\_', '\_\_delattr\_\_', '\_\_delitem\_\_', '\_\_doc\_\_', '\_\_eq\_\_', '\_\_format\_\_', '\_\_ge\_\_', '\_\_getattribute\_\_', '\_\_getitem\_\_', '\_\_gt\_\_', '\_\_hash\_\_', '\_\_init\_\_', '\_\_iter\_\_', '\_\_le\_\_', '\_\_len\_\_', '\_\_lt\_\_', '\_\_ne\_\_', '\_\_new\_\_', '\_\_reduce\_\_', '\_\_reduce\_ex\_\_', '\_\_repr\_\_', '\_\_setattr\_\_', '\_\_setitem\_\_', '\_\_sizeof\_\_', '\_\_str\_\_', '\_\_subclasshook\_\_', 'clear', 'copy', 'fromkeys', 'get', 'has\_key', 'items', 'iteritems', 'iterkeys', 'itervalues', 'keys', 'pop', 'popitem', 'setdefault', 'update', 'values', 'viewitems', 'viewkeys', 'viewvalues']*

***Clear:***

*>>> aplha.clear()*

*>>> aplha*

*{}*

***Copy:***

*>>> alpha.copy()*

*{'a': 'apple', 'c': 'cat', 'b': 'bat'}*

***Keys:***

*>>> alpha.keys()*

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

***Pop:***

*>>> alpha.pop('a')*

*'apple'*

***Popitem:***

*>>>alpha.popitem ()*

*('c', 'cat')*

***Zip OR Merge:***

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

*>>> b=[2,3,4]*

*>>> zip(a,b)*

*[('a', 2), ('b', 3), ('c', 4)]*

*/\*Ex:*

*>>> dict(zip (a,b))*

*{'a': 2, 'c': 4, 'b': 3}*

***Operators:***

***Arthimatic operator:***

***+ , - , % , \* , / , //(lower division),\*\* (power operator).***

***Floor value:***

*>>> 3//2*

*1*

***Division:***

*>>> 3/3*

*1*

*integer/integer is integer only.( version2.7)*

*integer/integer is float . ( version 3.2)*

***Power values:***

*>>> 3\*\*5 in maths it is like (3)^5*

*243*

***Modules:***

*>>> 3%2*

*1*

***Relational Operator: used to find the rel'n ship b/w 2 values.***

*< ,>,<= ,>=, == ,!= ,<>(i,e in ver 3.2 it is removed).*

***Results is given in boolean values.***

***Equals Operator:***

*>>> 2==2*

*True*

***Greater than or equals:***

*>>> 3>=2*

*True*

*>>> 2>=3*

*False*

***less than or equals:***

*>>> 3<=2*

*False*

*>>> 2<=3*

*True*

***Greater than:***

*>>> 3>2*

*True*

*>>> 2>3*

*False*

***Less than:***

*>>> 2<3*

*True*

*>>> 3<4*

*True*

*>>> 3<2*

*False*

***Not equals:***

*>>> 2!=3*

*True*

*>>> 2!=2*

*False*

***Same as not equals:***

*>>> 2<>3*

*True*

***Assignment operators:***

*The operators which are used to assign the values from the task of execution into variable is called as* ***Assignment operators.***

*Assignment operators are:*

*= ,+= , -= , \*= , /= , %= , //= , \*\*=*

*This is the abrivation of var = var + value.( OR )var+=value.*

*/\*Ex:*

*>>> a=40*

*>>> b=50*

*>>> a+=b*

*>>> print a*

*90*

***Logical operators:***

***and****--->when both cond'n is true o/p will be true.*

***Cond'ns:***

*>>> True* ***and*** *True*

*True*

*>>> True* ***and*** *False*

*False*

*>>> False* ***and*** *True*

*False*

*>>> False and False*

*False*

***or****--->when any one cond'n is true o/p will be true.*

*whenever i want the either of the cond'n to be true then i am going use logical operator.*

***Cond'ns:***

*>>> True or True*

*True*

*>>> True or False*

*True*

*>>> False or True*

*True*

*>>> False or False*

*False*

*/\*Ex:*

*>>> a>b or a==b*

*False*

*/\*Ex:*

*>>> b*

*20*

*>>> a*

*20*

*>>> a>b or a==b*

*True*

***not****--->compliment the cond'n ( i,e : reverse).*

***Cond'ns:***

*>>> not True*

*False*

*>>> not False*

*True*

*/\*Ex:*

*>>> not a>b*

*True*

*/\*Ex:*

*>>> not a<b*

*False*

***Bit-wise Operators:***

* *The operator which performs the operations on the bits of a value ( all the values will be stored in binary each and every binary called bit).*
* *Whenever the bits in same position are group then it is going to return true into the result of that bit place.*
* *The operator which is used is single ampersand.*

*/\*Ex:*

*>>> 8 & 6*

*0*

*/\*Ex;*

*>>> 8&9*

*8*

***Bit-wise Or:***

*Bit-wise or is going to perform the OR operation or the bits which are present in the same position of different value .The symbol which is Single pipe line ( | ).*

*/\*Ex:*

*>>> 8 | 9*

*9*

*/\*Ex:*

*>>> 8 | 6*

*14*

***Bit-wise not:****It is used to covert a signed number into unsigned number OR unsigned number into signed number.*

***Formula of bit-wise:****~(n)= -(n+1)*

*~(-n)= -(-n+1)*

***/\*Ex:***

*>>>****~****8*

*-9*

*/\*Ex:*

*>>> a=20*

*>>> ~a*

*-21*

*/\*Ex:*

*>>> ~60*

*-61*

*>>> ~(-11)*

*10*

***XOR Operator:****When ever both the values are true or When ever both the values are false then the o/p i am going to retuen as false Otherwise the o/p as true.*

*11---->0*

*00----->0*

*10----->1*

*01------>1*

*/\*Ex:*

*>>> 8^9*

*1*

*/\*Ex:*

*>>> 8^6*

*14*

***Right Shift Operation( >>) :****Shift operators are used to shift values which is present in binary format towords left or towords right.To shift the bits to right the we will use ' >> ' if wanted to shift then we use ' << ' operator.*

*/\*Ex:*

*>>> 8>>2*

*2*

*/\*Ex:*

*>>> 16>>2*

*4*

*/\*Ex:*

*>>> 5>>2*

*1*

*/\*Ex:*

*>>> 15>>2*

*3*

*/\*Ex:*

*>>> 8<<3*

*64*

*/\*Ex:*

*>>> 15<<2*

*60*

***MemberShip Operator:****Whenever the user wanted to check wheather the given data present in the group or not then we are going to use membership operator.*

***Types of membership operator:***

* *in*
* *not in.*

***Syntax:****data* ***in*** *group\_of\_values.*

*/\*Ex:*

*>>> a="hello"*

*>>> 'a' in a*

*False*

*/\*Ex:*

*>>> a="hello"*

*>>> 'e' in a*

*True*

***Syntax:***

***/\*Ex:***

*>>> a="hello"*

*>>> 'e' not in a*

*False*

*/\*Ex:*

*>>> 'a' not in a*

*True*

*/\*Ex:*

*>>> 2 in [2,3,4,5]*

*True*

*/\*Ex:*

*>>> 2 not in[2,3,4,5]*

*False*

*/\*Ex:*

*>>> 2 not in(2,3,4,5)*

*False*

*/\*Ex:*

*>>> 2 in(2,3,4,5)*

*True*

*/\*Ex:*

*>>> 2 in{2,3,4,5}*

*True*

*/\*Ex:*

*>>> 2 not in{2,3,4,5}*

*False*

*/\*Ex:*

*>>> not (2 in [2,3,4,5]) # 2 not in [2,3,4,5]*

*False*

*/\*Ex:*

*>>> 2,3,4,5 in [2,3,4,5,6,7,8]*

*(2, 3, 4, True)*

*/\*Ex:*

*>>> 'a','b','c' in "abcd"*

*('a', 'b', True)*

*/\*Ex:*

*Multiple char in group*

*>>> 2,3,4,5,6*

*(2, 3, 4, 5, 6)*

*/\*Ex:*

*>>> [2,3,4] in [[2,3,4],[2,3,4,5]]*

*True*

*/\*Ex:*

*>>> [2,3] in [[2,3,4],[2,3,4,5]]*

*False*

***Identity Operators:*** *Identity operators are used to check wheather both the elements are pointing to the same memory location are not.*

***Syntax:****element* ***is*** *elemwnt2.*

***Types of identity Operators:***

* *is*
* *not is*

*>>> a=[2,3,4]*

*>>> b=a*

*>>> a*

*[2, 3, 4]*

*>>> b*

*[2, 3, 4]*

*>>> a[0]="raj"*

*>>> a*

*['raj', 3, 4]*

*>>> b*

*['raj', 3, 4]*

*>>> a[0]=2*

*>>> c=['raj',3,4]*

*>>> c*

*['raj', 3, 4]*

*>>> a*

*[2, 3, 4]*

*>>> b*

*[2, 3, 4]*

*>>> c*

*['raj', 3, 4]*

*>>> a is b*

*True*

*>>> a is c*

*False*

*>>> a="dinga"*

*>>> b*

*[2, 3, 4]*

*>>> a is b*

*False*

*/\*Ex:*

*>>> aa=b*

*>>> aa*

*[2, 3, 4]*

*>>> b*

*[2, 3, 4]*

*>>> c*

*['raj', 3, 4]*

*>>> aa[0]="raju"*

*>>> b*

*['raju', 3, 4]*

*>>> aa*

*['raju', 3, 4]*

*/\*Ex:*

*>>> a is not b*

*True*

*>>>aa is not b # not(aa is b)*

*False*

***Control Statements:***

***If Statement:***

***Synatx:*** *if Condition:*

*/\*Ex:*

*>>> year = 2000*

*>>> if (year==2000):*

*print "true"*

***o/p:***

*true*

***else statement:***

***Syntax:****if Condition:*

*statements......*

*else:*

*statement2...*

***/\*Ex:***

*>>> year=2000*

*>>> if year%2==0:*

*print "year is leap"*

*else:*

*print "year is not leap"*

***o/p:***

*year is leap*

*/\*Ex:*

*>>> intrested ='java'*

*>>> if (intrested=='java'):*

*print "Take java"*

*print "Read java"*

*else:*

*print "take C#"*

*print "Read C#"*

*print "Continue class"*

***o/p:***

*Take java*

*Read java*

*/\*Ex:*

*>>>intrested ='java'*

*>>> if (intrested=='C#'):*

*print "Take java"*

*print "Read java"*

*else:*

*print "take C#"*

*print "Read C#"*

***o/p:***

*take C#*

*Read C#*

*/\*Ex:*

*>>> a,b=20,30*

*>>> if (a>b):*

*print ("a is greater")*

*else:*

*print ("b is greater")*

***o/p:***

*b is greater*

*/\*Ex:*

*>>> if (a>b):*

*print ("a is greater",a)*

*else:*

*print ("b is greater",b)*

***o/p:***

*('b is greater', 30)*

***if else ladder(statement):***

***Synatx:****if Condition:*

*statement........*

*elif:*

*statement2............*

*else:*

*statement3..............*

***/\*Ex:***

*>>> if(var==24):*

*print " not equals"*

*else:*

*if(var ==34):*

*print "equals"*

*else:*

*print "not equals??????"*

***o/p:***

*equals*

*/\*Ex:*

*>>> var2=9*

*>>> if(var2>9):*

*print " not greater than var2",var2*

*else:*

*if(var2==9):*

*print "equals",var2*

*else:*

*print "not equals??????",var2*

***o/p:***

*equals 9*

*/\*Ex:*

*>>>var2=9*

*>>> if(var2>9):*

*print " not greater than var2",var2*

*elif(var2==9): # real format of python*

*print "equals",var2*

*else:*

*print "not equals??????",var2*

***o/p:***

*equals 9*

*/\*Ex:*

*>>> a,b,c=20,30,40*

*>>> if(a>b and a>c):*

*print "a is greater"*

*elif b>c:*

*print "b is greater"*

*else:*

*print "c is greater"*

***o/p:***

*c is greater*

***Nested if:***

***/\*Ex:***

*>>> a,b,c=20,25,22*

*>>> if a>b:*

*if a>c:*

*print "a is greater "*

*else:*

*print "c is greater"*

*elif(b>c):*

*print "b is greater"*

*else:*

*print "c is greater"*

***o/p:***

*b is greater*

*/\*Ex:*

*>>>a,b,c,d,e=20,7,34,54,9*

*>>> if(a>b):*

*if(a>c):*

*if a>d:*

*if a>e:*

*print "a is greater"*

*else:*

*print "e is greater"*

*else:*

*if d>e:*

*print "d is greater"*

*else:*

*print "e is greater"*

*else:*

*if c>e:*

*if c>d:*

*print "c is greater"*

*else:*

*print "d is greater"*

*else:*

*if d>e:*

*print "d is greater"*

*else:*

*print "e is greater"*

*elif b>c:*

*if b>d:*

*if b>e:*

*print "b is greater"*

*else:*

*print "e is greater"*

*else:*

*if d>e:*

*print "d is greater"*

*else:*

*print "e is greater"*

*elif c>d:*

*if c>e:*

*print "c is greater"*

*else:*

*print "e is greater"*

*elif d>e:*

*print "d is greater"*

*else:*

*print "e is graeter"*

***o/p:***

*d is greater*

***Looping Statement:***

*3 types of looping statement:*

* *for*
* *for each*
* *while*

***For:***

***Syntax:****for<variable > in <sequence>*

*/\*Ex:*

*>>> i =2*

*>>> for i in range (2,9):*

*print i*

***o/p:***

*2*

*3*

*4*

*5*

*6*

*7*

*8*

*/\*Ex:*

*>>> for i in range (0,5):*

*print i*

***o/p:***

*0*

*1*

*2*

*3*

*4*

***/\*Ex: for loop with even number:***

*>>> for i in range (0,8,2):*

*print i*

***o/p:***

*0*

*2*

*4*

*6*

*/\*Ex:****for loop with odd numbers***

*>>> for i in range (1,8,2):*

*print i*

***o/p:***

*1*

*3*

*5*

*7*

***/\*Ex: string looping statement***

*>>> for i in "RAJ":*

*print i*

***o/p:***

*R*

*A*

*J*

***/\*Ex: for loop with in list with a delay time 2seconds***

*>>> import time*

*>>> for i in [2,34,5,4,3,45]:*

*print i*

*time.sleep (2)*

***o/p:***

*2*

*34*

*5*

*4*

*3*

*45*

*To check value is present in group.*

*while a[i] in range(stpt,endpt,update):*

*In 2.x version*

*print (var,var2,var3.............varn),*

*In 3.x version*

*print(var,var2,var3..........varn,end="")*

*write a program to print pattern*

***while :***

***Syntax:****while Condition:*

***Data Structure:***

***Bubble sort:***

*#from sre\_constants import RANGE*

*a=[34,5,2,6,4,78,43]*

*n=len(a)*

*print "length of elements is:",len(a)*

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

*for j in range (0,n-i):*

*if(a[j]>a[j+1]):*

*temp = a[j]*

*a[j]=a[j+1]*

*a[j+1]=temp*

*print ("pass number",i)*

*print a*

***Selection sort:***

*#from sre\_constants import RANGE*

*a=[34,5,2,6,4,78,43]*

*n=len(a)*

*print "length of the elements",len(a)*

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

*min=i-1*

*for j in range (i,n):*

*if(a[min]>a[j]):*

*min=j*

*temp = a[i-1]*

*a[i-1]=a[min]*

*a[min]=temp*

*print ("pass number",i)*

*print a*

***Python Programming in eclipse:***

*open in version eclipse neon 3.0*

*Goto help-->Eclipse marketplace-->enter pydev--->install-->conform-->accept-->conform-->restart the Eclipse*

*Open-->new-->project-->pydev-->pydev project-->next*

*Give project name--->click plese configure-->click on finish*

*Right click on project name-->new-->package name -->prgram name-->finish.*

*ok*

*ok*

***Binary:***

***Taking the dynamic input from the user by using input statement:***

* ***input()***
* ***raw\_input()***

***Input statement:(2.7 or 2.x input statements)***

*A function which is used to take an input from the user and returning the input as the user specified or returning the input without modifying parent type or earlier type is done by using input function.*

*Syntax: var=input( statements for user )*

*gudieness/statements for user is nothing but a statement which is displaying to the user to save that why cursor is waiting.*

***Raw\_input:*** *it is a function which is used to take an input from the user and the input will be coverted into the string at the time of returning.*

*Syntax: var = raw\_input( guideness to the user)*

*because of the redundancy the memory wasteage the raw i/p function is eliminated insted a modification for i/p fun'n as been happen like before the entered format was same as the recived format but in version 3.x the enterd format is been converted into string format.*

*Syntax:a=input("guideness to the user")*

*/\*Ex:*

*>>> input()*

*9*

*9*

*Type Casting :the process of coverting a value from one data type into another data type is called as type casting.*

*Syntax: data type ( value /var)*

*it is not going to modify the actual value.*

*to convert the value which we are getting from the function we use a syantax as var=datatype (i/p (instruction to user) )*

*var = datatype(var)*

*the datatypes are :*

*int,float,boolen,str,list,tuple,set.*

***Taking dynamic input for a tuple is been prohibited.***

*Because tuple is immutable.*

***Taking dynamic input for a list:***

*a=[]*

*n=int(input ("enter the number of elements"))*

*print ("enter the elements into the list:")*

*for i in range (0,n):*

*a.append(int(input()))*

*print a*

*o/p:*

*enter the number of elements5*

*enter the elements into the list:*

*20*

*30*

*40*

*50*

*60*

*[20, 30, 40, 50, 60]*

***Taking dynamic i/p for a set: To add an element into the set we use a fun'n called add.***

*a=set()*

*#b=[]*

*n= int (input ("enter the number of elements:"))*

*print ("Enter the elements into the list:")*

*for i in range(0,n):*

*a.add(int(input ()))*

*#b.extends(b)*

*print a*

*o/p:*

*enter the number of elements:3*

*Enter the elements into the list:*

*4*

*5*

*6*

*set([4, 5, 6])*

***Taking dynamic input element into dictionary:***

***write a program to count the number of values which are present in a group items:***

*a=[2,3,4,5,6,7,8,9,0]*

*b="marry went to marketing "*

*count=0*

*for i in b:*

*print i*

*count+=1*

*print count*

*o/p:*

*m*

*a*

*r*

*r*

*y*

*w*

*e*

*n*

*t*

*t*

*o*

*m*

*a*

*r*

*k*

*e*

*t*

*i*

*n*

*g*

*24*

***Sequence of elements:***

*Syntax:[i for i in range(1,n)]*

*>>> a=[m for m in range (2,22)]*

*>>> a*

*[2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21]*

*>>> a=[m for m in "yashwanth"]*

*>>> a*

*['y', 'a', 's', 'h', 'w', 'a', 'n', 't', 'h']*

***Cartation value:***

*>>> a=[(m,n)for m in range (1,5) for n in "yashwanth"]*

*>>> a*

*[(1, 'y'), (1, 'a'), (1, 's'), (1, 'h'), (1, 'w'), (1, 'a'), (1, 'n'), (1, 't'), (1, 'h'), (2, 'y'), (2, 'a'), (2, 's'), (2, 'h'), (2, 'w'), (2, 'a'), (2, 'n'), (2, 't'), (2, 'h'), (3, 'y'), (3, 'a'), (3, 's'), (3, 'h'), (3, 'w'), (3, 'a'), (3, 'n'), (3, 't'), (3, 'h'), (4, 'y'), (4, 'a'), (4, 's'), (4, 'h'), (4, 'w'), (4, 'a'), (4, 'n'), (4, 't'), (4, 'h')]*

***Format function:***

*>>> a=[m for m in range (1,5)]*

*>>> print ("my name is {0} and my age is {1}".format("yashwanth",23))*

*my name is yashwanth and my age is 23*

*>>> print ("my name is {1} and my age is {0}".format("yashwanth",23))*

*my name is 23 and my age is yashwanth*

*>>> "{0}{1}".format("yash","yash")*

*'yashyash'*

*>>>a="yash"*

*>>> b="yash"*

*>>> a,b*

*('yash', 'yash')*

*>>> a+b*

*'yashyash'*

*>>> "{} {}".format ("dinga","kumar")*

*'dinga kumar'*

*>>> "{1} {0}".format ("dinga","kumar")*

*'kumar dinga'*

***write a program to join 2 strings without using +operator:***

*>>> a="{} {}".format ("ABC","yash")*

*>>> a*

*'ABC yash'*

*>>> a="{} {} {}".format ("ABC","yash","ROCKSTAR")*

*>>> a*

*'ABC yash ROCKSTAR'*

***Functions:***

*Functions are the group of instructions which are designed to perform a specific appl'n OR which is designed to perform specific task.*

* *Built in function.*
* *user defined function.*

***Advatages:***

*Code reuseability.*

*Syntax:def nameofFunction(parmeter,parameter2,...........parametern)*

***Priority of Execution:***

*The first priority is given for the statement which are written outside the function. OR*

*The first priority is given to the statement which doesn't belongs to any purticular block.*

*The 2nd priority is given to the fun'n which is called for the first time.*

*We can take n return tpye value.*

*/\*Ex:*

*print ("Out of block")*

*def a():*

*print ("inside function")*

*print ("Out of block2")*

*a()*

*print ( " post extcution")*

***o/p:***

*Out of block*

*Out of block2*

*inside function*

*post extcution*

*/\*Ex:*

*print ("Out of block")*

*def a():*

*print ("inside function")*

*print ("Out of block2")*

*#a() // not specified the caller function*

*print ( " post extcution")*

***o/p:***

*Out of block*

*Out of block2*

*post extcution*

***Function without parameter and without return type:***

*def main():*

*a=[2,3,4,5,6,7,8,9]*

*print a*

*print (a[::-1])*

*rajakumara()*

*def rajakumara():*

*print ("rajakumara")*

*o/p:*

*>>> main()*

*[2, 3, 4, 5, 6, 7, 8, 9]*

*[9, 8, 7, 6, 5, 4, 3, 2]*

*rajakumara*

*/\*Ex:(dounder(double underscore) )*

*def main():*

*a=[2,3,4,5,6,7,8,9]*

*print a*

*print (a[::-1])*

*rajakumara()*

*def rajakumara():*

*print ("rajakumara")*

*print("\*\*\*\*\*\*")*

*if \_\_name\_\_=="\_\_main\_\_":*

*main()*

*o/p:*

*\*\*\*\*\*\**

*[2, 3, 4, 5, 6, 7, 8, 9]*

*[9, 8, 7, 6, 5, 4, 3, 2]*

*rajakumara*

***\_\_name\_\_:***

*This type of variables are called as dounder(double underscore) variable.These are variable which will have a pre-defined value which is set by developer .the value of \_\_name\_\_ is\_\_main\_\_.*

*The value of dounder name will be the name of the file when the file is invoked from the other file or the file is imported from other file.*

***Function without parameter with return value:***

***/\*Ex:***

*def wpwr():*

*a,b=20,30*

*return a+b*

*o/p:*

*>>> wpwr()*

*50*

***String replication:***

***/\*Ex:***

*>>> 'hello'\*5*

*'hellohellohellohellohello'*

*/\*Ex:*

*>>> 'hello '\*5*

*'hello hello hello hello hello '*

*/\*Ex:*

*>>> a='yash'*

*>>> a\*3*

*'yashyashyash'*

***Pattern programming:***

*/\*Ex:*

*>>> print ("\*"\*5)*

*\*\*\*\*\**

*/\*Ex:*

*for i in range (1,5):*

*print ("\*"\*5)*

*o/p:*

*\*\*\*\*\**

*\*\*\*\*\**

*\*\*\*\*\**

*\*\*\*\*\**

*/\*Ex:*

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

*print (str(i)\*5)*

*o/p:*

*11111*

*22222*

*33333*

*44444*

*55555*

*/\*Ex:*

*ch=ord('a')*

*for i in range ( 0,5):*

*print (chr(ch+i)\*4)*

*o/p:*

*aaaa*

*bbbb*

*cccc*

*dddd*

*eeee*

*/\*Ex:*

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

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

*print "\*",*

*print*

*o/p:*

*\**

*\* \**

*\* \* \**

*\* \* \* \**

*\* \* \* \* \**

*/\*Ex:*

*for i in range (2,7):*

*print (str(i)\*i)*

*o/p:*

*22*

*333*

*4444*

*55555*

*666666*

*/\*Ex:*

*ch=ord('a')*

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

*print (chr(ch+i)\*i)*

*o/p:*

*b*

*cc*

*ddd*

*eeee*

*fffff*

*/Ex:*

*ch=ord('a')*

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

*print (" "\*(i-1)),*

*print ("\*"\*(6-i))*

*o/p:*

*\*\*\*\*\**

*\*\*\*\**

*\*\*\**

*\*\**

*\**

*/\*Ex:*

*ch=ord('a')*

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

*print (" "\*(i-1)),*

*print (chr(ch+i-1)\*(6-i))*

*o/p:*

*aaaaa*

*bbbb*

*ccc*

*dd*

*e*

*/\*Ex:*

*ch=ord('a')*

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

*print (" "\*(i-1)),*

*print (str(i)\*(6-i))*

*o/p:*

*11111*

*2222*

*333*

*44*

*5*

*/\*Ex:*

*ch=ord('a')*

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

*print (" "\*(i-1)),*

*print ("\*")*

*o/p:*

*\**

*\**

*\**

*\**

*\**

*/\*Ex:*

*ch=ord('a')*

*n=5*

*for i in range (1,n+1):*

*print (" "\*(6-i-1)),*

*print ("\*")*

*o/p:*

*\**

*\**

*\**

*\**

*\**

*/\*Ex:*

*ch=ord('a')*

*n=5*

*for i in range (1,n+1):*

*print (" "\*(6-i-1)),*

*print ("\*"\*i)*

*o/p:*

*\**

*\*\**

*\*\*\**

*\*\*\*\**

*\*\*\*\*\**

*/\*Ex:*

*ch=ord('a')*

*n=5*

*for i in range (1,n+1):*

*print (" "\*(6-i-1)),*

*print ("\* "\*i)*

*o/p:*

*\**

*\* \**

*\* \* \**

*\* \* \* \**

*\* \* \* \* \**

*/\*Ex:*

*ch=ord('a')*

*n=5*

*for i in range (1,n+1):*

*print (" "\*(6-i-1)),*

*print ("\*"\*i)*

*o/p:*

*1*

*22*

*333*

*4444*

*55555*

*/\*Ex:*

*ch=ord('a')*

*n=5*

*for i in range (1,n+1):*

*print (" "\*(n-i)),*

*print (chr(ch+i-1)\*i)*

*o/p:*

*a*

*bb*

*ccc*

*dddd*

*eeeee*

*/\*Ex:*

*ch=ord('a')*

*n=5*

*for i in range (1,n+1):*

*print ("\*"\*(n-i+1))*

*o/p:*

*\*\*\*\*\**

*\*\*\*\**

*\*\*\**

*\*\**

*\**

*/\*Ex:*

*ch=ord('a')*

*n=5*

*for i in range (1,n+1):*

*print (str(i)\*(n-i+1))*

*o/p:*

*11111*

*2222*

*333*

*44*

*5*

*/\*Ex:*

*ch=ord('a')*

*n=5*

*for i in range (1,n+1):*

*print (chr(ch+i-1)\*(n-i+1))*

*o/p:*

*aaaaa*

*bbbb*

*ccc*

*dd*

*e*

*/\*Ex:*

*ch=ord('a')*

*n=5*

*for i in range (1,n+1):*

*print (" "\*(n-i)),*

*print ("/"),*

*print (" "\*(i-1)),*

*print (" "\*(i-1)),*

*print ("\\")*

*for o/p run the program.*

***Recursive function:****A function call itself until some particular condition then it is called as Recursive statement.*

*Factorial and Recursive programs are done ..refer program.*

***Objected Orinted Principle:***

***Class:***

* *Class is a blueprint which describes the property of an entity.*
* *Class is a user defined datatype through which we can create variables those variables are called as object.*
* *Class does'nt require any memory space.*

***Object(Instance of a class):***

*Object is an instance of a class or we can say Object is a real word entity which describes the properties which are mentioned inside a class.*

*To create a class we use the syntax as:*

*class classname:*

*data members*

*or*

*class members*

*member functions*

*/\*Ex:*

*class name:*

*a=20*

*b=30*

*c=40*

*print (name.a)*

*obj=name()*

*print(obj.a)*

*#obj.a=240*

*print "After modification using obj"*

*print ("name.a",name.a)*

*print ("obj.a",obj.a)*

*name.a=300*

*print "After modification using class"*

*print ("name.a",name.a)*

*print ("obj.a",obj.a)*

***o/p:***

*20*

*20*

*After modification using obj*

*('name.a', 20)*

*('obj.a', 20)*

*After modification using class*

*('name.a', 300)*

*('obj.a', 300)*

***Constructors:***

* *Constructor is a special function which is used mainly to inititized the value and it is the first function which will be executed when you create a instance of a class.*
* *Under constructor we use a member function we say constructor is a function which is present inside a class.*
* *The name of a constructor is (double underscore init dunder init).*
* *All the member function must have one argument that is self.*
* ***Syntax :*** *def \_\_init\_\_(self,arguments):*

*body of constructor.*

* *In python None=null*
* *none specified no value is present.*
* *It should not have any return value.*

***Usage of Self:***

* *Self is similar to this keyword in java.*
* *Self is used for point out the current Object.*
* *objname.fname(arg)*
* *classname.fname(objname,arg)*

***Object Oriented principles:***

*In python we can achive only one object oriented externally remaining all the three object oriented principle like polymorphism, encapsulation, abstraction will be done internally by the application and one and only principle we can do externally is inheritance.*

*All types of inheritance is possible in python including multiple inheritance also.*

***Inheritence:***

*It is a phenomenon of getting the properties of one class inside the other class is known as* ***Inheritence.*** *(OR) The process of inheriting the feature of one class inside the other class is called as inheritance.*

***Single inheritance:***

*/\*Ex:*

*class A:*

*s=20*

*y=30*

*b=45*

*def add(self):*

*return self.s+self.y+self.b*

*class B(A):*

*d=67*

*r=34*

*obj=B()*

*obj2=A()*

*print (obj.s)*

*A.s=40*

*print(obj.s)*

*print (obj.add())*

*print (B.add(obj))*

*print (obj2.add())*

***For output run the program.***

***Multilevel Inhertance:***

*If the inheritance happens in levels like* ***A*** *is inherited to* ***B*** *,****B*** *is inherited* ***C****, and* ***C****is inherited to* ***D*** *so on if it happens like that so on then we use a concept of multilevel inheritence.*

***Multiple inheritance:***

*Whenever a class is inherited from multiple classes then that kind of inheritance is called as multiple inheritance.*

*Here, one child n parent.*

*Ex:*

*class A:*

*a=20*

*def \_\_init\_\_(self,b,c): #this line is called as Constructors in python*

*self.b=b*

*self.c=c*

*def add(self):*

*return self.a+self.b+self.c*

*class B(A):*

*d=67*

*a=23,4*

*def \_\_init\_\_(self,e):*

*self.e=e*

*class C(A,B):*

*def \_\_init\_\_(self,b,c,e,f):*

*self.f=f*

*A.\_\_init\_\_(self,b,c)*

*B.\_\_init\_\_(self,e)*

*obj=C(34,56,3,243)*

*print (obj.a)*

*print(obj.d)*

*print(obj.c)*

*print(obj.d)*

*print(obj.f)*

*print (obj.e)*

***Hirechical inheritence:***

*Whenever baseClass is inherited to multiple childclass then we call it has*

***hirechical inheritance.***

*For example refer program.*

***Hybrid inheritence:***

*It is a type of inheritence which all the kinds d/f inheritence are combined.*

***Method Overriding:***

*Whenever there are 2 classes* ***a*** *and* ***b*** *where* ***b*** *is inherited from* ***a*** *and both will have the method with the same name then if you call that method with an instance of* ***b*** *then the method which is there instance of* ***b*** *will executed because the programing language follows the principle of shortest execution*

*/\*Ex:*

*class A:*

*a=20*

*def disp(self):*

*print "disp is running"*

*class B(A):*

*obj2=A()*

*def disp(self):*

*print "Overridden"*

*# print "b value is ",b*

*obj=B()*

*#obj.disp()*

*#A.disp(obj)*

*obj.obj2.disp()*

*If you wanted to get* ***a*** *method prestent in* ***a*** *either you can call with help of class name or you can create a has a relationship.*

***Class methods:*** *The methods which are belongs to the class and which are used to modify the actual class members is called as class methods. For class methods we must pass a class as an arguments and that arguments represented as CLS.*

*CLS is similar to the self but for self we pass object address and for CLS we pass address of class as an arguments.*

*/\*Ex:*

*class A:*

*a=20*

*b=30*

*@classmethod*

*def swap(cls):*

*cls.a,cls.b=cls.b,cls.a*

*print "Before swaping"*

*print (A.a,A.b)*

*print "After swaping"*

*A.swap()*

*print (A.a,A.b)*

*print A*

***o/p:***

*Before swaping*

*(20, 30)*

*After swaping*

*(30, 20)*

*\_\_main\_\_.A*

***Note:static method will not have any rel'nship b/w class data or object data.***

*/\*Ex:*

*class A:*

*a=20*

*b=30*

*@classmethod*

*def swap(cls):*

*cls.a,cls.b=cls.b,cls.a*

*@staticmethod*

*def disp(): # if we donot specify any static method then*

*#we should pass one argument either self or anything*

*print "static method is running"*

*A.disp()*

***Output:***

*static method is running*

***Import data from one class to another class:***

*/\*Ex:*

*from static import A*

*obj=A()*

*print (obj.a,obj.b)*

*obj.swap()*

*print (obj.a,obj.b)*

*print (A.a,A.b)*

***Output:***

*static method is running*

*(20, 30)*

*(30, 20)*

*(30, 20)*

*Another way to import:*

*/\*Ex:*

***import static:***

*obj=static.A()*

*print (obj.a,obj.b)*

*obj.swap()*

*print (obj.a,obj.b)*

***Output:***

*static method is running*

*(20, 30)*

*(30, 20)*

***Aliasing:***

*/\*Ex:*

*import static as st*

*obj=st.A()*

*print (obj.a,obj.b)*

*obj.swap()*

*print (obj.a,obj.b)*

***Output:***

*static method is running*

*(20, 30)*

*(30, 20)*

***Generating list of values by using the for loop in a Sequential manner:***

***Lambda Functions:***

*Functions without the specific name and which is going to do or perform the given task in one line of its explantion this kind of function is called as Lambda function .All the lambda function starts with a keyword Lambda parameters :operation*

*>>>a=lambda :print ("hello")*

*>>>a()*

*hello*

*/\*Ex:*

*>>> a=lambda x: x+2*

*>>> a(20)*

*22*

*/\*Ex:*

*>>> a=lambda x,y=30:x+y*

*>>> a(45)*

*75*

*/\*Ex:*

*>>> a=lambda x,y=30:x+y*

*>>> a(34,65)*

*99*

***Passing function as an argument:***

***/\*****Ex:*

*def add(x,y):*

*return x+y*

*def sub(x,y):*

*return x-y*

*def fx(f,x,y): # f==>name of function*

*z=(f(x,y))*

*print (z)*

***Output:***

*>>> fx(add,30,45)*

*75*

*The phenomenon of a we are going to pass the name of the function as arguments is called as functional programming basically when we pass the name of the function an argument and if you say y=f(x) automatically the name of the function what we have passed will get executed.*

*y=f(x)*

***Iterator:***

*Iterator is a process of producing the values in the sequence from the group is*

*Known as iteration and the function which performs the iteration is called is iterator.*

*/\*Ex:*

*>>> a=[2,3,4,5,6]*

*>>> i=iter(a)*

*>>> next(i)*

*2*

*>>> next(i)*

*3*

*>>> next(i)*

*4*

*>>> next(i)*

*5*

*>>> next(i)*

*6*

*Stop the iterator all the values are iterated.*

*Instead of for loop like “for” i in range a” use can perform iterator.*

***User defined iterator:***

*User defined iterator is created by using the function name “\_\_iter\_\_” and “\_\_next\_\_”.*

*/\*Ex:*

*class powthree:*

*def \_\_init\_\_(self,max):*

*self.max=max;*

*def \_\_iter\_\_(self):*

*self.n=0*

*return self*

*def \_\_next\_\_(self):*

*if self.n<=self.max:*

*res=3\*\*self.n*

*self.n+=1*

*return res*

*print "Out of bound"*

***For output: we have to type in shell.***

*>>> a=powthree(3)*

*>>> i=iter(a) #i=a.\_\_iter\_\_()*

*>>> i.\_\_next\_\_()*

*1*

*>>> i.\_\_next\_\_()*

*3*

*>>> i.\_\_next\_\_()*

*9*

*>>> i.\_\_next\_\_()*

*27*

*>>> i.\_\_next\_\_()*

*Out of bound*

***Iterator to produce factorial of numbers:***

*/\*Ex:*

*class powthree:*

*def \_\_init\_\_(self,max):*

*self.max=max;*

*def \_\_iter\_\_(self):*

*self.n=0*

*return self*

*def \_\_next\_\_(self):*

*if self.n<=self.max:*

*res=self.fact()*

*self.n+=1*

*return res*

*print "Out of bound"*

*def fact(self):*

*re=2*

*for i in range(2,self.n+1):*

*re\*=i*

*return re*

***Output:***

*>>> a=powthree(3)*

*>>> i=a.\_\_iter\_\_*

*>>> i=a.\_\_iter\_\_()*

*>>> i.\_\_next\_\_ ()*

*2*

*>>> i.\_\_next\_\_ ()*

*2*

*>>> i.\_\_next\_\_ ()*

*4*

*>>> i.\_\_next\_\_ ()*

*12*

*>>> i.\_\_next\_\_ ()*

*Out of bound*

***Generators:***

*Generator is a process of producing the sequence of values but getting one at a time.*

*/\*Ex:*

*def pow3(n):*

*i=0*

*while(i<=n):*

*res=fact(i)*

*i+=1*

*yield res*

*print "Out of bound"*

*def fact(n):*

*if n in (0,1):*

*return 1*

*return n\*fact(n-1)*

***Write a program using generator to print series of Fibonacci numbers by using recursion.***

***/****\*Ex:*

*def pow3(n):*

*i=0*

*while(i<=n):*

*res=fib(i)*

*yield res*

*i+=1*

*def fib(n):*

*if n==1:*

*return 0*

*elif n==2 or n==3:*

*return 1*

*return fib(n-1)+fib(n-2)*

***Diff b/w Iterators and generators***

***Generators:*** *All generators are iterators but not all iterators are generators.*

***Iterators:*** *It will traverse on the generated value or the collection and gives one value at a time.*

*Generators are used to generate or create the sequence and gives one element at a time. Basically as the name says generate the values by using generator as a function.*

***Decorators:***

***/\*Ex:***

*def a(func):*

*def inner(a,b):*

*print "i am decoder"*

*res=func(a,b)*

*print "the res of fun is",res*

*return inner*

*@a*

*def add(a,b):*

*return a+b*

*@a*

*def add(a,b):*

*return a-b*

*@a*

*def add(a,b):*

*return a/b*

*@a*

*def add(a,b):*

*return a\*b*

***o/p:***

***write a program to calculate the time taken to print the square of series as well as cube series by using phenomenon of decoder to count the time.***

*import time*

*def timein(func):*

*def inner(a,b):*

*start =time .time()*

*res=func(a,b)*

*end=time.time()*

*print ("time taken",str(end-start))*

*return inner*

*@timein*

*def sqr(start,end):*

*a=[]*

*for i in range(start,end):*

*a.append(i\*i)*

*return a*

*@timein*

*def cube(start,end):*

*a=[]*

*for i in range(start,end):*

*a.append(i\*i\*i)*

*return a*

***Unpacking:***

*/\*Ex:*

*def disp(a,b,c):*

*print (a)*

*print (b)*

*print (c)*

*a=(2,3,4)*

*b=[5,6,7]*

*c={8,9,0}*

*d="hai"*

*e={'a':"apple",'b':"ball",'c':"cat"}*

*print ("tuple unpacking")*

*disp(\*a)*

*print ("list unpacking")*

*disp(\*b)*

*print ("set unpacking")*

*disp(\*c)*

*print ("string unpacking")*

*disp(\*d)*

*print ("dictionary unpacking")*

*disp(\*e)*

***Output :***

*Refer and run program.*

***File:***

*To store the data and retrieve the data.*

***Syntax:*** *fileobject = open(filepath,mode)*

***Mode:***

*It specifies access permission is given to the program.*

***There are 3 major mode:***

*r If we specifies a mode as r then the permission will be given only to read the data.*

*And whenever the open the file with read mode always the cursor is going to start reading from first position.*

*w This is going to specifies that we can do the right thing into the file. Even the file is not present you can able to create the file automatically when you open it with write mode. Always the cursor will start its writing when you open the file in write mode.*

*a If you open the file in append mode then you can able to read and write from the last position of the data present in a file.* ***OR***

*The reading and writing will happen from the position where you have specified the last case or when you stop in the last case.*

***In read mode we have several modes;***

*R+ Specifies permission as read then write.*

*Rb opens the file in read only mode and the file is readed in the binary format.*

*Rb+Opens a file for both reading and writing in beginning of the file but only in binary format.*

***Write also support several modes:***

*W+ Opens a file for both reading after writing . If the file is not present then it will creates a new file and writes into it.*

*Wb Opens the file in write only mode but the data will be written in the binary format.*

*Wb+ Opens a file for both writing and reading in binary format.*

*Overwrites the existing file if the file exists. If the file does not exists, create a new file for reading and writing.*

***Append:***

*a Opens a file for appending. The pointer is at the end of the file if the file exists. I,e: the file is in the append mode.If the file does not exists, it create a new file for writing.*

*Ab Opens a file for appending in binary format. The file pointer is at the end of the file if the file exists. That is, the file is in the append mode. If the file does not exists, it create a new file for writing.*

*A+ Opens a file in both reading and appending. The file pointer is at the end of the file if the file exists. The file opens in the append mode. If the file does not exists, it create a new file for reading and writing.*

*Ab+ Opens a file in both reading and appending in binary format. The file pointer is at the end of the file if the file exists. The file opens in the append mode. If the file does not exists, it create a new file for reading and writing.*

*/\*Ex:*

*>>> fobj=open("jsp.txt","r")*

*Traceback (most recent call last):*

*File "<pyshell#0>", line 1, in <module>*

*fobj=open("jsp.txt","r")*

*IOError: [Errno 2] No such file or directory: 'jsp.txt'*

*>>> fobj=open("jsp.txt","w")*

*>>> fobj.close()*

*>>> fobj*

*<closed file 'jsp.txt', mode 'w' at 0x029C6078>*

*>>> fobj = open("jsp.txt","a")*

*>>> fobj.name*

*'jsp.txt'*

*>>> fobj.mode*

*'a'*

*>>> fobj.closed*

*False*

*>>> fobj.close()*

*>>>fobj.closed*

*True*

*>>> fobj = open("jsp.txt","a")*

*>>> fobj.write("hello")*

*>>> fobj.write("Hai how are you")*

*>>> fobj.close()*

*>>> fobj=open("jsp.txt","r")*

*>>> fobj.read()*

*'helloHai how are you'*

*>>> fobj.close()*

*>>> fobj=op*

*Traceback (most recent call last):*

*File "<pyshell#19>", line 1, in <module>*

*fobj=op*

*NameError: name 'op' is not defined*

*>>> fobj=open("jsp.txt","r")*

*>>> fobj.read()*

*'helloHai how are you'*

*>>> fobj.tell()*

*20L*

*>>> fobj=open("jsp.txt","a+")*

*>>> fobj.write("hai bro how are you, what are you doing")*

*>>> fobj.close()*

*>>> fobj.closed*

*True*

*>>> fobj=open("jsp.txt","a+")*

*>>> fobj.writelines(["apple","ball","\n","cat"])*

*>>> fobj.close()*

***To read one line of data.***

*/\*Ex:*

*>>> fobj=open("jsp.txt","r")*

*>>> fobj.readlines*

*<built-in method readlines of file object at 0x02E9BF40>*

*>>> fobj.readlines()*

*['hai bro how are you, what are you doingappleballcatappleball\n', 'cat']*

***Seek:***

*Seek is a function which is used to travel b/w the position .*

***Syntax:*** *fobj.seek(offset,from) ie,offsetshould be zero*

*From as given 3 position*

*From start(0)*

*From the current position (1)*

*From last position(2)*

*(0,1,2)are the values which you can given in from position which specifies from which position the cursor need to travel.*

*/\*Ex:*

*>>> f.seek(0,2)*

*>>> f.tell()*

*65L*

***OS module:***

*>>> import os*

*>>> os.mkdir("jspider")*

*>>> os.rmdir("jspider")*

*>>> os.mkdir("d:\\jspiders")*

*>>> os.getcwd()*

*'C:\\Python27'*

*>>> os.chdir("d:\\jspiders")*

*>>> os.getcwd()*

*'d:\\jspiders'*

*>>> os.mkdir("jece26")*

*>>> os.chdir("d:\\jspiders\\jece26")*

*>>>os.getcwd()*

*'d:\\jspiders\\jece26'*

*>>> os=open("file.txt","w")*

*>>> os.close ()*

***Create a file blog:***

*With is keyword which is used to create blog where you can access the data related to file or you can write the data into the file but when you come out of the blog then the operation will be restricuted. Basically with is used to perform the set of operation on a single short*

***/****\*Ex:*

*>>> with open ("file.txt","r") as f:*

*a=f.read(8)*

*b=f.read(20)*

*c=f.read(30)*

*>>> a*

*''*

*>>> b*

*''*

*>>> c*

*''*

***JSON(JAVA SCRIPT OBJECT NOTATION):***

*JSON*

***Map:***

*Map is a function used to perform operation .*

*/\*****To do addition operation in map***

*>>> map(lambda x:x+1,[2,3,4,5])*

*[3, 4, 5, 6]*

*>>> a=map(lambda x:x+1,[2,3,4,5])*

*>>> a*

*[3, 4, 5, 6]*

*>>> a=map(lambda x:x+10,[2,3,4,5])*

*>>> a*

*[12, 13, 14, 15]*

*>>> a=map(lambda x:x+10,[2,3,4,5])*

*>>> for i in a:*

*print i*

*12*

*13*

*14*

*15*

*/\*****Print sqr numbers in map***

*>>> def sqr(x):*

*return x\*x*

*>>> a=map(sqr,[2,3,4,5])*

*>>> a*

*[4, 9, 16, 25]*

***Pickle:***

*Pickle is a process of converting group of data as a single data item and when you are retrieving you need to convert the data which is present inside the file into native format.*

*The process of converting the multiple data items as a single data items is called as* ***pickling*** *and the process of re-converting from pickle object into the collection of data is called as* ***unpickling.***

*Pickling is done with the help of pickle.dumps*

***Syntax:*** *pickle.dump(the data,fileObject,pickle.HIGHEST\_PROTOCOL)*

*The process of unpickling will be done with the help of a function called as load.*

***Syntax:*** *Variable=pickle.load(fileObject)*

*When we want to store the data in pickle format then we need to open the file in .pkl extension.*

*This format is heighly secured then the json format.*