Logic Gate Review - Part II

1 min

Good job! You've started with little bits, and you've built out an entire toolkit of logic gates you now can use to perform more complicated operations. You're well on your way to creating an entire operating system. Watch out, Unix!

We've covered:

- Bits and current
- Truth tables
- NAND gates
- NOT gates
- OR gates
- XOR gates

You've done a great job of creating these gates from scratch in Python! All of the basic gates exist already as bitwise operators in Python. Now that you've done the hard work to understand how each works, you can save time by using these built-in operators:

```
#NOT:
~ x
#E.g. ~1 == 0 and ~0 == 1

#AND:
x & y
#E.g. 1&1 == 1 and 1&0 == 0

#OR:
x | y
#E.g. 1|1 == 1 and 0|0 == 0

#XOR:
x ^ y
#E.g. 1^1 == 0 and 1^0 == 1
```

Copy to Clipboard

Instructions

We've included all the gates we covered in this lesson in **script.py**. Feel free to play around with them and try to make a gate of your own!

We like the Yes-And gate, or the YAND gate. It takes all the inputs, turns them into 1, and then ANDs them together.

script.py

```
from nand import NAND_gate

from not_gate import NOT_gate

from and_gate import AND_gate

from or_gate import OR_gate

from xor_gate import XOR_gate
```