

```
In [1]: ##1. What are the two values of the Boolean data type? How do you write them?  
This can take only the value true or false.  
1 = true  
0 = false
```

```
In [ ]: ##2. What are the three different types of Boolean operators?  
and, or & not are the boolean operators.
```

```
In [3]: ##3. Make a list of each Boolean operator's truth tables (i.e. every possible combin  
and operator:  
  A      B      Q  
  0      0      0  
  0      1      0  
  1      0      0  
  1      1      1  
  
or operator:  
  A      B      Q  
  0      0      0  
  0      1      1  
  1      0      1  
  1      1      1  
  
not operator:  
  A      Q  
  1      0  
  0      1
```

```
File "<ipython-input-3-b802b3da301b>", line 2  
  and operator:  
    ^  
SyntaxError: invalid syntax
```

```
In [4]: 4. What are the values of the following expressions?  
(5 > 4) and (3 == 5) =  
not (5 > 4)  
(5 > 4) or (3 == 5)  
not ((5 > 4) or (3 == 5))  
(True and True) and (True == False)  
(not False) or (not True)
```

```
File "<ipython-input-4-dbdaf705ad54>", line 2  
  (5 > 4) and (3 == 5) =  
                                ^  
SyntaxError: invalid syntax
```

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

```
Out[5]: False
```

```
In [6]: not (5 > 4)
```

```
Out[6]: False
```

```
In [7]: (5 > 4) or (3 == 5)
```

Out[7]: True

```
In [8]: not ((5 > 4) or (3 == 5))
```

Out[8]: False

```
In [9]: (True and True) and (True == False)
```

Out[9]: False

```
In [10]: (not False) or (not True)
```

Out[10]: True

```
In [11]: ##5. What are the six comparison operators?  
Less than(<)  
Greater than(>)  
Less than or equal to(<=)  
Greater than or equal to(>=)  
Equal to(==)  
Not equal to (!=).
```

File "<ipython-input-11-4b79773327bf>", line 2

Less than(<)

^

SyntaxError: invalid syntax

```
In [12]: ##6. How do you tell the difference between the equal to and assignment operators?De  
The "=" is an assignment operator is used to assign the value on the right to the va  
EX:a = 10;  
The '==' operator checks whether the two given operands are equal or not. If so, it  
EX:a == a, a == b;
```

File "<ipython-input-12-3a15b8f15e50>", line 2

The "=" is an assignment operator is used to assign the value on the right to th  
e variable on the left.

^

SyntaxError: invalid character in identifier

```
In [13]: ##7. Identify the three blocks in this code:
```

```
In [14]: spam = 0  
if spam == 10:  
    print('eggs')  
if spam > 5:  
    print('bacon')  
else:  
    print('ham')  
    print('spam')  
    print('spam')
```

File "<ipython-input-14-3f64fe7ce118>", line 3

print('eggs')

^

IndentationError: expected an indented block

```
In [15]: the three blocks are everything
print('eggs')
if spam > 5:
    print('bacon')
else:
    print('ham')
print('spam')
```

```
File "<ipython-input-15-71f66270df5c>", line 1
    the three blocks are everything
    ^
SyntaxError: invalid syntax
```

```
In [16]: ##8. Write code that prints Hello if 1 is stored in spam, prints Howdy if 2 is store
spam = input()

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

```
1
Greetings!
```

```
In [18]: ##9.If your programme is stuck in an endless loop, what keys you'll press?
```

```
In [19]: Ctrl + c
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-19-0eddc49d0a19> in <module>
----> 1 Ctrl + c

NameError: name 'Ctrl' is not defined
```

```
In [20]: ##10. How can you tell the difference between break and continue?
Break statement: when break keyword is encountered,it will exit the loop.
Continue statement: The continue statement skips the code that comes after it, and t
```

```
File "<ipython-input-20-da1262383aba>", line 2
    Break statement: when break keyword is encountered,it will exit the loop.
    ^
SyntaxError: invalid syntax
```

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

```
In [21]: for i in range(10):
        print(i)
```

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

8  
9

In [25]:

```
for i in range(0,10):  
    print(i)
```

0  
1  
2  
3  
4  
5  
6  
7  
8  
9

In [26]:

```
for i in range(0, 10, 1):  
    print(i)
```

0  
1  
2  
3  
4  
5  
6  
7  
8  
9

In [ ]:

*##12. Write a short program that prints the numbers 1 to 10 using a for loop. Then w*

In [32]:

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

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

In [33]:

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

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

