Strings

```
str11 = 'Hello';  str12 = "World"
```

Slicing:

```
str1[:], str1[4]; str1[-7]; str1[3:],
str1[:3]; str1[2:6]; str1[2:10:2]
str1[::-1];  str1[-2,-10,-1];
str1[1:-8]
boolresult  = 'A'  in 'NASDAQ'
```

Operators

```
Arithmetic: + - * /  ** // %
Logical:  and   or  not
Relational: == >  <  >=  <=
Bitwise:  &   |  ~  ^  >>=  <<=
Assignment: =  +=  -=  *= /=
Membership: in,  not in
Identity:  is, is  not
```

Decision Making

```
n = 74
if n >= 70:
        print("Distinction")
elif n >= 60 and n < 70:
        print("First Class")
elif n >= 50 and n < 60:
        print("Second Class")
elif n >= 35 and n < 50:
        print("Pass Class")
else:
        print("Fail")
```

Indexing from Start of String ---------->

0	1	2	3	4	5	6	7	8	9	10	11
A	B	C	D	E	F	G	H	I	J	K	L
-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

<----------Indexing from End of String

for loop

```
for letter in "string":
        print(letter)
for num  in range(20):
        process(num)
for element in mylist:
        print(element)
for element in mytuple:
        print(element)
for num in range(len(mylist)):
        print(mylist[num])
for key,value in user_dict.items():
        print(key,value)
for index,value in enumerate(mylist):
        print(index,value)
for num in range(10):
        print(num)
        if num == 5:
                break
for num in range(10):
        if num == 5:
                continue
        print(num)
```

Few Builtin Functions

```
usrstring = input("Enter Stock Name")
print(type(usrstring))
usrvalue = int(input("Enter Stock Value"))
numstr = str(34567)
print(numstr,  type(numstr),  len(numstr)
boolresult = isinstance(usrstring, int)
print(boolresult)
bool(0);  bool(1); bool(-1)
mytuple = tuple(mylist)
mylist  =  list(mytuple);  mylist = list('Hello')
dir(list);  help(list.append)
```

Dictionary

```
myd1 = {"Xchange":"Nifty", "Name": "Reliance", "Value":1023}
N = myd1['Name'];   myd1['Name'] = "RPL"; print(myd1)
myd1["City"] = "Mumbai";  del myd1['Value']
k = myd1.keys(); v = myd1.values(); i = myd1.items()
mytuple = ("Bank", "City", "Rank")
myd2 = dict.fromkeys(mytuple,'na')
```

Few String Functions

```
print(mystr.find('n',6,9))
mystr.startswith('N');  mystr.count('a');
mystr.lower();  mystr.endswith('ange')
mystr.swapcase();  mystr.replace('a','O');
newstr = mystr.strip(':');
newstr = mystr.lstrip(';');
newstr = mystr.rstrip();
boolresult = mystr.islower();
boolresult = "2019".isdigit()
mystrlist = mystr.split(',');
newstr = ':'.join(mystrlist)
```

Function def

```
def myfunction1(p1,p2,p3):
        s1 = p1 + p2 + p3
        return s1

def myfunction2(p1,p2,p3):
        print(p1,p2,p3)

r1 = myfunction1(2,3,4)
myfunction2('ab', 'cd', 'ef')
```

Lists

```
mylist3 = mylist1 + mylist2
mylist4 = mylist1 * 5
mylist1.append(21)
mylist1.extend([5,10,15])
mylist1.insert(3,(77,88,99)
mylist1.pop(-5); mylist1.pop()
mylist1.remove(21)
mylist1.reverse()
mylist1.sort(reverse=True)
mylist1.index(5,2,7)
mylist1.count(5)
mylist1.clear()
matrix = [[1, 2], [4, 5], [7, 8]]
ml5 = mylist1[2:6:2]
boolresult  = 'NIFTY' in mylist
```

Set

```
myset2 = {22,24,26,'a','b',46,2,6,7,24,2}
myset3 = myset1.union(myset2)
myset3 = myset1.intersection(myset2);
myset3 = myset1.symmetric_difference(myset2)
myset3 = myset1.difference(myset2)
```

Tuples

```
mytuple  = (1,2,'a',3,'b',4,5,'f')
cnt = mytuple.count(2)
idx = mytuple.index('b')
idx = mytuple.index('a',1,7)
```

print(...)

```
print(numc, index, xchange)
print("There are %d stocks in %s of %s " %(numc,index,xchange))
print("There are {} stocks in {} of {} " .format(numc,index,xchange))
print("There are {:d} stocks in {:s} of {:s} " .format(numc,index,xchange))
print("s{:10}e".format('PYTHON')); print("s{:>10}e".format('PYTHON'));
print("s{:^10}e".format('PYTHON')); print("s{:*^20}e".format('PYTHON'))
for num in range(10):
        print(num,num+1,end=':',sep='-')
```

while loop

```
while (condition == True):
        Process_here

lcount = 10
while(lcount > 0):
        print(lcount)
        lcount = lcount -  1
```

Map, Filter, Reduce

```
feetlist = [10,20,30,40,50,60]
inchlist = map(lambda F: F * 12 , feetlist)
print(list(inchlist))

numlist = list(range(1,60,3))
evenlist = list(filter(lambda num: not(num % 2),numlist))
print(evenlist)

from functools import reduce

list1 = [1,2,3,4,5]
rsum = reduce(lambda x,y: x+y,  list1)
print(rsum)

list3 = [10,16,7,5,2,1,8,25]
grnum = reduce(lambda x,y: x if (x > y) else y, list3)
print(grnum)
```

Lambda

```
addsum = lambda x, y : x + y
ms = addsum(2,5)

inchlist = map(lambda F: F * 12 , feetlist)
absdiff = lambda x,y   :   x-y if x >= y else y-x
```

Handling Modules

```
import math
sqr = math.sqrt(36)

import time
tick = time.time()

import random  as rnd
rnum = rnd.randint(2,10)

from math import ceil,floor
print(ceil(2.5))
print(floor(2.5))
```

Dictionary Comprehension

```
newdict = {n:n**2 for n in range(10)}
newdict = {'a'+str(n):n**2 for n in range(10)}
newdict = {n:n**2 for n in range(3,19,3) if n%2 == 0}
```

Set Comprehension

```
newset = {n % 3 for n in range(30)}
```

List Comprehension

```
numlist = [ 2**x for x in range(10)]
evenlist = [ x for x in range(10) if x % 2 == 0]
primelist = [ i for i in range(2,1000) if isPrime(i) == True]
intlist = [ x if x >= 45 else -x for x in range(1,100)]
newlist = [val if val % 2 else -val  for val in range(2,20) if val % 3]
primelist = [x for x in range(2, n1) if x not in { j for i in range(2, int((n1**0.5)+1)) for j in range(i*2, n1, i)}]
doubleloop = [(i,j) for i in range(4) for j in range(i,4)]
```