while loop

In [1]:
```
n=int(input('enter an input'))
i=0
while i<=n:
    print(i)
    i=i+1 #i +=1
```

```
enter an input10
0
1
2
3
4
5
6
7
8
9
10
```

In [1]:
```
# reverse of a number 123 - 321
123/10
```

Out[1]:
```
12.3
```

In [2]:
```
# reverse of a number 123 - 321
n=int(input('enter value :'))
rev=0
while(n>0):
    rem = n%10
    rev=rev*10+rem
    n=n//10
print(rev)
```

```
enter value :1223
3221
```

In [3]:
```
# Functions
```

- A function s a Group of Statement to do a specific task
- function breaks code into small modules to look more organised
- code re-useability
- Types of Functions -Built-in functions -user defined fuctions

In [ ]:
```
# user defined functions
### syntax in c
```
function fname(){
    cond/stmts to execute
}
syntax to python
def fname():
    cond/stmts
    return
fname()
```
- advantages
- making large code into small modules
- reuse of acode in a function by calling its frame ..
# types of argyments
-required arguments
-keyword arguments
-default arguments
-variable - length arguments
```

In [8]:
```
a=4
b=10
sum([a,b])
```

Out[8]:
```
14
```

In [10]:
```
a=[1,2,3,4,5]
max(a)
```

Out[10]:
```
5
```

In [11]:
```
len(a)
```

Out[11]:
```
5
```

In [12]:
```
min(a)
```

Out[12]:
```
1
```

In [15]:
```
a='ismart'
len(a)
```

Out[15]:
```
6
```

def add (a,b): c=a+b return c a=int(input('enter a value')) b=int(input('enter b value')) add(a,b)

In [ ]:
```
def add (a,b):
    c=a+b
    return c
a=int(input('enter a value'))
b=int(input('enter b value'))
a=5
b=1
add(b,a)
```

In [34]:
```
#keyword arguments
def key(str):
   print(str)
key(str=123)
```

```
123
```

In [4]:
```
def keyw(name,clz):
    print('name:',name)
    print('clz:',clz)
keyw(clz='aits',name='abc')
```

```
name: abc
clz: aits
```

In [3]:
```
# n odd numbers using functions
n=int(input('enter a value'))
def odd(n):
    for i in range(1,n+1):
        if i%2 !=0:
            print(i,end=' ')
        return
odd(n)
```

```
enter a value40
1
```

In [ ]:

In [ ]: