Assignemnt 2

1.What are the two values of the Boolean data type? How do you write them?

True (T) and False (F) are two different value of boolean data type

In [9]:

```
a = True
b = False

print(type(a))
print(type(b))
```

```
<class 'bool'>
<class 'bool'>
```

2.What are the three different types of Boolean operators?

there are three different types of boolean operators `and` , `or` and `not` . `and` in and when both condition is true then it true otherwise false , in `or` condtion is true when atleast one condition true otherwise false and `not` it convert false to true or vise varsa.

3 .Make a list of each Boolean operator's truth tables (i.e. every possible combination of Boolean values for the operator and what it evaluate).

AND

`False and False - False`

`False and True - False`

`True and False - False`

`True and True - True`

OR

`False and False - False`

`False and True - True`

`True and False - True`

`True and True - True`

Not

`False - True`

`True - False`

4. What are the values of the following expressions?

(5 > 4) and (3 == 5) --> False

not (5 > 4) --> False

(5 > 4) or (3 == 5) --> True

not ((5 > 4) or (3 == 5)) --> False

(True and True) and (True == False) --> False

(not False) or (not True) --> True

5. What are the six comparison operators?

- == equal operator
- != not equal to operator
- < less than operator
- > greater than operator
- <= less than or equal to operator
- >= greater than or equal to operator

6. How do you tell the difference between the equal to and assignment operators?Describe a condition and when you would use one.

The assignment operators are used to assinging value to varaiable and the equal operator are comparison operator that helps to comparing two values.

Ex

In [3]:

```python
a = 10 # this is assignment operator
b = 20 # this is assignment operator

if a == b: # this is comparison operator
    print('a is equal to b')
else:
    print('a is not equal to b')
```

```
a is not equal to b
```

7. Identify the three blocks in this code:

spam = 0

if spam == 10:

print('eggs')

if spam > 5:

print('bacon')

else:

print('ham')

print('spam')

print('spam')

In [4]:

```
spam = 0

if spam == 10:
    print('eggs')

if spam > 5:
    print('bacon')

else:
    print('ham')
    print('spam')
    print('spam')
```

```
ham
spam
spam
```

8. Write code that prints Hello if 1 is stored in spam, prints Howdy if 2 is stored in spam, and prints Greetings! if anything else is stored in spam.

In [5]:

```
spam = int(input("please enter the value : "))

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

```
please enter the value : 2
Howdy
```

In [6]:

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

In [7]:

```
x(1)
```

```
Hello
```

In [8]:

```
x(2)
```

```
Howdy
```

In [9]:

```
x(3)
```

```
Greetings!
```

9.If your programme is stuck in an endless loop, what keys you'll press?

if programme is stuck in an endless loop then we use `CTRL + C`

10.How can you tell the difference between break and continue

the break statement is used to terminate the loop and the continue statment is used to skip code withing the loop and after skiping the loop continues where it left off.

11. In a for loop, what is the difference between range(10), range(0, 10), and range(0, 10, 1)?

In [13]:

```
# The range(10) print values form 0 to 9
for i in range(10):
    print(i)
print()
```

```
0
1
2
3
4
5
6
7
8
9
```

In [15]:

```
# it print number upto 10
for i in range(0,10):
    print(i)
print()
```

```
0
1
2
3
4
5
6
7
8
9
```

In [16]:

```
# it use range to print divisible by 1
for i in range(0,10,1):
    print(i)
print()
```

```
0
1
2
3
4
5
```

```
6
7
8
9
```

12. Write a short program that prints the numbers 1 to 10 using a for loop. Then write an equivalent program that prints the numbers 1 to 10 using a while loop.

In [19]:

```
for i in range(1,10):
    print(i)
```

```
1
2
3
4
5
6
7
8
9
```

In [8]:

```
n = 1
while n <= 10:
    print(n)
    n += 1
```

```
1
2
3
4
5
6
7
8
9
10
```

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

after importing spam module the function can be called by name bacon()