Logic Gates

Binary to Logic

After learning about binary number systems and binary counting, you've got a solid understanding of computers, and how they count. But computers need to do so much more than just count. They need to perform complicated tasks and calculations, with just ones and zeros.

You might be wondering, how do these computers do it? The answer? Boolean logic and logic gates.

Inputs & Outputs

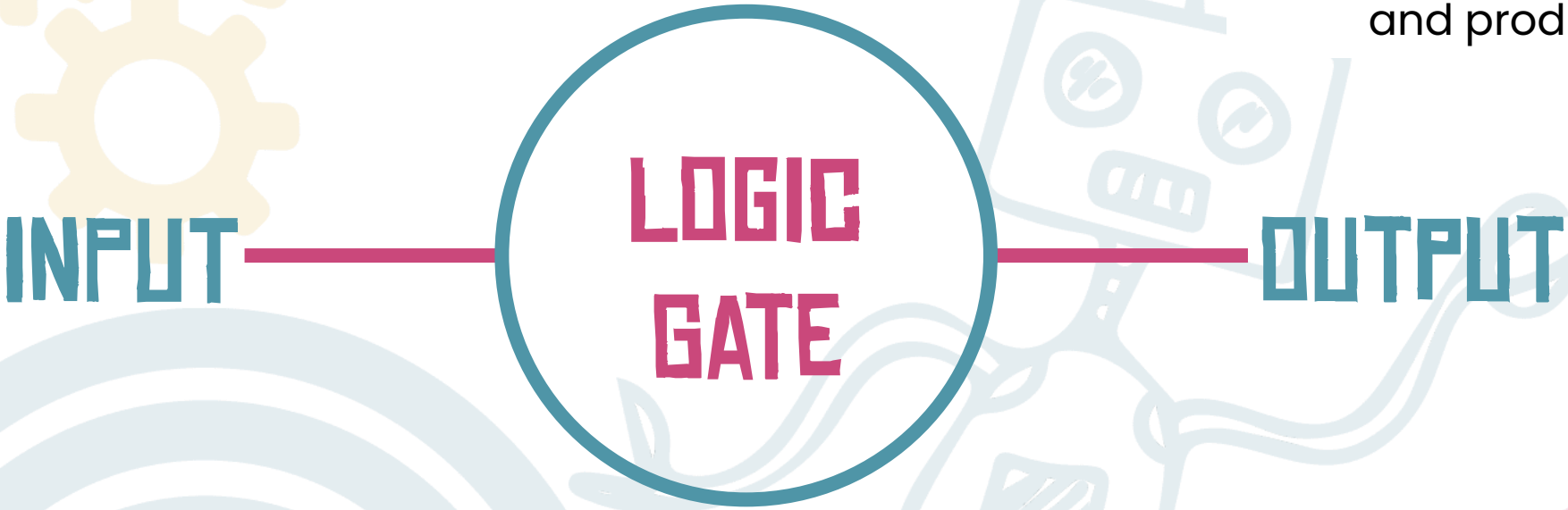
When looking at the inputs and output of logic gates we can use various terminology. We can use **0**, **off** or **false** interchangeably and we could use **1**, **true** or **on** interchangeably.

INPUT	OUTPUT
0	1
Off	On
False	True

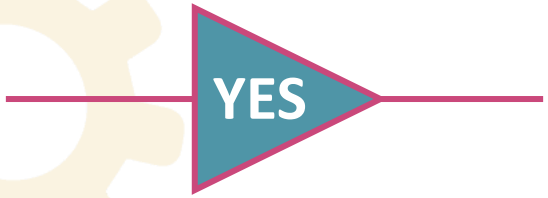
We can absolutely have many different options for input and outputs, but for our lessons we will use only 0 and 1 as the input and output values of the logic gates

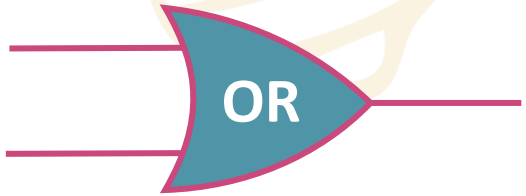
Logic Gates – How Computers Think

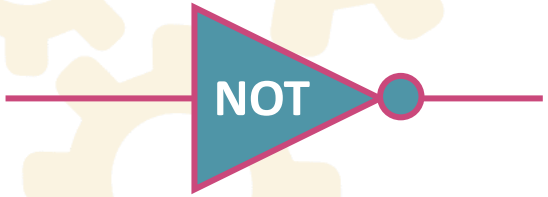
A logic gate is an electronic component in a circuit that performs a specific logical operations based on predetermined rules. It takes one or more binary inputs and produces a binary output.

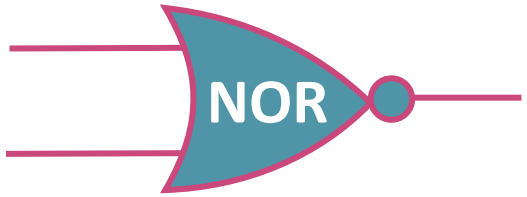


Logic Gates

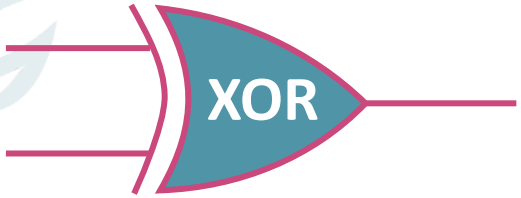


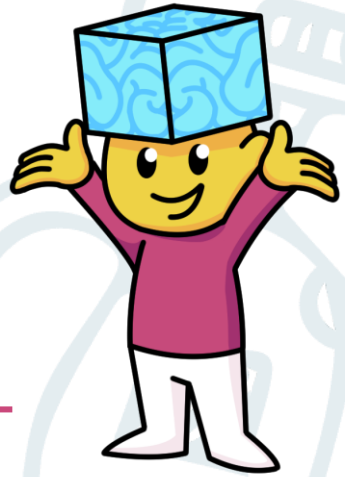


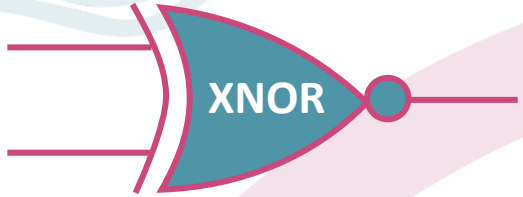




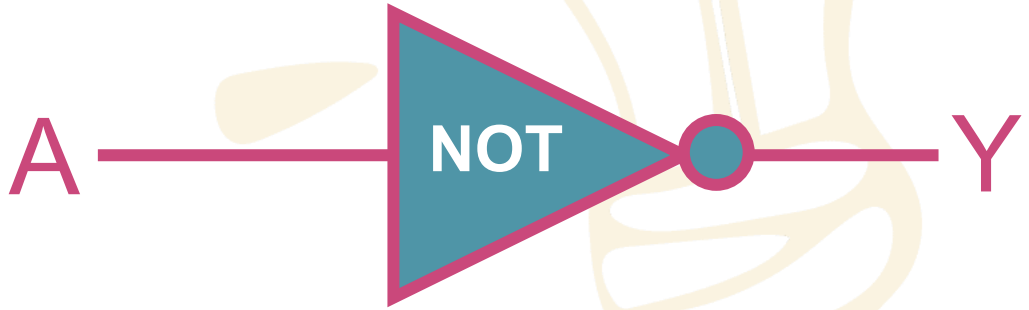
We will be exploring several types of logic gates and the rules that govern what inputs create a specific output.







NOT Gate



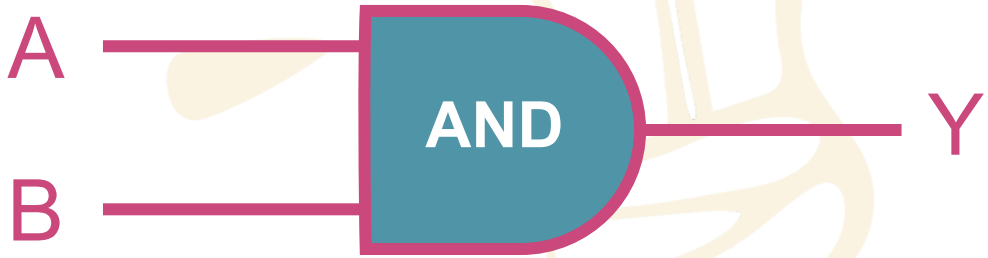
NOT Truth Table	
Input	Output
A	Y
0	1
1	0

← These boxes are called "Truth Tables"

Truth tables can tell us what the output of a specific logic gate will be given all the possible combinations of inputs. With only one input, we only have two possibilities.

A NOT gate inverts or flips whatever the input is.

AND Gate

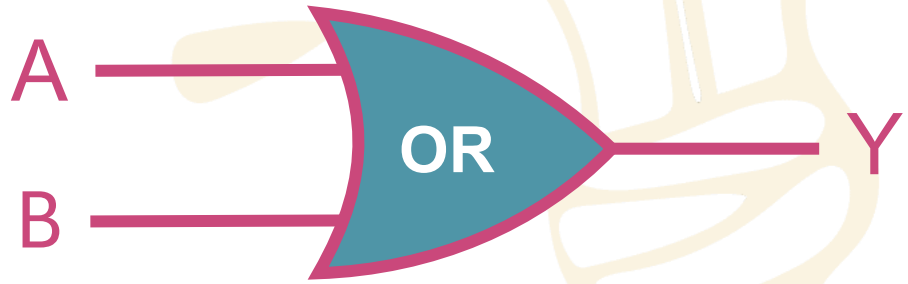


AND Truth Table		
Input		Output
A	**B**	**Y**
0	0	0
0	1	0
1	0	0
1	1	1

With most logic gates, we will have more than one input, causing our truth tables to be bigger. For two inputs, our truth table will have **four** possible outcomes.

An AND gate only is true when input A AND Input B are true.

OR Logic

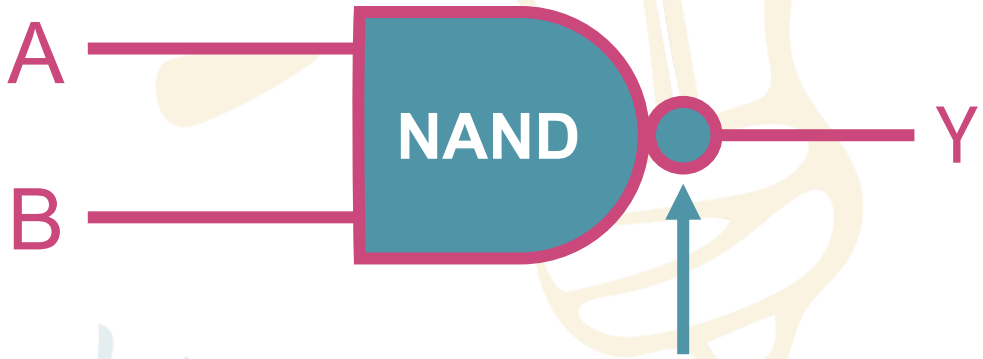


OR Truth Table		
Input		Output
A	**B**	**Y**
0	0	0
0	1	1
1	0	1
1	1	1

An OR gate is true when Input A OR Input B are true. Or if both Input A and Input B are true.

NAND Logic

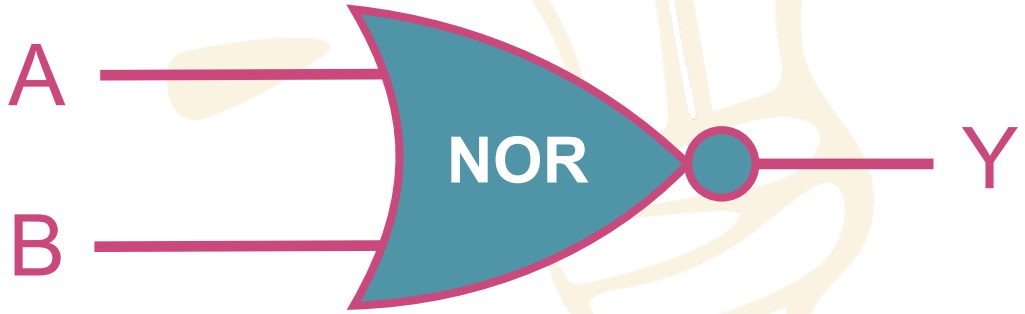
NAND Truth Table		
Input		Output
A	**B**	**Y**
0	0	1
0	1	1
1	0	1
1	1	0



This little circle is an inverter, or basically a mini NOT gate. You can think of a NAND gate as an AND gate and then a NOT gate right after it.

A NAND gate is true when either Input A or Input B are NOT true.

NOR Logic



NOR Truth Table		
Input		Output
A	**B**	**Y**
0	0	1
0	1	0
1	0	0
1	1	0

You can think of a NOR gate as an OR gate and then a NOT gate right after it.

A NOR gate is true only when neither Input A NOR Input B are true.

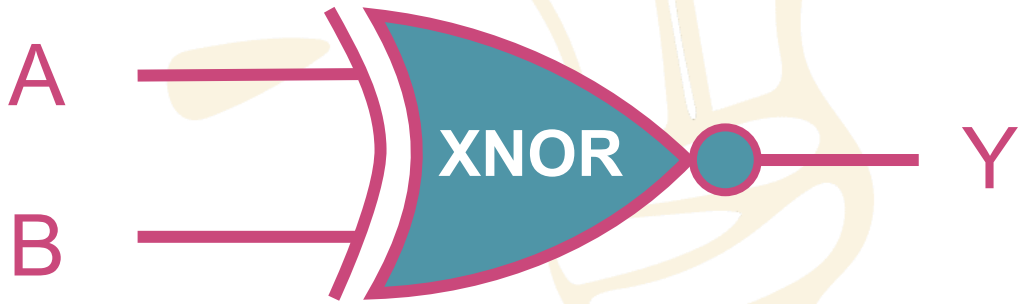
XOR Logic

XOR Truth Table		
Input		Output
A	**B**	**Y**
0	0	0
0	1	1
1	0	1
1	1	0

The X in XOR and XNOR gates stands for "exclusive" and means only an exclusive set of the normal outputs are returned.

A XOR gate is true only Input A OR Input B are true, but not when BOTH are true.

XNOR Logic



XNOR Truth Table		
Input		Output
A	**B**	**Y**
0	0	1
0	1	0
1	0	0
1	1	1

A XNOR gate is true Input A AND Input B are both true or both false.