

In []: 1)What are the two values of the Boolean data type?
How do you write them?

True and False, using capital T and F, with the rest of the word in lowercase

In []: 2)What are the three Boolean operators?

and, or, and not

In []: 3)Write out the truth tables of each Boolean operator
(that is, every possible combination of Boolean values for
the operator and what they evaluate to).?

True and True is True.

True and False is False.

False and True is False.

False and False is False.

True or True is True.

True or False is True.

False or True is True.

False or False is False.

not True is False.

not False is True.

In []: 4)What do the following expressions evaluate to?

```
(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) ?
```

```
False
False
True
False
False
True
```

In []: 5)What are the six comparison operators?

```
==, !=, <, >, <=, and >=.
```

In []: 6A)What **is** the difference between the equal to operator **and** the assignment operator?

== **is** the equal to operator that compares two values **and** evaluates to a Boolean, **while =** **is** the assignment operator that stores a value **in** a variable.

6B)Explain what a condition **is and** where you would use one?

A condition **is** an expression used **in** a flow control statement that evaluates to a Boolean value.

In []: 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')
    ?
```

The three blocks are everything inside the **if** statement
and the lines `print('bacon')` **and** `print('ham')`.

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

In []: 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?

The code:

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

In []: 9)What can you press **if** your program **is** stuck **in** an infinite loop?

Press CTRL-C to stop a program stuck **in** an infinite loop.

In []: 10)What **is** the difference between **break** and **continue**?

The **break** statement will move the execution outside **and** just after a loop.
The **continue** statement will move the execution to the start of the loop.

In []: 11)What **is** the difference between **range(10)**, **range(0, 10)**, **and range(0, 10, 1)** in a **for** loop?

They **all** do the same thing. The **range(10)** call ranges **from 0** up to (but **not** including) **10**, **range(0, 10)** explicitly tells the loop to start at **0**, **and range(0, 10, 1)** explicitly tells the loop to increase the variable by **1** on each iteration.

In []: 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?

The code:

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

and:

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

In []: 13) If you had a function named `bacon()` inside a module named `spam`, how would you call it after importing `spam`?

This function can be called **with** `spam.bacon()`.