# Assignment - II

1) Two Boolean datatypes : True and False

For example: if
```
a = 'Welcome'
if a == string:
```
then it prints True

```
if a == int:
```
then it prints False.

2) Three types of Boolean operators:

* AND * OR * NOT

3) Truth tables:-

AND

| x | y | x and y |
|---|---|---|
| F | F | F |
| T | F | F |
| F | T | F |
| T | T | T |

OR

| x | y | x or y |
|---|---|---|
| F | F | F |
| T | F | T |
| F | T | T |
| T | T | T |

NOT

| x | not x |
|---|---|
| T | F |
| F | T |

4) * (5>4) and (3==5) * not(5>4) * not((5>4) or (3==5))

False False False

* (True and True) and (True==False) * (not False) or (not True)

False True

5) Six comparison operators:-

* less than (<)
* greater than or equal to (>=)
* less than or equal to (<=)
* equal to (==)
* greater than (>)
* not equal to (!=)

6) Difference between equal to and assignment operator

An assignment operator (=) assigns values to a variable whereas aequal to operator (==) checks if the operands on both sides are equal or not

```
a = "ineuron"
assignment operator
```

```
a == string:
  'True'
Equal to operator
```

7) spam=0

```
if spam==0:          — block
    print('eggs')
if spam>5:           — block
    print('bacon')
else:                — block
    print('ham')
    print('spam')
```

8) spam = 'Hi'

```
if spam==1:
    print('Hello')
elif spam==2:
    print('Howdy')
else:
    print('Greetings!')
```

9) If a program is stuck in an endless loop press CTRL + C key to end the loop.

10. Difference between break and continue

Break statement terminates the execution of remaining iteration of the loop whereas a continue statement only terminates the current iteration of loop and resumes the control of program on next iteration.

11. Difference between range(10), range(0,10), range(0,10,1)

All three statement prints the same values.

o/p →
0
1
2
3
4
5
6
7
8
9

12.)
```
l = []
for i in range(1,10):
    l.append(i)
    print(i)
```

Using while loop

```
a = 1
while (a <= 9):
    print(a)
    a += 1
```

13.) If you had a function named bacon() inside a module named spam, how would you call it after importing spam?

importing → from spam import *

bacon()