Python

Loops for

Loops

Execute statements multiple times under a controlled conditions

Basics of a loop

Head - loop control

Loop body – statements executed

Iteration - number of times the loop is executed

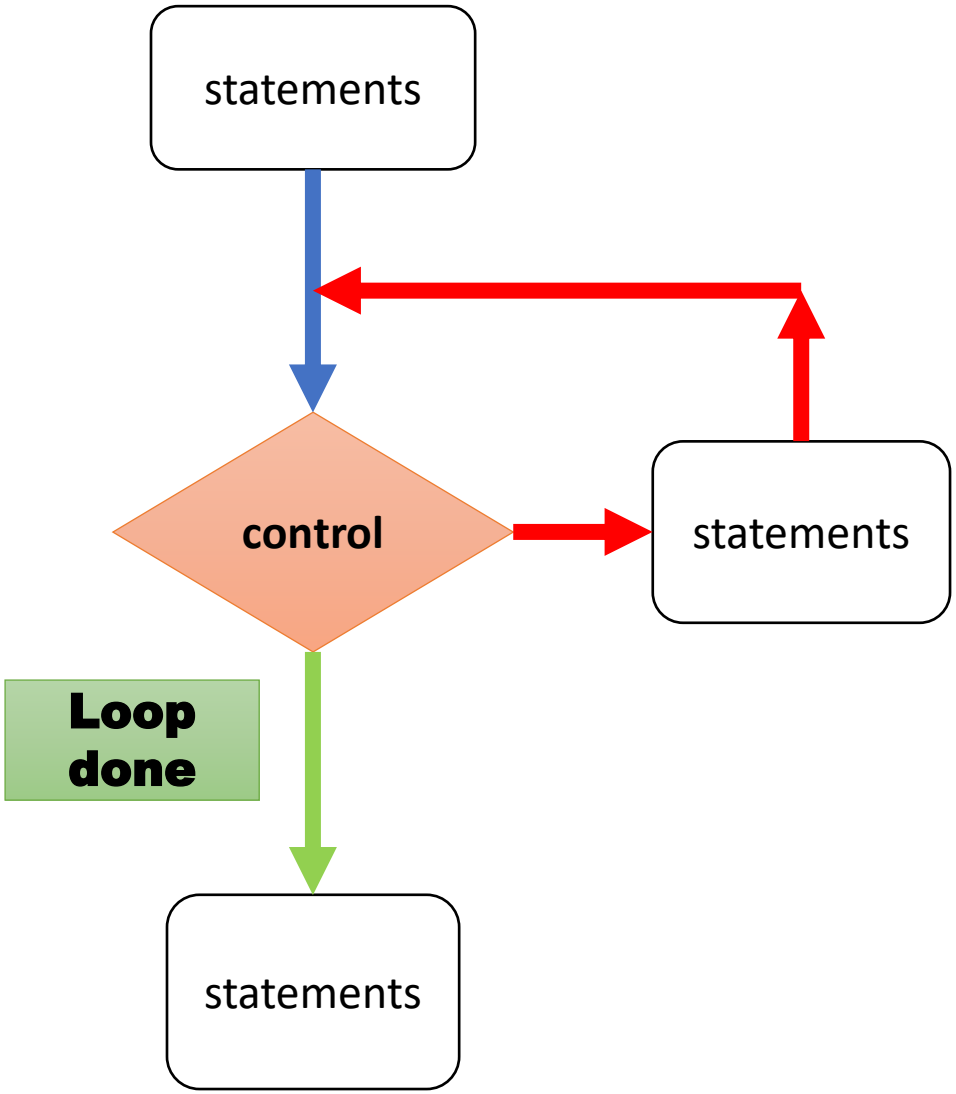
Two type of loops

Fixed (forced)

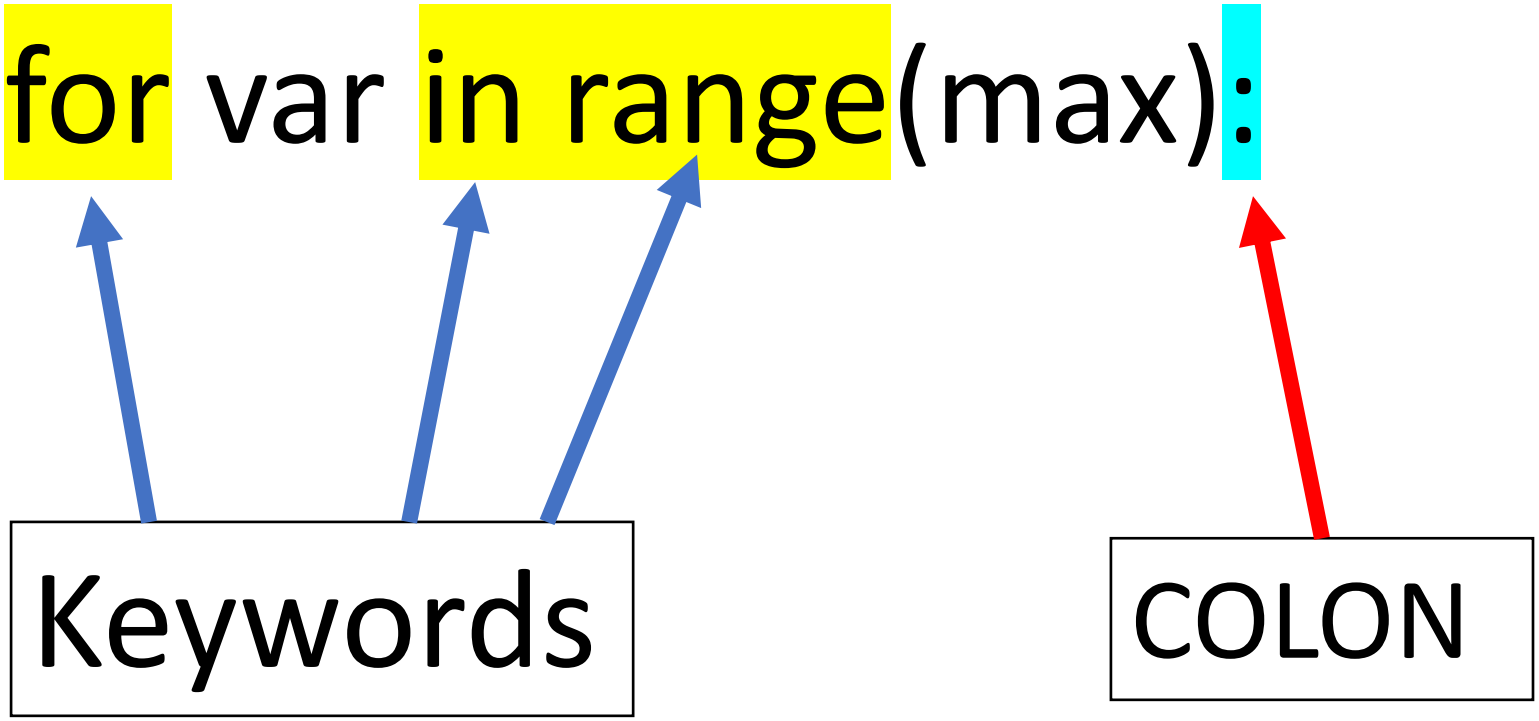
Controled

For loop

Fixed


statements
control
statements
Loop done
statements

For loop



For Loop

Forms:

CONTROL

ITERATION

```
for var in range(max):
    statements
```

HEAD

BODY

For Loop

Form:

Maximum number to stop (ending) looping when var is equal

```
for var in range(max):
    statements
```

Defaults:
Start = 0
Increment = 1

NOTE

```
print('----start—')
for x in range(5):
    print(' x -> ',x)
print('-----done----')
```

Example

```
for x in range(5):
    print(‘ x -> ‘,x)
```

For Loop more

Maximum number (ending) to stop looping when var is equal

Form:

```
for var in range(start,end):
    statements
```

Starting value Must be less than end value

Default: Increment = 1

Examples

```
for x in range(1,5):
    print (‘x_> ‘,x)
```

For Loop more

Form:

```
for var in range(start,end,inc/dec):
    statements
```

Increment (add to start value)
Decrement (subtract from start value)

Note:
Increment: start must be less than ending
Decrement: start must be greater than ending

Examples

```
for x in range(1,20,3):
    print(‘x-> ‘,x)

for x in range(25,0,-5):
     print(‘x-> ‘,x
```

NOTE

```
for x in range(1,20,3):
```

MUST BE INTEGERS

Example

```
def spacer():
    z=input('hit enter key to continue')
    print('--------------')
    return
def main():
    for a in range(5):
        print('a ->',a)
    spacer()
    for b in range(1,10):
        print('b ->',b)
    spacer()
    for c in range(0,20,4):
        print('c ->',c)
    spacer()
    for d in range(30,5,-5):
        print('d ->',d)
    spacer()
    m=0.0
    for m in range(1.0,5.0, .5):
        print('m ->',m)
main()
```

for01

NOTE

```
for x in range(1,20,3):
```

CAN BE VARIABLES

example

```
def spacer():
    z=input('hit enter key to continue')
    print('--------------')
    return
def main():
    spacer()
    srt = int(input('enter loop start number: '))
    end = int(input('enter loop ending: '))
    incdec = int(input('enter increment or decrement: '))
    spacer()
    for c in range(srt,end,incdec):
        print('c ->',c)
    spacer()
main()
```

for02

Nested loops

```
for x in range(1,10,1):
    statements
    for y in range (1,5,1):
        statements
```

note

```
def main():
    print('Program start')
    for aa in range(1,11,1):
        zz = input('hit enter key to continue')
        print(" aa -> ",aa)
        for bb in range(1,11,1):
            cc = aa * bb
            print(aa, ' X ', bb, ' = ',cc)
    print('Program done')
main()
```

Inside for loop

Outside for loop

Executed when outside loop is done

Try This

```
def main():
    print('Program start')
    for aa in range(1,11,1):
        zz = input('hit enter key to continue')
        print(" aa -> ",aa)
        for bb in range(1,11,1):
            cc = aa * bb
            print(aa, ' X ', bb, ' = ',cc)
  print('Program done')
main()
```

break

break

Break out of loop before the loop is done

Command: break

note

```
def main():
    for x in range(1,50,1):
        print('x -> ',x)
        if x == 10:
            print('time to break')
            break
    print(' -- done --')
main()
```

Stop looping, go to next statement

LOOP

Executed this statement when loop is done or when 'break' is executed in the loop

Try this

```
def main():
    for x in range(1,50,1):
        print('x -> ',x)
        if x == 10:
            print('time to break')
            break
    print(' -- done --')
main()
```

for05

Else

Where execution goes when loop terminated normally

For … else

```
for x in range(1,10):
    statements
else:
    statements
statements
```

For … else example

```
def main():
    print('\n'*2)
    for x in range(1,10):
        print(' x-> ',x)
    else:
        print(' end of loop')
    print('----done')
main()
```

for03

MORE

We will see other uses for the ‘for’ loop

And:

range is an object that can be used by itself

For Break and else

```
def main():
    limit = int(input('enter limit: '))
    for x in range(1,limit,1):
        print('x -> ',x)
        if x == 10:
            print('time to break')
            break
    else:
        print(' Else - loop done')
    print(' -- done --')
main()
```

for06

done