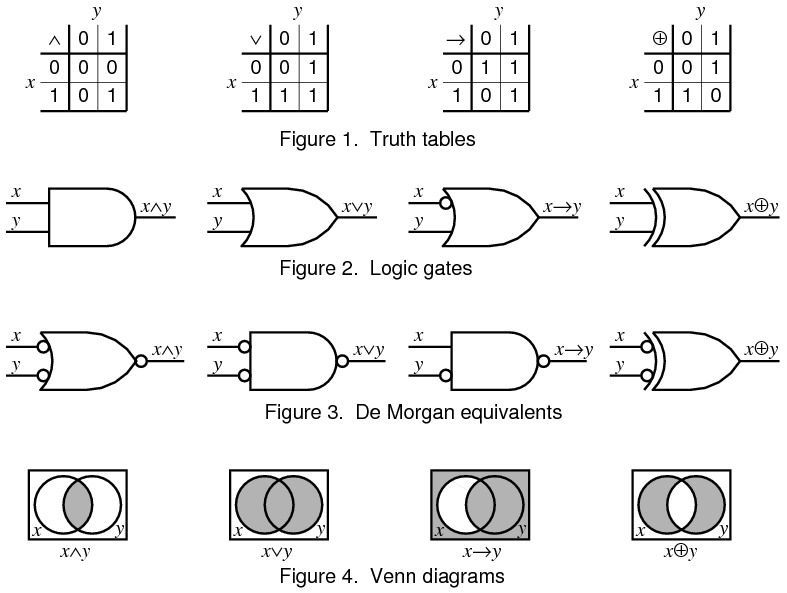
Boolean Logic

The Foundational Logic that is used in computation.

Watch [this delightful video](https://youtu.be/gI-qXk7XojA) from Crash Course to start to understand Boolean logic. If you are endlessly interested in this type of logic, you should try to go and find a corresponding mathematics course on the subject.

Khan academy has also produced a lovely course on “[How Computers Work](https://www.khanacademy.org/computing/computer-science/how-computers-work2#how-computers-work2)” that is probably worth your time. However, to not overwhelm you, let’s just recommend you watch [this video featuring Limor Fried about binary information](https://www.khanacademy.org/computing/computer-science/how-computers-work2/v/khan-academy-and-codeorg-binary-data) before you complete this lesson.



# Example Sketch 1

The if() function is foundational to all Boolean Logic

The important player in boolean logic is the ‘**if(){}’** function. This function has three parts. To see them better, we will need to look at a simple example:

**bool a = true;**  //global variable called ‘a’ to store the variable

**void setup() {**

**Serial.begin(9600);** // communication speed of 9600 baud

**if(a){** //evaluate if the variable ‘a’ is true

**Serial.println("I like cookies.");** // do this if ‘a’ is true

**}**

**}**

**void loop() {**  // the loop isn’t doing anything in this program

**Serial.println("kittens...");**

**delay (1000);**  // wait for one second before looping

**}**

In the program above, the green highlighted area is where you are stating that you would like an if-then logical function to be used. The parentheses that follow, which have their contents highlighted yellow here, are then ‘evaluated’ to be either true or false. This part of the function is sometimes called “the conditional.” This statement that gets evaluated is a space where you can really get creative with the function, but here we keep it simple. In this specific example, the program will evaluate if the variable ‘**a**’ is true. Since the variable has a value of ‘**true**’, the function will evaluate the conditional as true, and move on to the next portion of the function.

The last part of the **if()** function above is the cyan highlighted region. This area contains what to do, if the conditional is evaluated as being true. So, this program, will Serial Print the phrase “I like cookies.” However, if the conditional had been false, then the program would have skipped the cyan region of the If function. Either way, the program will then move on to the **loop**, where it will Serial Print ‘**kittens…**’ forever, every second.

Try taking this program and copying it into your Arduino IDE. run the program, and look at your serial monitor. Then try changing the variable to being false. Or try putting something more complicated in the conditional or the **if()** function’s output. The **if()** function is critical to understand because it is built into nearly every other type of function (even though you may not have to worry about them directly).

Finally, you can include an **}else{** along with an if function. Whatever follows the ‘else’ command will become what happens if the conditional is evaluated to be false. There is also **}elif(){** that works the same, but lets you add a second conditional. The **}elif(){** is for cascading logic.

# Example Sketch 2

Here is an example set of code, that you need to interpret below:

/\* A basic boolean example, that will show some basic logic use. \*/

**bool a = true;**  //global variable called ‘a’ to store the variable

**void setup() {**

**Serial.begin(9600);** // communication speed of 9600 baud

**if(a){** //evaluate if the variable ‘a’ is true

**Serial.println("I like cookies.");** // do this if ‘a’ is true

**}else{** //if the variable ‘a’ is not true

**Serial.println("Ice cream is better.");** // do this if ‘a’ is false

**}**

**}**

**void loop() {**  // the loop isn’t doing anything in this program

**delay (1000);**  // wait for one second before looping

**}**

## Interpretations:

1. What will the output of this program be? Be very specific.
2. Explain the logic of this process in detail.

# Example Sketch 2

Here is an example set of code, that you need to interpret below:

/\* A basic boolean example, that will show some basic logic use. \*/

**bool a = true;**  //global variable called ‘a’ to store the first variable

**bool b = false;** //global variable called ‘b’ to store the second variable

**void setup() {**

**Serial.begin(9600);** // communication speed of 9600 baud

**if(a && b){** // returns ‘true’ if both a and b are true values

**Serial.println("I like cookies.");** // do this when if() returns ‘true’

**}else{** //else{ defines what happens when if() returns ‘false’

**Serial.println("Always hungry...");** // do this when if() returns ‘false’

**}**

**}**

**void loop() {**

**delay (1000);**  // wait for one second before looping

**}**

## Interpretations:

1. What will the Serial Monitor output of this program be? Be very specific.
2. Explain the logic of the program that ultimately leads to the Serial Monitor output.

# Arduino’s Own Explanations

[Boolean Data Type](https://www.arduino.cc/reference/en/language/variables/data-types/boolean/) << a written explanation of the bool variable, and how it works on Arduino

[Bool example sketch](https://www.arduino.cc/reference/en/language/variables/data-types/bool/) << an example implementation of bool variables, used in a sketch

[Logical NOT operator](https://www.arduino.cc/reference/en/language/structure/boolean-operators/logicalnot/) << a written explanation of the operator ‘!’ and how it works on Arduino

[Logical AND operator](https://www.arduino.cc/reference/en/language/structure/boolean-operators/logicaland/) << a written explanation of the operator ‘&&’ and how it works on Arduino

[Logical OR operator](https://www.arduino.cc/reference/en/language/structure/boolean-operators/logicalor/) << a written explanation of the operator ‘||’ and how it works on Arduino

# Write your own Sketches

Write a sketch that proves you understand each of the concepts below. Once sketch per concept. Include screenshots of the sketches and their serial monitor outputs:

## AND

* Concept Explained:
* Screenshot(s)

## NOT

* Concept Explained:
* Screenshot(s)

## OR

* Concept Explained:
* Screenshot(s)

## XOR

* Concept Explained:
* Screenshot(s)