OR Gate

3 min

The next gate we are going to build is the *OR gate*. This gate receives two inputs and returns 1 if either one of the inputs is 1.

To build your OR_gate(), you should use any combination of the gates you've already made: NAND_gate(), NOT_gate(), and AND_gate().

Here's the truth table:

a	b	output
0	0	0
0	1	1
1	0	1
1	1	1

Instructions

1. Checkpoint 1 Passed

1.

Define OR_gate() which takes two inputs, a and b, and returns the outputs specified in the truth table.

You can do this using traditional Python logic. **Or,** using a combination of the other gates you have learned about so far, you can make this in one line!

Hint

OR_gate() returns 1 if either a or b is 1.

Just like with the other gates, push yourself to use the previous gates in creating your OR_gate()!

One way to do this would be:

NAND(NAND(a, a), NAND(b, b))

Copy to Clipboard

Can you see why?

script.py

```
from nand import NAND_gate

from not_gate import NOT_gate

from and_gate import AND_gate


def OR_gate(a, b):

  return NAND_gate(NAND_gate(a, a), NAND_gate(b,b))
```

```
# TEST CASES


print("A: 0, B: 0 | Output: {0}".format(OR_gate(0, 0)))
print("A: 0, B: 1 | Output: {0}".format(OR_gate(0, 1)))
print("A: 1, B: 0 | Output: {0}".format(OR_gate(1, 0)))
print("A: 1, B: 1 | Output: {0}".format(OR_gate(1, 1)))
```