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

In [ ]:
```
True and False
```

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

In [ ]:
```
Ans: AND , OR , NOT
```

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).

In [6]:
```
# First we make the AND Truth Table
print(" AND Truth Table ")
print(" A= False , B= False \t" ,0 and 0)
print(" A= False , B= True \t", 0 and 1 )
print(" A= True , B= False \t", 1 and 0)
print ( " A= True, B= True \t", 1 and 1)
```

```
AND Truth Table
A= False , B= False     0
A= False , B= True      0
A= True , B= False      0
A= True, B= True        1
```

In [15]:
```
print(" OR Truth Table ")
print(" A= False, B= False\t", 0 or 0)
print(" A= False, B= True\t",  0 or 1 )
print(" A = True, B= False\t", 1 or 0)
print(" A = True, B= True\t", 1 or 1)
```

```
OR Truth Table
A= False, B= False      0
A= False, B= True       1
A = True, B= False      1
A = True, B= True       1
```

In [27]:
```
# Not gate not able to done
```

4. What are the values of the following expressions?

In [31]:
```
(5>4) and (3==5)
```

Out[31]:
```
False
```

In [32]:
```
not (5>4)
```

Out[32]:
```
False
```

In [33]:
```
(5>4) or (3==5)
```

Out[33]:
```
True
```

In [34]:
```
not((5>4) or (3==5))
```

Out[34]:
```
False
```

5. What are the six comparison operators?

In [ ]:
```
== Equal
```

In [ ]:
```
!= Not Equal
```

In [ ]:
```
> Greater than
```

In [ ]:
```
< Less than
```

In [ ]:
```
>= Greater than or Equal to
```

In [ ]:
```
<= Less than or Equal to
```

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

In [ ]:
```
"=" is a assignment operator,
values are assign to variable in right and variable are left
```

In [ ]:
```
"==" is a equal to operator which check the value is equal or not, Logical condition
```

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 [41]:
```
spam=0
if spam ==10:   # Block first
    print('eggs')
    if spam> 5:    # Block second
        print('bacon')
else:
    print('ham')
    print('spam')
    print('bacon')
```

```
ham
spam
bacon
```

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 [62]:
```
spam=int(input())

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

```
2
Howdy
```

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

In [ ]:
```
Ctrl+c
```

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

In [ ]:
```
The Break statement is break the loop statement
```

In [ ]:
```
The continue statement is not break the loop statement
```

11. In a for loop, what is the difference

between range(10), range(0, 10), and range(0, 10, 1)?

In [67]:
```
for i in range(10):
    print(i)
```

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

In [68]:
```
for i in range(0,10):
    print(i)
```

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

In [70]:
```
for i in range(0,10,2):
    print(i)
```

```
0
2
4
6
8
```

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 [5]:
```
# While Loop
input=1
while input<=10:
    print(input)
    input +=1
```

```
1
2
3
4
```

```
5
6
7
8
9
10
```

In [8]:

```
# For loop
for i in range(1,11):
    print(i)
```

```
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?

In [ ]:

```
# module. function
spam.bacon()
```