**Truth Tables**

3 min

Logic gates have certain rules that determine what the outputs are with respect to the inputs a and b. When we are analyzing a logic gate, we can visualize all of the possible outputs by making a *truth table*. A truth table shows the output for all possible inputs.

For example, if we have a gate that returns 1 if and only if both the inputs a and b are 1, we can create the truth table:

| ***a*** | ***b*** | **output** |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Each row represents an observation of output depending on the values of a and b. So the first row shows that when a is 0 and b is 0, the output will be 0.

When we make a truth table, we want to represent the entire universe of possibilities. This means that we want every combination of inputs to be represented. For 2 variables (a and b), we will need 4 rows to represent all of these combinations.

In Python, and many other programming languages, 1 evaluates to be True, and 0 evaluates to be false.

**Instructions**

1. Checkpoint 1 Passed

**1.**

Let's suppose we have a gate that returns 1 if either a is 1 or b is 1, but not if they are both 1.

Here is the truth table that corresponds to the gate, with the third row missing:

| ***a*** | ***b*** | **output** |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| val1 | val2 | val3 |
| 1 | 1 | 0 |

Create the variables val1, val2, and val3 and assign them to the values that will complete the truth table for this gate.

Hint

We are missing the situation where a is 1 and b is 0. What would the output be in this case?

**script.py**

```
# Truth table:
# _____________________
# a     b     output
# 0     0     0
# 0     1     1
# val1  val2  val3
# 1     1     0
#


val1 = 1 #add 0 or 1 here
val2 = 0 #add 0 or 1 here
val3 = 1 #add 0 or 1 here
```