

Control Statements :

- Whenever you want execute a statement or group of statement over and over [again and again], then we need to work the control statements
- Control statements also known as looping statement or iterations
- In C and Java [while | do...while | for ...]
- In Python [while and for ...]

While

Syn:

<exp1> ---> initialization
while <exp2>: --> condition
....stmt(s)
....exp3 ---> inc | Dec

hint	exp1	exp2	Exp3
1 to 10	1	< or <=	10
			increment +1
10 to 1	10	> or >=	1
			Decrement -1

In while loop it will check the condition first , if the condition is True then only it will enter into loop and execute looping statements till condition is False

i=1

while i<=10:

print(i)

i=i+1

Eg 2:

```
i=1
while i<=20:
    print("Welcome")
    i=i+1
```

Even Number:

```
i=2
while i<=10:
    print(i)
    i=i+2
```

Note : sleep() used to delay the execution of the programming statement(s) till the specified time in seconds ,It is from “time” module

```
import time
n=int(input("Enter a number "))
i=1
while i<=10:
    time.sleep(.5)
    print(n," X ",i,"=",n*i)
    i=i+1
```

Example for sum of natural No's:

```
import time
n=int(input("Enter a number "))
s=0
i=1
while i<=n:
    time.sleep(.5)
    s=s+i
    i=i+1
print("Sum is : ",s)
```

Example for Factorial :

```
import time
n=int(input("Enter a number ")) #n=3
f=1
i=1
while i<=n:
    time.sleep(.5)
    f=f*i
    i=i+1
print("Fact is : ",f)
```

Example :

```
import time
```

```
n=input("Enter u r passcode ")  
while n!="shashi":  
    time.sleep(.5)  
    print("Plz Try Again...!")  
    n=input()
```

For Loop:

Syn:
for <variable> in <iterables>:
....statement(s)

iterable objects nothing but
collections [str|list|tuple|set| frozenset | dict |
orderedDict | cursor | file | callable_iterator ...]
range

```
import time  
print("Reading the values From List collection")  
lst=[10,20,30]  
for i in lst:  
    time.sleep(1)  
    print(i) ""
```

```
print("Reading the values From tuple collection")
```

```
t=(10,"A","B",3.14,None)
```

```
print(t)
```

```
for i in t:
```

```
    time.sleep(.2)
```

```
    print(i)
```

```
print("Reading Data From str Object ")
```

```
s="Have a nice Day"
```

```
for i in s:
```

```
    time.sleep(.2)
```

```
    print(i)
```

```
print("Reading Data From Dict Collection ")
```

```
stu={"sno":101,"sname":"ramesh","scity":"Kadapa"}
```

```
for i in stu:
```

```
    time.sleep(.2)
```

```
    print(i)
```

```
Range():
```

```
'''
```

Syn: range(stop) -> range object

```
>>>range(10)
```

1.It will return a range object with range(0,10)

2.Predefined class type for range object is <class 'range'>

3.range object is also iterable object thus we can use it in for loop"

```
r=range(10)
```

```
print("Result is : ",r) # output: Result is : range(0,10)
```

```
print("Type is : ",type(r)) #Type is : <class 'range'>
```

```
import time
```

```
for i in r: # here r is range(0,10)
```

```
    time.sleep(.2)
```

```
    print(i)
```

```
#range(stop)
```

```
#Eg: range(10) -> range(0,10)
```

```
#range(start,end[,step] ) ->range object
```

```
#Eg: range(1,11,1)
```

```
r=range(1,11,1)
```

```
print("Type is : ",type(r)) #<class 'range'>
```

```
print("Result is : ",r) #range(1,11)
```

```
import time
```

```
for i in r:
```

```
    time.sleep(.2)
```

```
print(i)
```

Note : step value can be a +ve or -ve number but it should not be 0

```
import time
```

```
for i in range(10,101,10):
```

```
    time.sleep(.2)
```

```
    print(i)
```

```
print("In Reverse ")
```

```
for i in range(10,0,-1):
```

```
    time.sleep(.2)
```

```
    print(i)
```

```
print("Even No from 2 to 40")
```

```
for i in range(2,41,2):
```

```
    time.sleep(.1)
```

```
    print(i)
```