Python

Loops while

Loops

Execute code multiple time 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

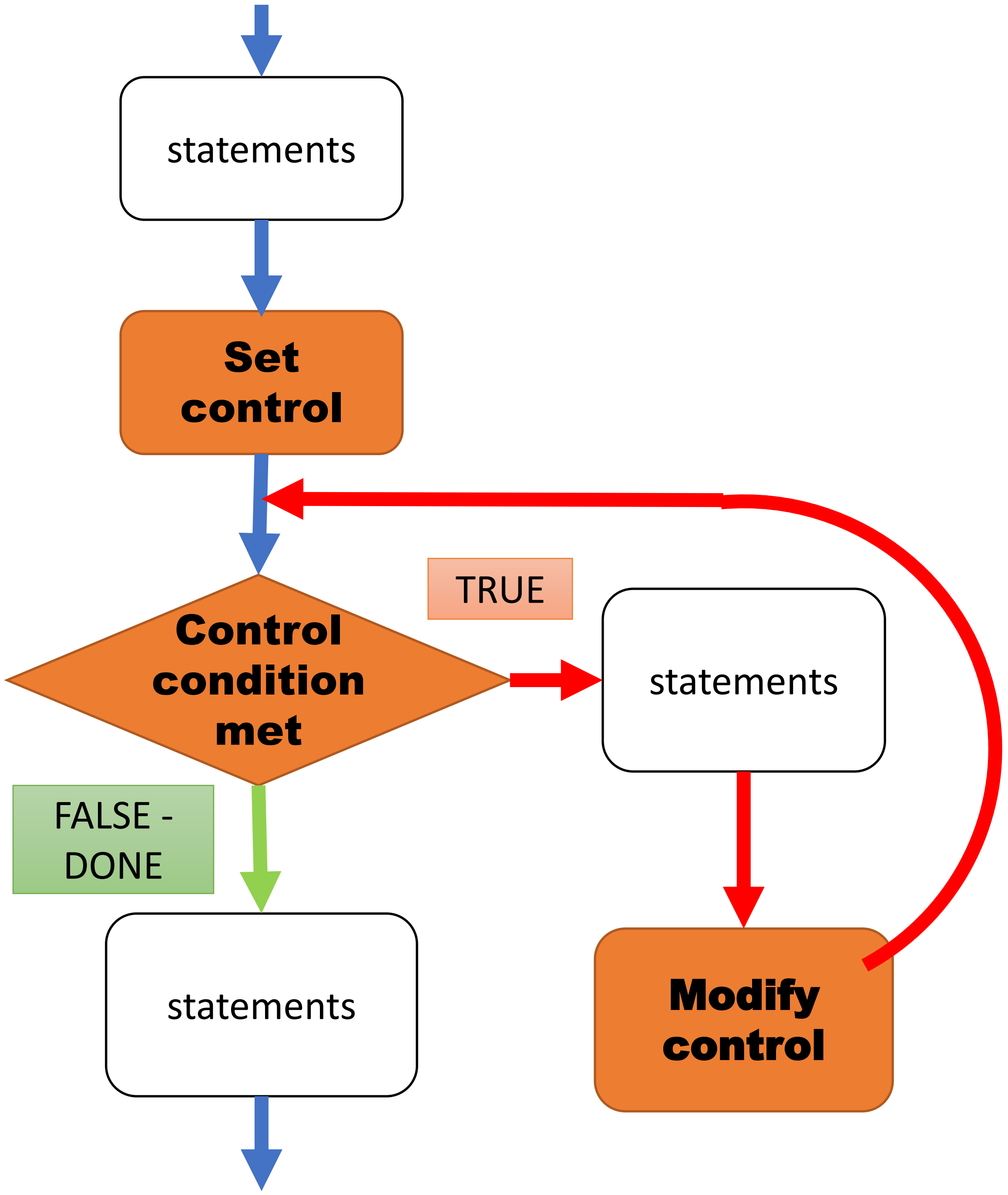
Fixed (forced)

Controled

While loop

conditions

While loops use conditions and Boolean logic to control looping


statements
Set control
Control condition met
TRUE
statements
Modify control
FALSE - DONE
statements

While FORM

```
While expression:
    statements
```

CONDITION
and or
BOOLEAN

While Expression (Condition)

The while executes the statement that indented underneath when the expression is TRUE.

The expression (aka condition) has the same rules as the ‘if’ statement

Example loop control

```
x = 1
While (x < 5):
     print(‘ line->’,x)
      x = x + 1
```

LOOP CONTROL

note

```
def main():
    x = 1
    y = 2
    z = 1
    while (z < 5):
        a = x * y
        print('a->',a)
        x = x + 1
        z = z + 1
    print('--done--')
main()
```

loop

Loop done

Try This

```
def main():
    x = 1
    y = 2
    z = 1
    while (z < 5):
        a = x * y
        print('a->',a)
        x = x + 1
        z = z + 1
    print('--done--')
main()
```

while01

Input control using while

Set up condition to test a response to continue or stop looping

Example

```
While (ans == ‘n’ or ans == ‘N’):
    statements
    ans = input(‘Continue y/n’)
```

note

```
def main():
    ans = 'y'
    gp = 0.0
    while ((ans == 'y') or (ans == 'Y')):
        print('------------')
        pay = float(input('enter pay rate: '))
        hrs = float(input('enter hours: '))
        gp = pay * hrs
        print('Gross pay: ',gp)
        ans = input('\nContinue y/n -> ')
    print(' –done—')
main()
```

Loop control

condition

statements

Loop control

while02

Try this

```
def main():
    ans = 'y'
    gp = 0.0
    while ((ans == 'y') or (ans == 'Y')):
        print('------------')
        pay = float(input('enter pay rate: '))
        hrs = float(input('enter hours: '))
        gp = pay * hrs
        print('Gross pay: ',gp)
        ans = input('\nContinue y/n -> ')
    print(' –done—')
main()
```

while02

Nested loops

```
while condition:
    statements
    while condition:
        statements
```

Nested while note

```
def main():
    x = 5
    m = 3
    out1 = 1
    while (out1 < x):
        in1 = 1
        print(' outside-> ',out1)
        while (in1 < m):
            print('       inside-> ',in1)
            in1 = in1 + 1
        out1 = out1 + 1
    print('program terminated')
    return
main()
```

OUTSIDE LOOP

INSIDE LOOP

Nested while example

```
def main():
    x = 5
    m = 3
    out1 = 1
    while (out1 < x):
        in1 = 1
        print(' outside-> ',out1)
        while (in1 < m):
            print('        inside-> ',in1)
            in1 = in1 + 1
        out1 = out1 + 1
    print('program terminated')
    return
main()
```

while03

break

break

Break out of loop before the loop is done

Command: break

note

```
def main():
    x =1
    while x <100:
        print('x -> ',x)
        if x == 25:
            print('time to break')
            break
        x = x + 1
    print(' -- done --')
main()
```

statements

loop02

Try this

```
def main():
    x =1
    while x <100:
        print('x -> ',x)
        if x == 25:
            print('time to break')
            break
         x = x + 1
    print(' -- done --')
main()
```

while04

Else

Where execution goes when loop terminates normally instead of next statement

While … else

```
statements
while    condition:
    statements
else:
    statements
statements
```

while with else note

```
def main():
    x = 5
    numb = 1
    print('\n***while start***\n\n')
    while (numb < x):
        print('--number-> ',numb)
        numb = numb + 1
    else:
        print('\nloop done')
    print('\nprogram terminated')
    return
main()
```

Example while with else

```
def main():
    x = 5
    numb = 1
    print('\n***while start***\n\n')
    while (numb < x):
        print('--number-> ',numb)
        numb = numb + 1
    else:
        print('\nloop done')
    print('\nprogram terminated')
    return
main()
```

while05

Using the while statement

While loop Menu 1 of 3

```
def sel001():
    print('you selected 1 \n')
    input('hit enter to continue\n')
    return
def sel002():
    print('you selected 2 \n')
    input('hit enter to continue\n')
    return
def displaymenu():
    print('\n'*20)
    print(' '*10,'1  select number one')
    print(' '*10,'2 select number two')
    print('\n')
    print(' '*10,'9 quit')
    print('\n'*5)
    return
```

while10

Menu 2 of 3

```
def menu():
    selection = 0
    while (selection != 9):
        displaymenu()
        selection = input(' Select 1,2 or 9 [to quit]: ')
        if (selection =='1' ):
            sel001()
        elif (selection == '2'):
            sel002()
        elif (selection == '9'):
            print('--- quit the menu')
            return
        else:
            print(' invalid selection, try again')
    return
```

Menu 3 of 3

```
def main():
    ans = 'y'
    while (ans == 'y'):
        menu()
        ans = input('again y/n ')
    print('program terminated')
    return
main()
```

While with menu

Program with a:

- while
- functions

while20M

caution

DO NOT DO THIS

```
While True:
      statements
```

DO NOT DO THIS

If you do this in your program:
Program will be rejected

This is an infinite loop
Why, the condition is always TRUE
And there is no way to change the condition

Infinite loop

An infinite loop is code (program) that will run forever.

The loop controls do not work to stop the looping.

```
def main():
    x = 1
    y = 2
    z = 1
    while (z < 5):
        a = x * y
        print('a->',a)
        x = x + 1
        z = z - 1
    print('--done--')
main()
```

Infinite loop
try this

Crtl-c
To stop

infinite1

done