**Boolean Logic**

* Boolean logic stems from a type of mathematics that deals with only True/False statements (Boolean Algebra)
  + This comes from a man named George Boole who was a self-taught mathematician from the 1800s
  + Boolean algebra consists of equations that use True and False rather than numbers and variables
* Boolean Logic Consists of 3 main statements AND NOT OR
  + The values for these statements are True / False

**AND –** when comparing two statements, both must be true or both must be false for the entire statement to be True.

Ex: let x = 20

If x <31 and x%2==0:

This statement says: if x is less than 31 and x is divisible by 2

This is TRUE because 20<31 and 20 is divisible by 2

**OR –** When comparing two statements if one and not the other is True, then the whole equation is true. If both are false, then the entire statement is false.

Ex: let x = 20

If x <19 or x%2==0:

This statement says: if x is less than 19 or x is divisible by 2

When x = 20 this statement becomes TRUE since one of the two conditions is true.

When x = 21 this statement becomes false because neither conditions are true.

**NOT –** when comparing two statements, if neither statements are true then the statement is true.

* Generally in computer science we use the ! symbol to denote not.

Ex: let x = 20

If x != 19: True

If x%2 != 0: False

If x != 30 and x %3 !=0: True for when x = 20

**XOR –** (aka exclusive or) this statement is used for when only one of two statements is true – this is similar to or –but often times this is used for binary conversions in encryption and cryptography so we will not worry about that for now.

Exercises:

1. Let x = 36 , will the following statement be True or False?

If len(“Katie”) > 6 or x %6 ==0:

1. Let x = -8, will the following statement be True or False?

If abs(x) < 10 and x%2==0:

1. Let x = 24, will the following statement be True or False?

If (x/3)%2==0 or x!=5

1. Write a function that mimics how the abs() built in function works (absolute value operator).

* Write conditions that turn a negative number into a positive