

In [1]: *## 1. Two values of boolean data type ? How do you write them ?*  
 The two values allowed in boolean data type are true and false.  
 Internally python represents True as 1 and False as 0

File "<ipython-input-1-0895addc5dc5>", line 2  
 The two values allowed in boolean data type are true and false.  
 ^

**IndentationError:** unexpected indent

In [2]: *## 2. what are the three different types of Boolean operators?*  
 The three basic boolean operators are: AND, OR, and NOT.

File "<ipython-input-2-167f1466022a>", line 2  
 The three basic boolean operators are: AND, OR, and NOT.  
 ^

**SyntaxError:** invalid syntax

In [3]: *## 3. make a List of each boolean operators truth tables(i.e every possible combin  
 and what it evaluate)*

condition 1 (x)	condition 2 (y)	not x	x and y	x or y
false	false	true	false	false
false	true	true	false	true
true	false	false	false	true
true	true	false	true	true

File "<ipython-input-3-7671b4d4e973>", line 2  
 and what it evaluate)  
 ^

**SyntaxError:** invalid syntax

In [4]: *## 4. what are the values of the following expressions ?*

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

In [6]: print(a)

False

In [7]: a = not(5 > 4)  
 print(a)

False

In [11]: a = (5 > 4) or (3 == 5)  
 print(a)

True

In [12]: a = not((5 > 4) or (3 == 5))  
 print(a)

False

```
In [13]: a=(True and True) and (True == False)
         print(a)
```

False

```
In [14]: a=(not False)or(not True)
         print(a)
```

True

```
In [15]: ## 5. what are six comparison operators?
         The six Python comparison operators are Less than(<), Greater than(>), Less than or
         Equal to(==) and Not equal to (!=).
```

File "<ipython-input-15-717ef8e34844>", line 2  
 The six Python comparison operators are Less than(<), Greater than(>), Less than  
 or equal to(<=), Greater than or equal to(>=),  
 ^  
**SyntaxError:** invalid syntax

```
In [16]: ## 6. How do you tell the difference between the eual to and assignment operators ?
         ## would use one?

         The "=" is an assignment operator is used to assign the value on the right to the va
         operands are equal or not.
```

File "<ipython-input-16-bae8c893aaf2>", line 4  
 The "=" is an assignment operator is used to assign the value on the right to th  
 e variable on the left. The '==' operator checks whether the two given  
 ^  
**SyntaxError:** invalid character in identifier

```
In [17]: ## 7. identify the three blocks in this code:
```

```
In [18]: spam = 0
         if spam == 10:
             print('eggs')
             if spam > 5:
                 print('bacon')
             else:
                 print('ham')
             print('spam')
         print('spam')
```

spam

```
In [19]: The three blocks are everything inside the if statement and the lines print('bacon')

         print('eggs')
         if spam > 5:
             print('bacon')
         else:
             print('ham')
         print('spam')
```

File "<ipython-input-19-50e37902bfda>", line 1  
 The three blocks are everything inside the if statement and the lines print('bac  
 on') and print('ham').  
 ^  
**SyntaxError:** invalid syntax

In [20]: `## 8. write a code that prints Hello if 1 is stored in spam , prints howdy if 2 is s  
## if anything else is stored in spam`

In [3]: `spam = input()  
  
if spam == 1:  
 print('Hello')  
elif spam == 2:  
 print('Howdy')  
else:  
 print('Greetings!')`

1  
Greetings!

In [ ]: `## 9 if your is stuck in an endless loop , what keys you will press?`

In [ ]: `Ctrl + C`

In [4]: `## 10. how can you tell the difference between break and continue?`

The main difference between **break** and **continue** statement **is** that when **break** keyword it will exit the loop. In case of **continue** keyword, the current iteration that **is** ru **with** the next iteration

File "<ipython-input-4-7c3fcef7ae>", line 3  
The main difference between break and continue statement is that when break keyw  
ord is encountered,  
^  
**SyntaxError:** invalid syntax

In [5]: `## 11. in a for loop what is the difference between range(10),range(0,10),range(0,10`

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

0  
1  
2  
3  
4  
5  
6  
7  
8  
9

In [11]: `for i in range(0,10):  
 print(i)`

0  
1  
2  
3  
4  
5  
6

7  
8  
9

```
In [17]: for i in range(0,10,1):  
         print(i)
```

0  
1  
2  
3  
4  
5  
6  
7  
8  
9

```
In [18]: ## 12 write a short program that prints the numbers 1 to 10 using a for loop . then  
         ## that prints the number 1 to 10 using a while loop
```

```
In [20]: for i in range(1, 11):  
         print(i)
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

```
In [19]: i = 1  
         while(i<=10):  
             print(i)  
             i += 1
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

```
In [ ]: ## 13. if you had a function named bacon() inside a module named spam, how would you  
         This function can be called with spam.bacon().
```