PYTHON

TAJENDAR ARORA

Operators in Python

- Arithmetic Operators

(+,-,*,/,%,**(exponent),//(9//2 = 4 and 9.0//2.0 = 4.0)

- Comparison (Relational) Operators

(>,<,>=,<=,==,!=)

- Assignment Operators

(=,+=,-=,/=,%=)

- Logical Operators

(and ,or ,not)

- Membership Operators

(in , not in use in list)

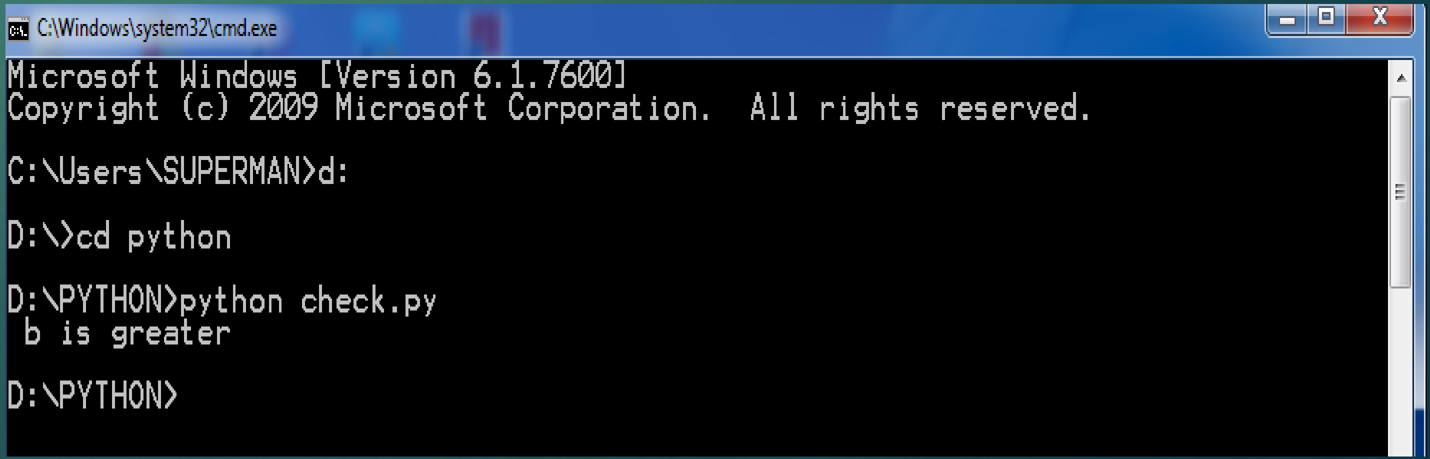
CONTROL FLOW STATEMENTS IN PYTHON

If-else : if - else is a control flow statement in python. if else is used when we have two condition first is in if block and second is in else block if first condition is true then if block executed if second is true then else block is executed.

```
a=10

b=20


if a>b:

	print ('a is highest‘)

else :

	print (' b is greater‘ )
```

#Before the print command we use tab key.

OUTPUT



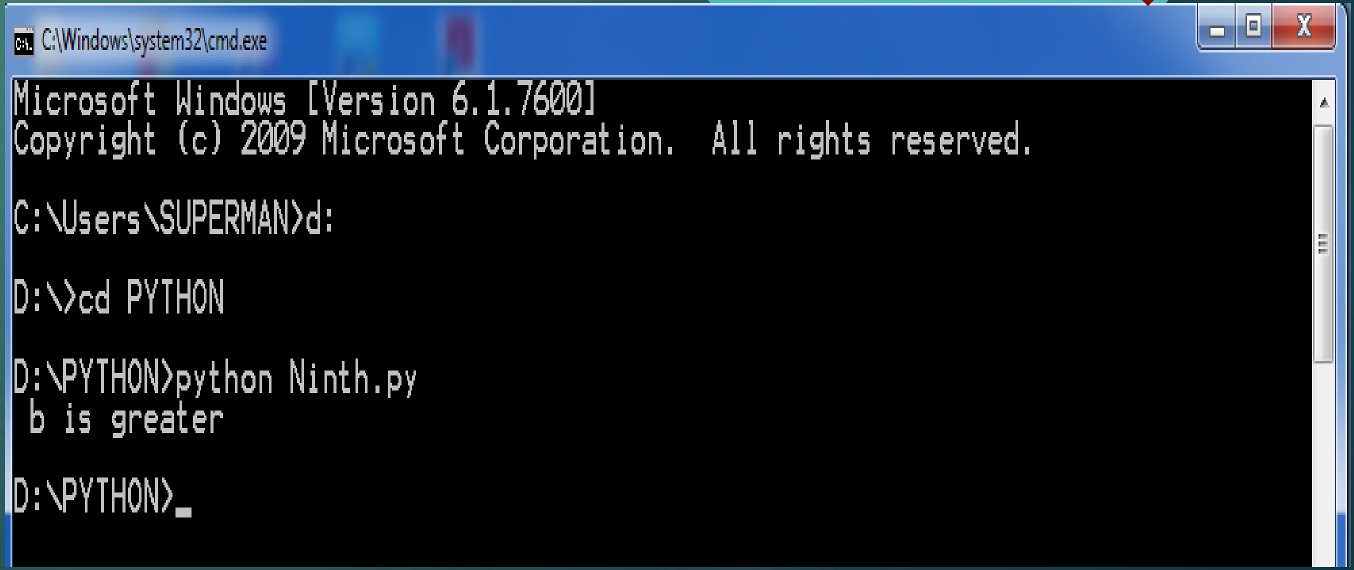
```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\SUPERMAN>d:

D:\>cd python

D:\PYTHON>python check.py
 b is greater

D:\PYTHON>
```

If elif

elif : As we use if –else statement same as we used elif control flow statement when we have more than two condition. It is like nested if –else in python.

```
a=10
b=20
c=15
if a>b and a>c:
print ('a is highest')
elif b>a and b>c:
print ('b is highest')
else:
print ('c is highest')
```

OUTPUT



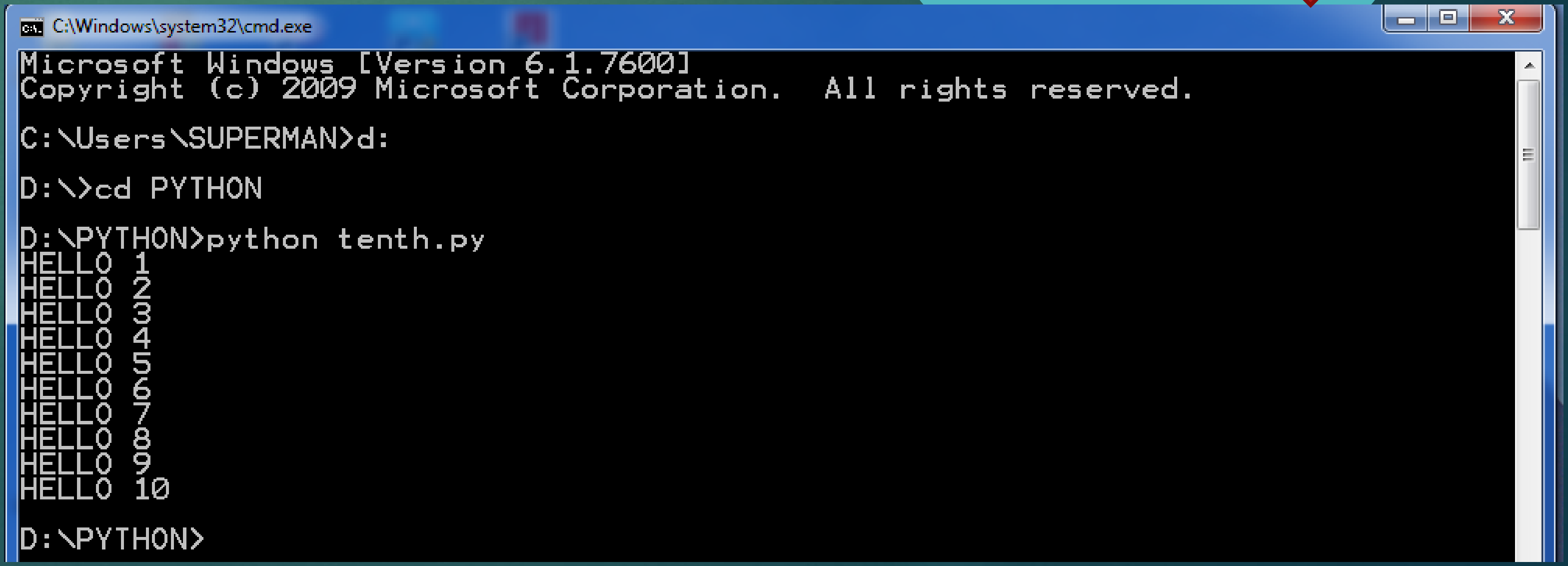
```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\SUPERMAN>d:

D:\>cd PYTHON

D:\PYTHON>python Ninth.py
 b is greater

D:\PYTHON>_
```

While Loop

while Loop: It is another loops to check condition is true or false. when condition is true loop is executed as the condition becomes false loop execution ends. Example:

```
a=1
while a<=10:
    print ('HELLO',a)
    a=a+1
```

OUTPUT



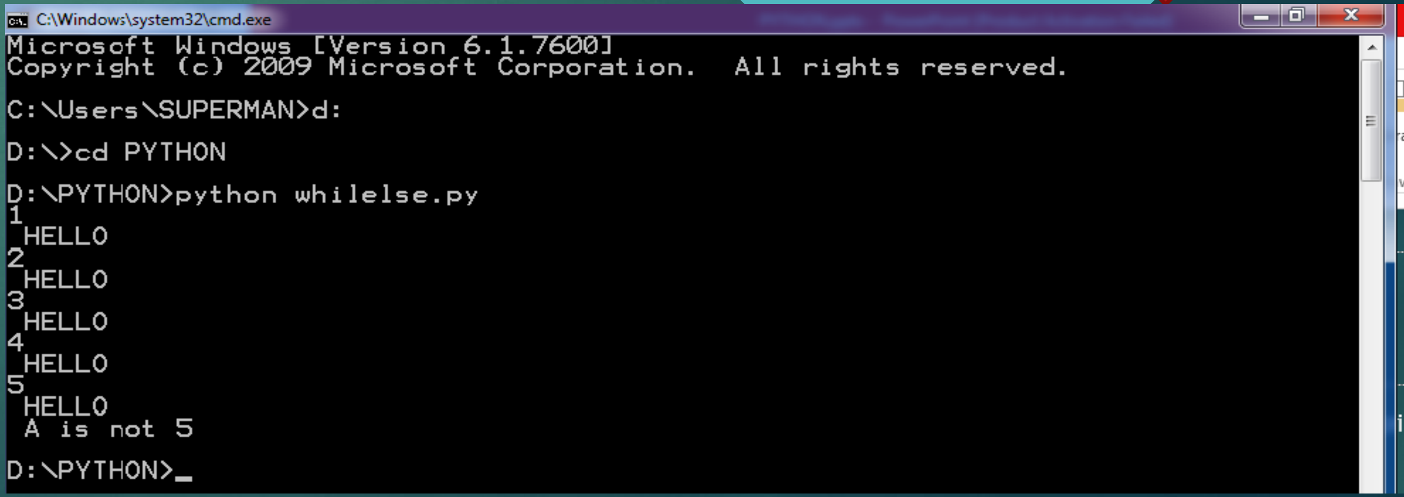
```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\SUPERMAN>d:

D:\>cd PYTHON

D:\PYTHON>python tenth.py
HELLO 1
HELLO 2
HELLO 3
HELLO 4
HELLO 5
HELLO 6
HELLO 7
HELLO 8
HELLO 9
HELLO 10

D:\PYTHON>
```

else with loop

<u>while-else:</u> We can use the else with while same as if-else. while is executed when while condition is true .If while condition is false then else part executed.

```
a=1
while a<=5:
    print (a)
    print (' HELLO ‘)
    a=a+1
else:
    print (' A is not 5 ‘)
```

OUTPUT



```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\SUPERMAN>d:

D:\>cd PYTHON

D:\PYTHON>python whilelse.py
1
 HELLO
2
 HELLO
3
 HELLO
4
 HELLO
5
 HELLO
 A is not 5

D:\PYTHON>_
```

For loop

It has the ability to iterate over the items of any sequence, such as a list or a string.

```
for iterating_var in sequence:
statements(s)
```

Simple loop

```
a=1
for a in range(1,10):
    print (a)
```

OUTPUT

For loop break

- **It terminates the current loop and resumes execution at the next statement, just like the traditional break statement in C.**
- **The most common use for break is when some external condition is triggered requiring a hasty exit from a loop. The break statement can be used in both *while* and *for* loops.**

```
var = 10
while var > 0:
        print ('Current variable value :', var )
        var = var -1
        if var == 5:
                break
print ("Good bye!" )
```

OUTPUT

For loop break

```
for letter in 'Python':
    if letter == 'h':
    print ('Current Letter :', letter )
    break
```

OUTPUT

For loop pass

- It is used when a statement is required syntactically but you do not want any command or code to execute. The **pass** statement is a *null* operation; nothing happens when it executes.

OUTPUT

```
for letter in 'Python':
    if letter == 'h':
        pass
        print ('This is pass block' )
    print ('Current Letter :', letter )
print ("Good bye!" )
```

For loop

<u>for loop used with the List</u>

We can use the for loop with list also.

```
city=['AGRA','DELHI','MUMBAI']
for a in city:
      print (a)
```