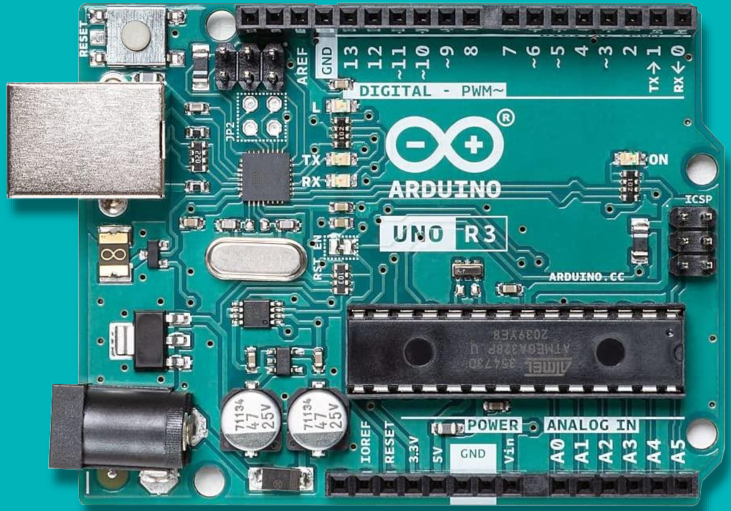
Day 2: Programmable Electronics Workshop

STEM Workshop at ASA Now - HSD2

Projects Overview

- Serial Monitor Hello World
- Built-in LED Blink
- External LED Blink
- Button-Held LED
- Button-Toggle LED
- Multiple LED Blink

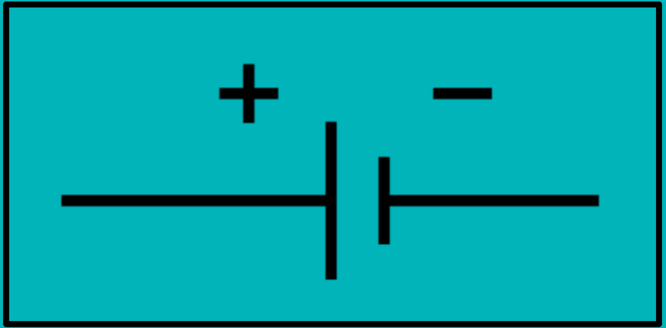


Time-Permitting Challenges

- Arduino LED Dice
- Binary Counter

Section 1: The Basics

Batteries

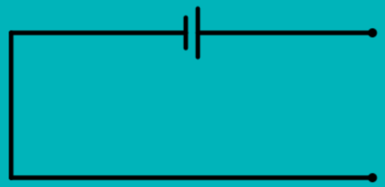


- The Power Source of the circuit
- Creates a “Potential Difference” (aka Voltage) in the circuit it is connected to.
- This Potential Difference causes electrons to flow through the wires of your circuit, provided that your closed loop connects to both the positive and negative terminals of your battery.
- This flow of electrons is known as “Current”, and it can be used to power other devices (such as LEDs and Motors) in the circuit.

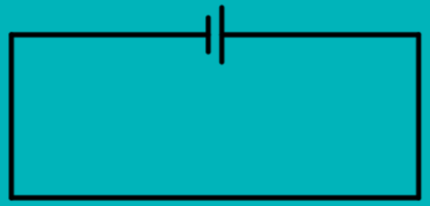
Open vs Closed Circuits

In order to have a functioning circuit, you must have a continuous loop of conductive material connected between the positive terminal of the battery and the negative terminal.

Example of an open circuit:

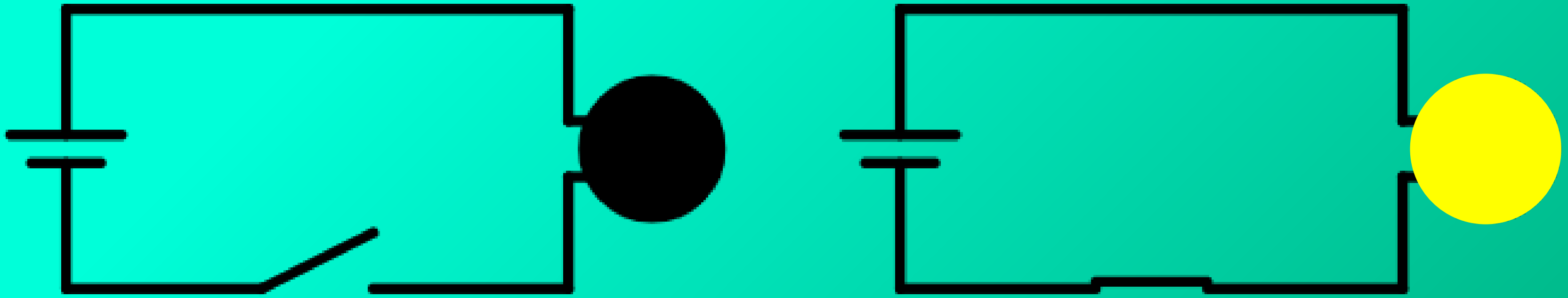


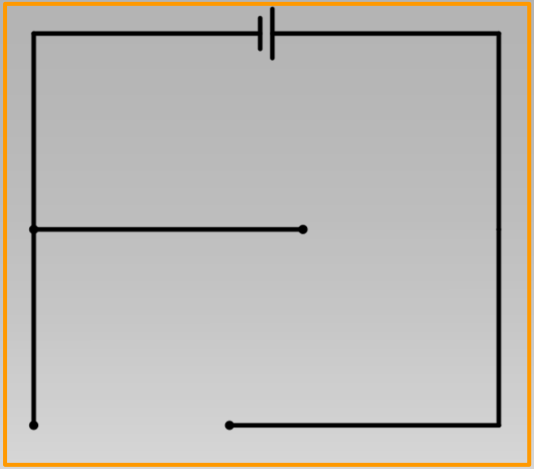
Example of a closed circuit:

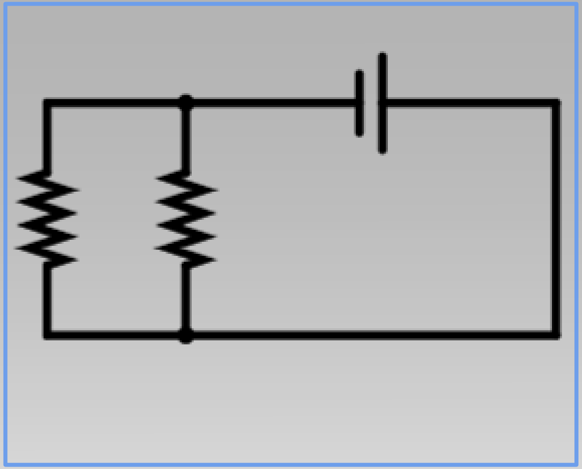


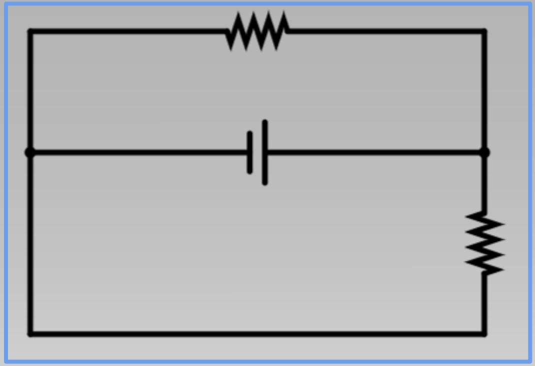
Open vs Closed Challenge!

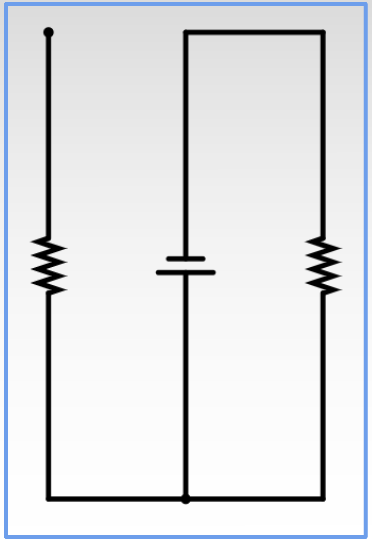
Which of the following circuits would be classified as closed (having at least one complete loop) and which would be classified as open (having no complete loops)?

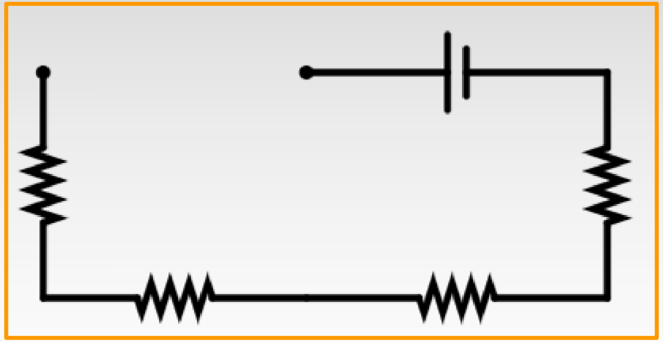


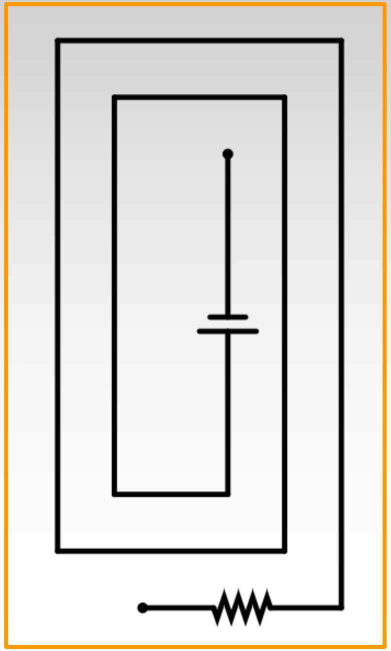


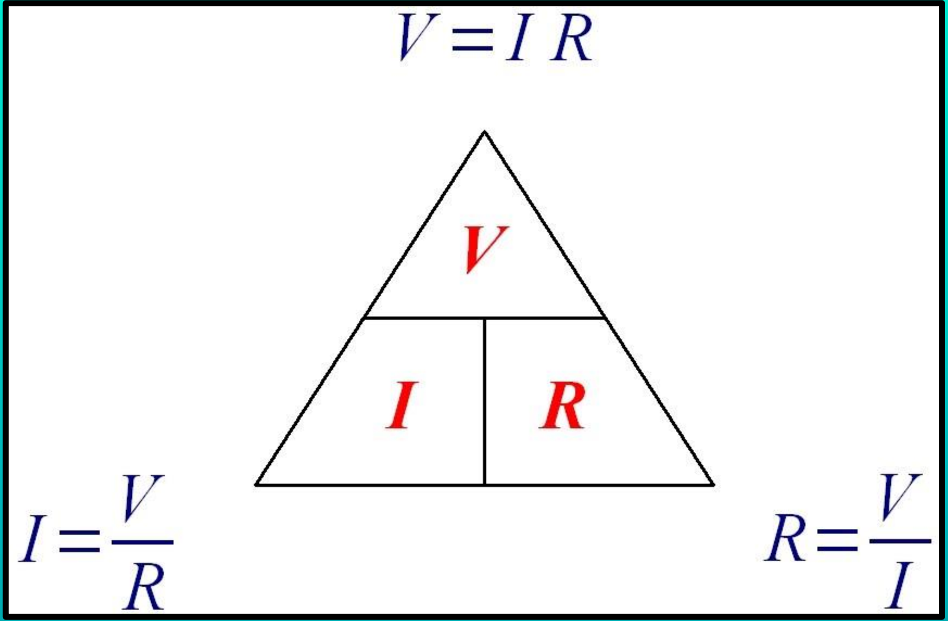












$$V = IR$$

$$I = \frac{V}{R}$$

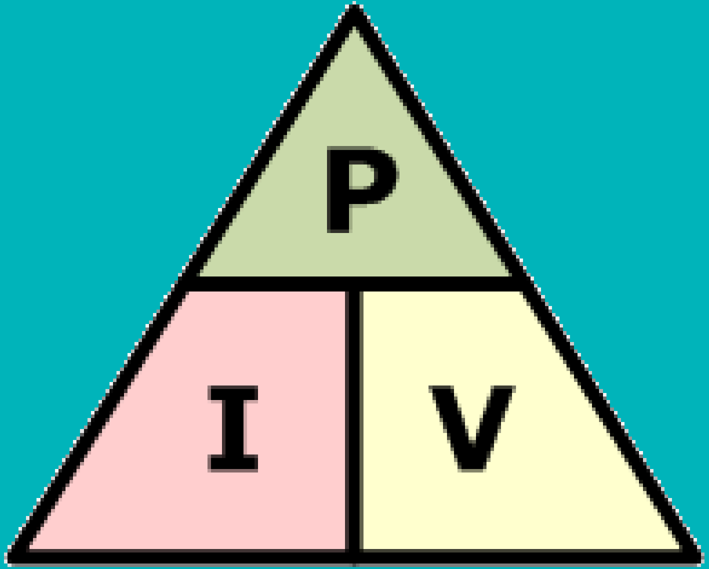
$$R = \frac{V}{I}$$

Voltage (Volts) - The potential difference in a circuit

Current (Amps) - The flow of electrons in a circuit

Resistance (Ohms) - A measure of the opposition to current flow in an electrical circuit

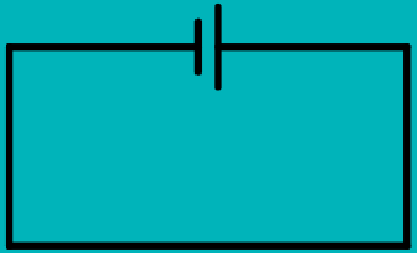
POWER!!

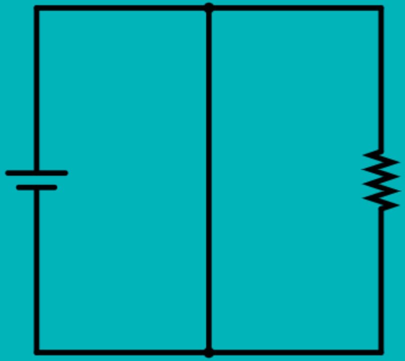


- **Power = Current x Voltage**
- Determines the brightness of a bulb is
- Measure in Watts (Whats?)

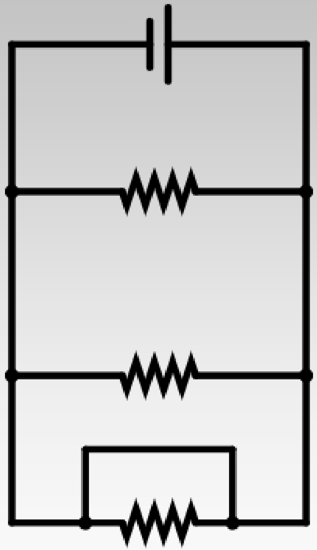
Short Circuits

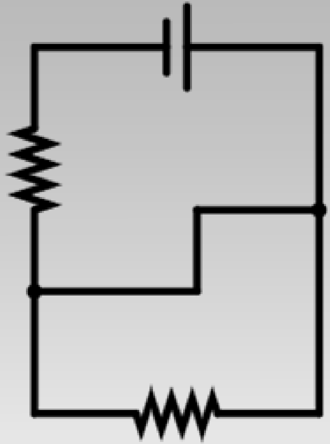
- Created when a battery is connected in a circuit with no resistive elements.
  - This creates a near-infinite current in the circuit
- This can be extremely dangerous
  - Best-case scenario: the battery rapidly discharges, sharply reducing the battery’s lifespan to only a few hours.
  - Worst-case scenario: the battery and other components rapidly heat up, causing a fire or explosion.

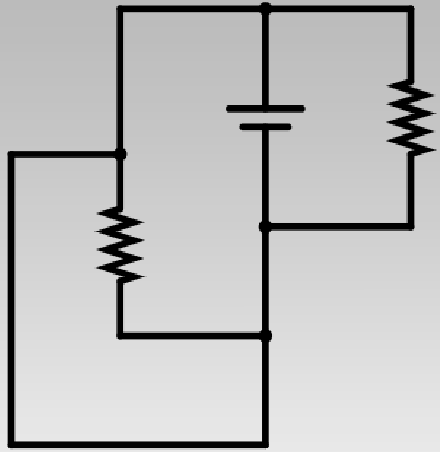


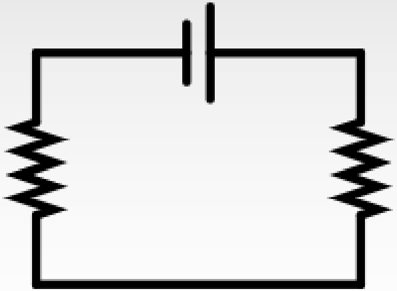


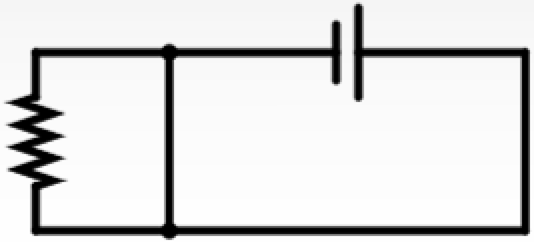
Is it Shorted Challenge!



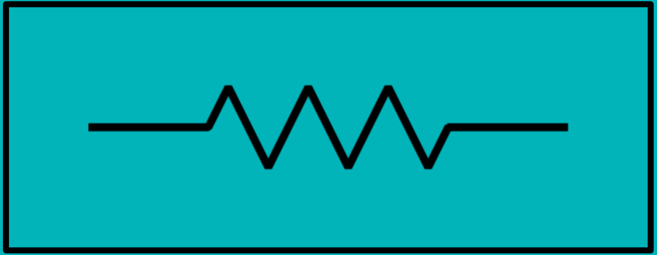


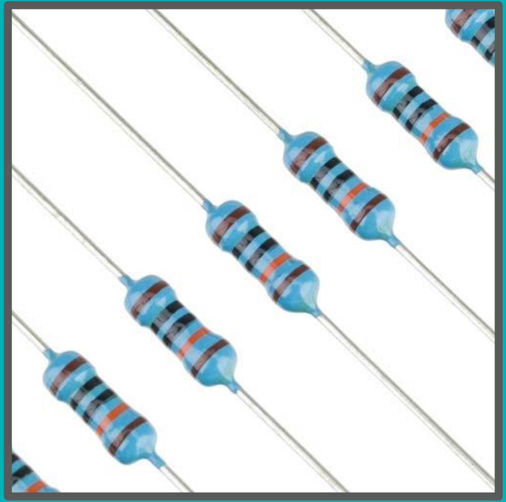






Resistors

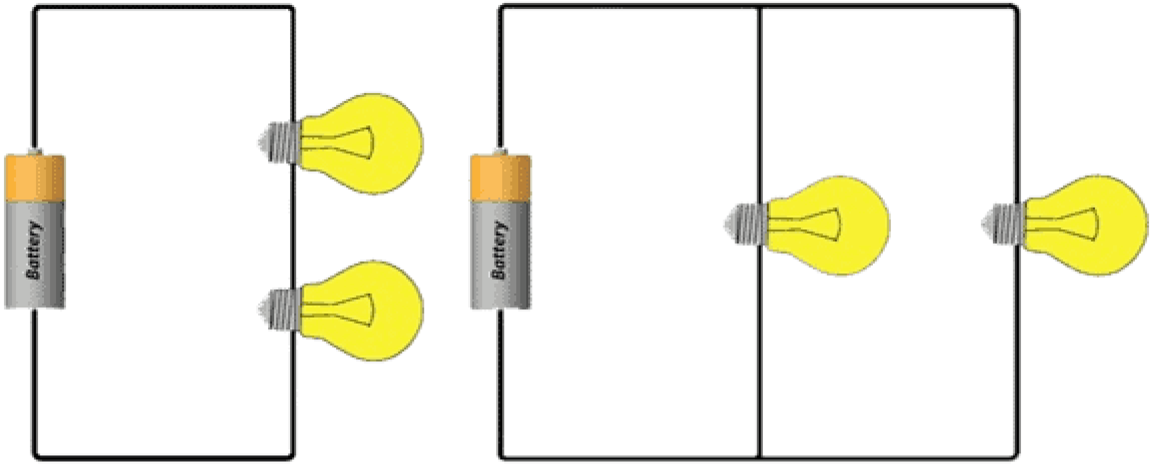




- Resist the flow of current in a circuit
- **Why would we want to do this?**
  - Safety
    - Some components simply cannot handle having too much current flowing through them at once, and will burn if this occurs
  - Control
    - We can use special Variable resistors, called "Potentiometers" to physically control the current flowing through a circuit, and with it, the output of various devices, including LEDs and Motors.

LEDs

- LED stands for “Light Emitting Diode”
- Diodes are a special type of electronic component that only allow for current to flow through them in one direction.
    - If you think you have built a circuit correctly, but the LED is not glowing, try flipping it around to check if you have just accidentally inserted it in the wrong direction.
- LEDs are very prone to burning out and becoming permanently unusable when connected to a battery without any resistive elements.
    - When working with LEDs, always make sure it is next to a resistor!



Comparison of Series and Parallel

Section 2: Arduino

Definition of a computer (Review)

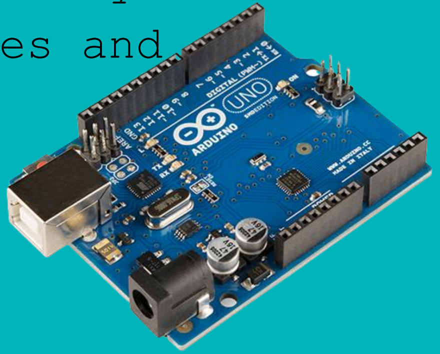
- Anyone remember?
- **A computer is a machine or device that performs processes, calculations and operations based on instructions provided by a software or hardware program. It has the ability to accept data (input), process it, and then produce outputs. - Technopedia**

What is Arduino?

Arduino is many things:

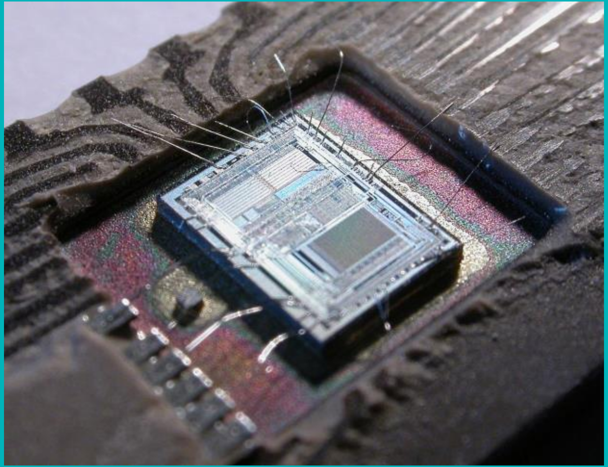
- A company that develops open-source MCUs and tools for them.
- A way to refer to the boards and components this company designs and produces.
- A word synonymous with inexpensive, easy-to-use, hobbyist MCUs (Similar to what “Google” is of Search Engines and “Kleenex” is of Tissue Paper).

We will be using an Arduino Uno for this workshop. →

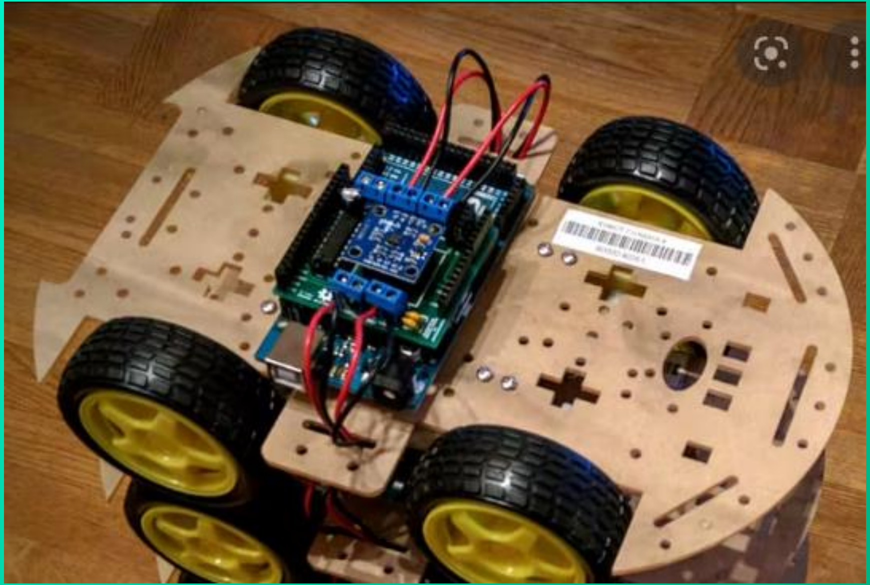


What is a Microcontroller?

A Microcontroller (also known as a Microcontroller Unit/ MCU) is a small computer designed for embedded applications, in contrast to the “Microprocessors” of most personal computers.



What you can do with arduino



Setup time!

1. Connect the USB port to the Arduino and your computer.
2. Open the Arduino IDE (if it is not already open)

Test Program (Project 1)

After you have selected the connected board, copy the following code into the IDE.

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600); // Open a serial connection that communicates at 9600 bits per second
  Serial.println("Hello World!"); // Print "Hello World" (without the quotes) to the serial monitor
}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Then, click the “upload” button and open the “Serial Monitor” with button above and to the right of the code area of the IDE.

Test Program Explained

The test example uses the Serial Monitor to quickly test if the Arduino is successfully connected and communicating without having to connect any other components.

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600); // Open a serial connection that communicates at 9600 bits per second
  Serial.println("Hello World!"); // Print "Hello World" (without the quotes) to the serial monitor
}

void loop() {
  // put your main code here, to run repeatedly:

}
```

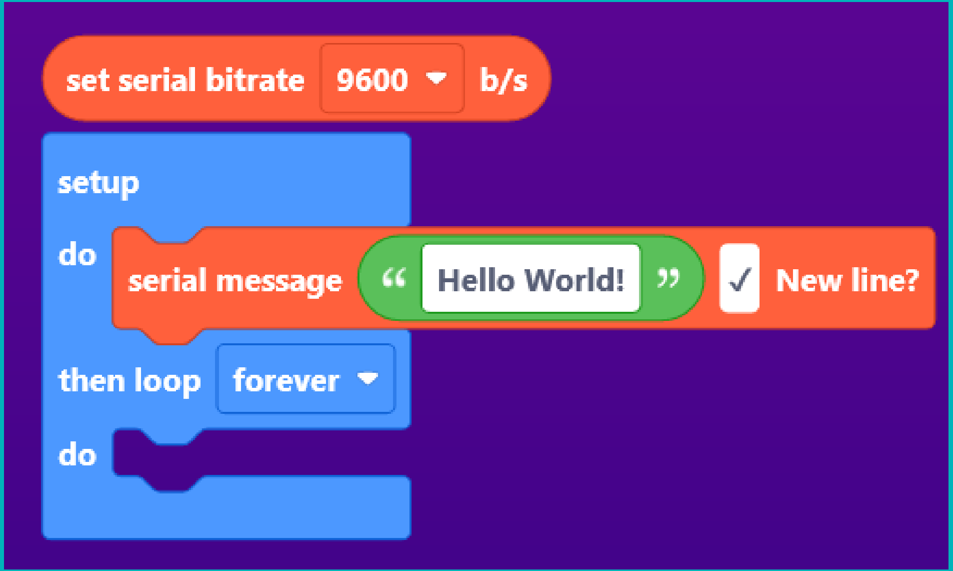
Edukits Drag and Drop Arduino Block Code

Open your web browsers and navigate to https://edukits.co/code/

You can use this tool to generate Arduino-compatible code with drag-and-drop code blocks (similar to what many of you likely did Monday during the Hour of Code).

You can then copy and paste this code into the Arduino IDE so that you can upload it to your Arduino Boards.

Recreating the Test Program with Block Code



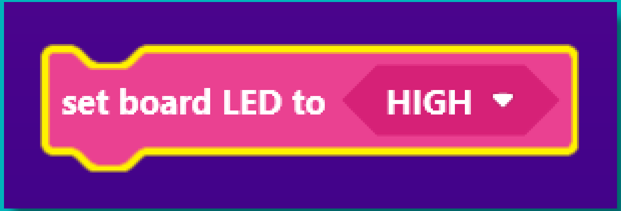
```
set serial bitrate 9600 ▾ b/s
setup
do  serial message “ Hello World! ” ✓ New line?
then loop forever ▾
do
```

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Serial.println("Hello World!");
}



void loop() {
  // put your main code here, to run
repeatedly:

}
```

The Built-in LED

The Arduino Uno (along with most similar modern hobbyist Microcontrollers) come with an LED embedded on the device. We can control this LED without having to add other components to the system.

You can control this with the "set board LED to <___> " block inside of the "Input/ Output" Block folder on the edukit IDE.



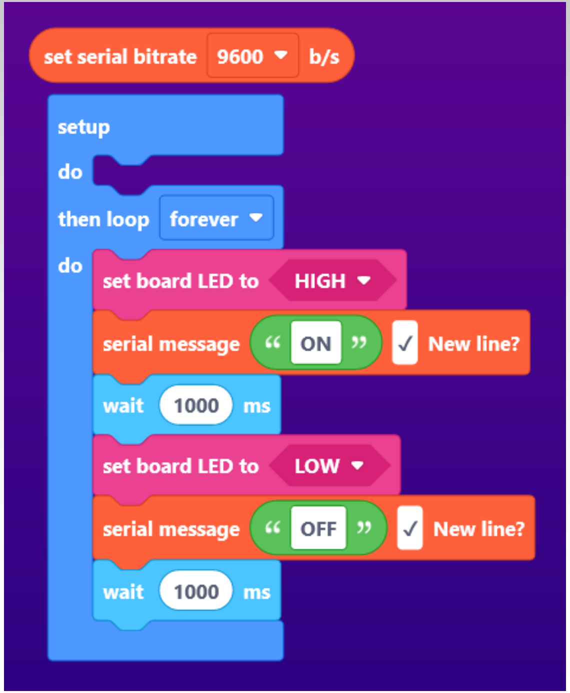
set board LED to HIGH

Project 2: “Built-in LED Blink”

Goal: Write a program that “Blinks” the Built-in LED of your Arduino Board, infinitely turning the LED on for one second before turning it off for one second.

Project 2: “Built-in LED Blink” Code

Edukit



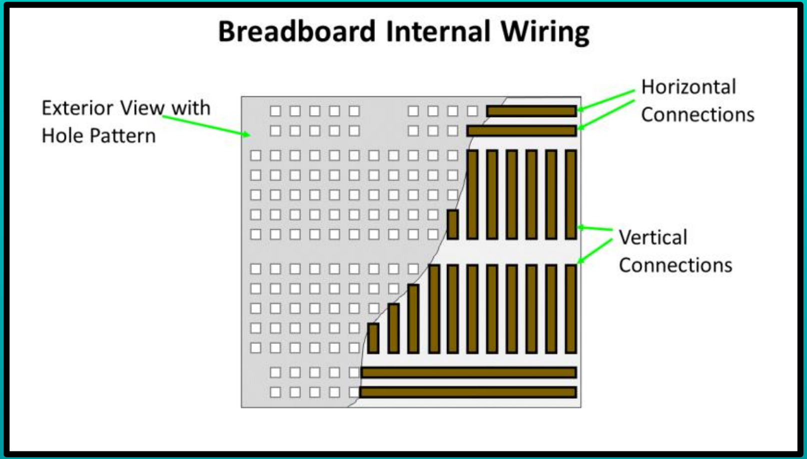
```
set serial bitrate 9600 ▾ b/s
setup
do
then loop forever ▾
do
    set board LED to HIGH ▾
    serial message “ ON ” ✓ New line?
    wait 1000 ms
    set board LED to LOW ▾
    serial message “ OFF ” ✓ New line?
    wait 1000 ms
```

Arduino

```
void setup() {
  Serial.begin(9600);
  pinMode(LED_BUILTIN, OUTPUT);
}
void loop() {
    digitalWrite(LED_BUILTIN, HIGH);
    Serial.println("ON");
    delay(1000);
    digitalWrite(LED_BUILTIN, LOW);
    Serial.println("OFF");
    delay(1000);
}
```

Breadboards

- The base of circuits that we will build.
- Plastic boxes with holes that are interconnected underneath with wires.
- Insert devices in specific sequence and they are connected.

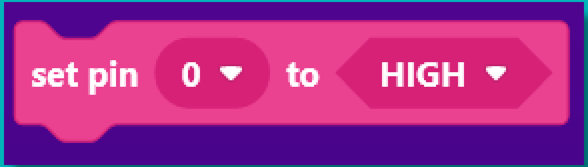


External LEDs

In addition to the built-in LED on the Arduino Uno, we can also connect external LEDs to use with the board.

We can control these LEDs by using a wire to connect it to a numbered GPIO pin on the Arduino board.

Then, use the "set pin (##) to <___>" block located inside of the "Input/ Output" Block folder to control the pin.

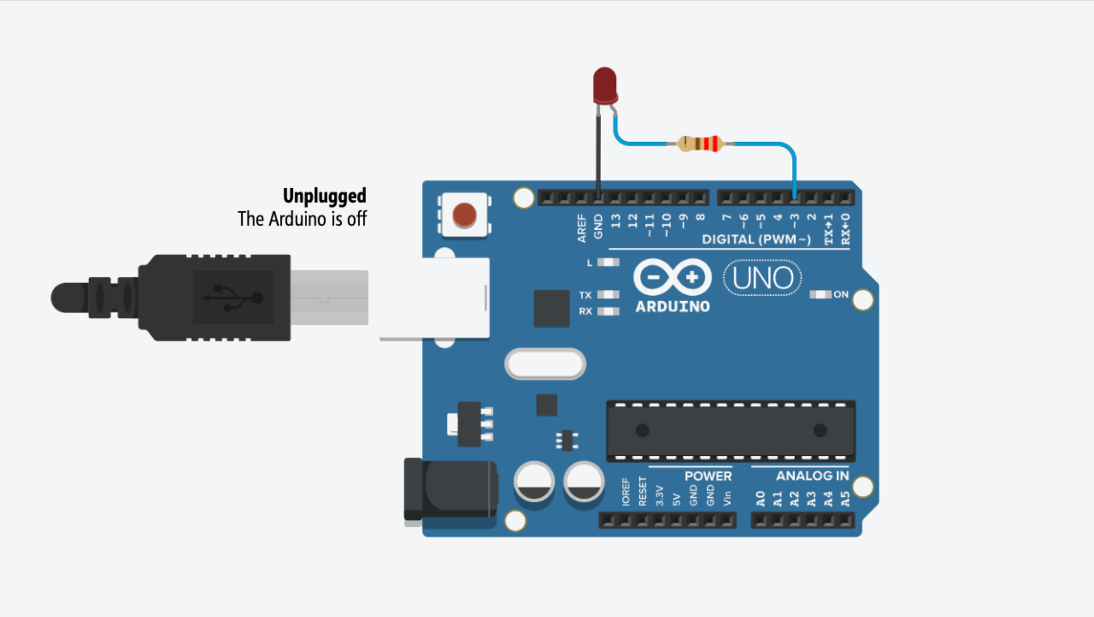


set pin 0 ▾ to HIGH ▾

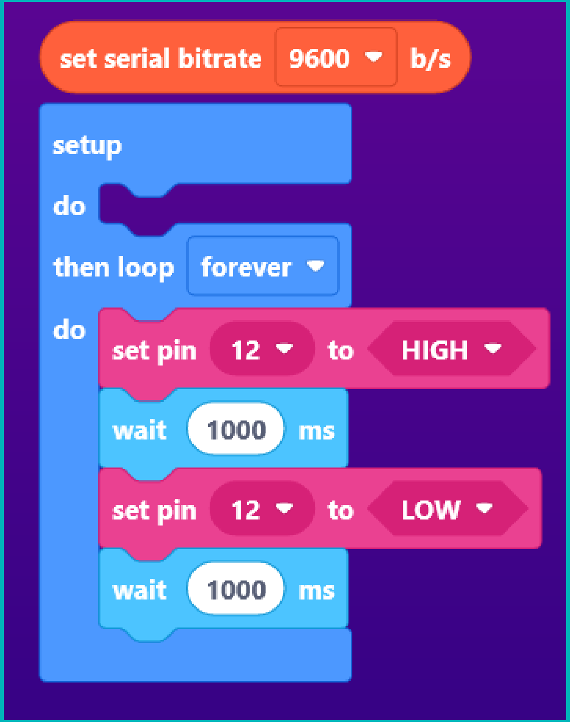
Note: Avoid using pins 0, 1, and 13 as GPIO pins with the Arduino Uno, as those are reserved for other things.

Project 3: “External LED Blink”

Goal: Recreate the Blink program from Project 2, but instead blinking an external LED instead of the built-in one this time.



Project 3: "External LED Blink" Code



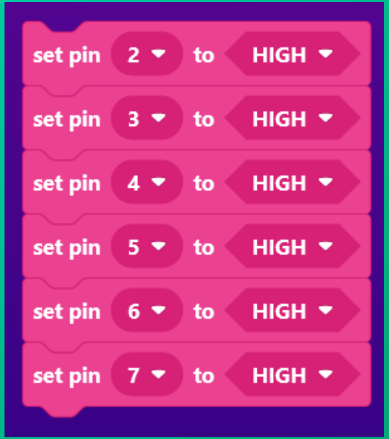
```
set serial bitrate 9600 b/s
setup
do
then loop forever
do
    set pin 12 to HIGH
    wait 1000 ms
    set pin 12 to LOW
    wait 1000 ms
```

```
void setup() {
  Serial.begin(9600);
  pinMode(12, OUTPUT);

}

void loop() {
    digitalWrite(12, HIGH);
    delay(1000);
    digitalWrite(12, LOW);
    delay(1000);
}
```

Multiple External LEDs

By connecting multiple different LEDs to multiple different GPIO pins on the Arduino board, we can control them independently using multiple “set pin” blocks.

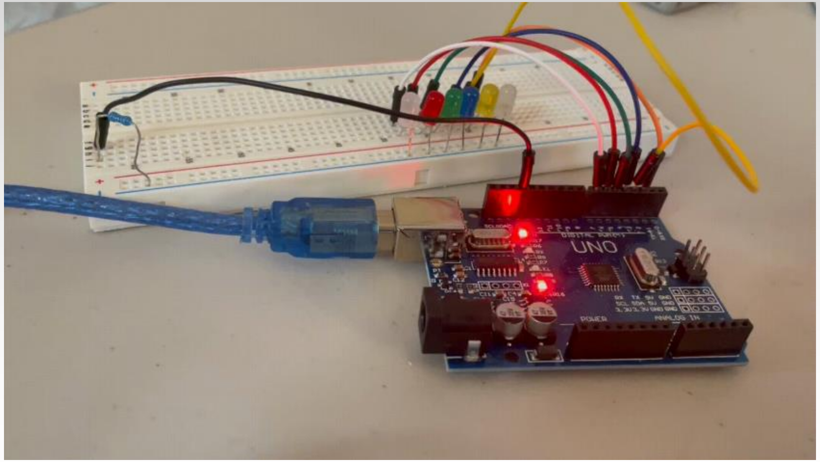
I recommend using pins 2-7 for this, as those are the lowest continuous set of 6 pins that the Arduino Uno does not reserve for other things (such as the USB communication on pins 0 & 1, or the built-in LED on pin 13)



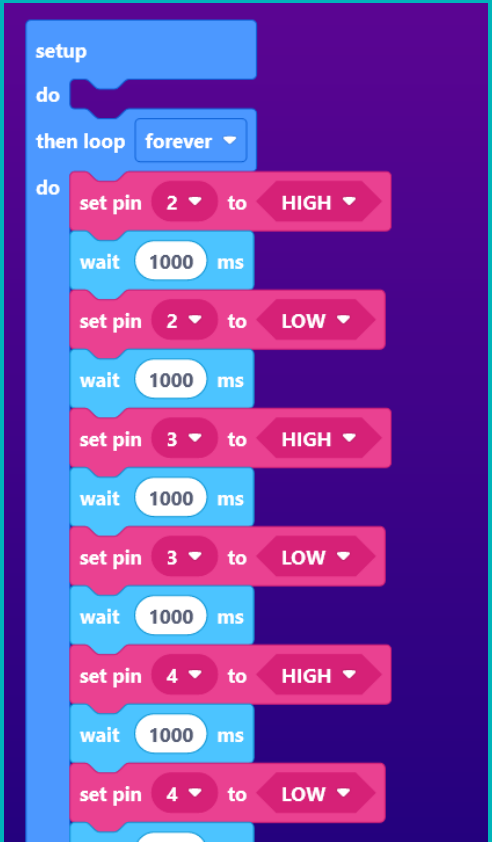
```
set pin 2 to HIGH
set pin 3 to HIGH
set pin 4 to HIGH
set pin 5 to HIGH
set pin 6 to HIGH
set pin 7 to HIGH
```

Project 6: Multiple LED blink

Goal: Build a circuit where multiple (6) external LEDs “blink” in order.



Project 6: “Multiple LED blink” Code



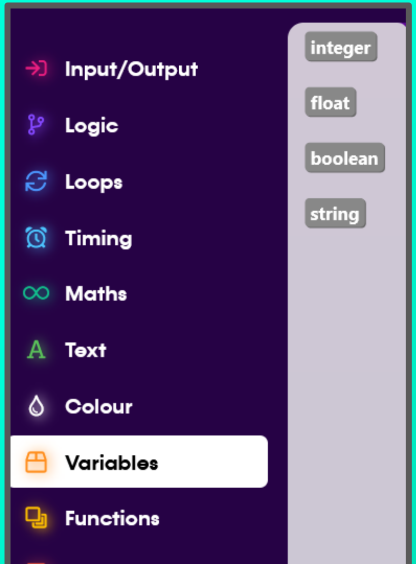
```
setup
do
then loop forever
do  set pin 2 to HIGH
    wait 1000 ms
    set pin 2 to LOW
    wait 1000 ms
    set pin 3 to HIGH
    wait 1000 ms
    set pin 3 to LOW
    wait 1000 ms
    set pin 4 to HIGH
    wait 1000 ms
    set pin 4 to LOW
```

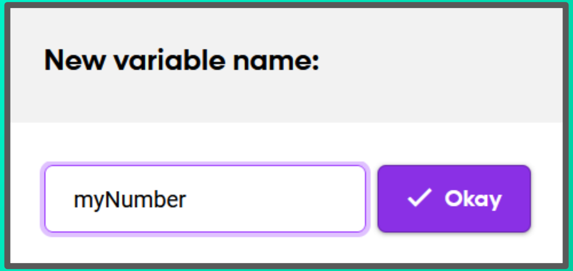
And so on...

```
void setup() {
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  // And so on
void loop() {
    digitalWrite(2, HIGH);
    delay(1000);
    digitalWrite(2, LOW);
    delay(1000);
    digitalWrite(3, HIGH);
    delay(1000);
    digitalWrite(3, LOW);
    delay(1000);
    // And so on
}
```

Variables

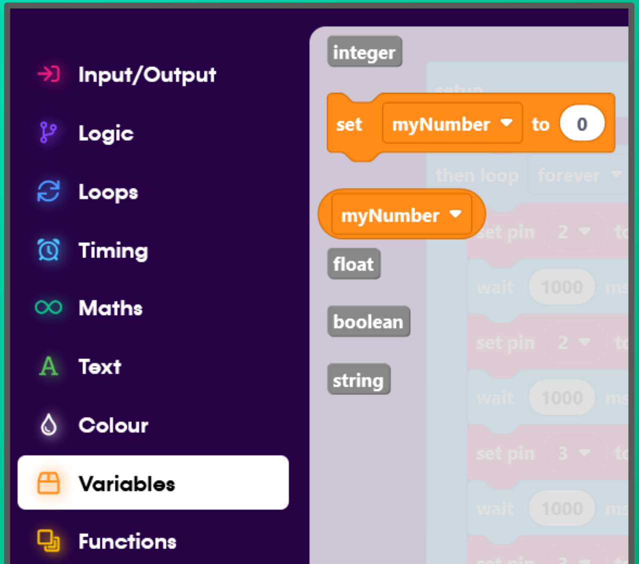
Variables in Computer Programming let you save a value once (such as a number [ex. 14], text [ex. "Hello123"], or Boolean [ex. True/ False]) and use and modify it later in your program.




New variable name:

myNumber

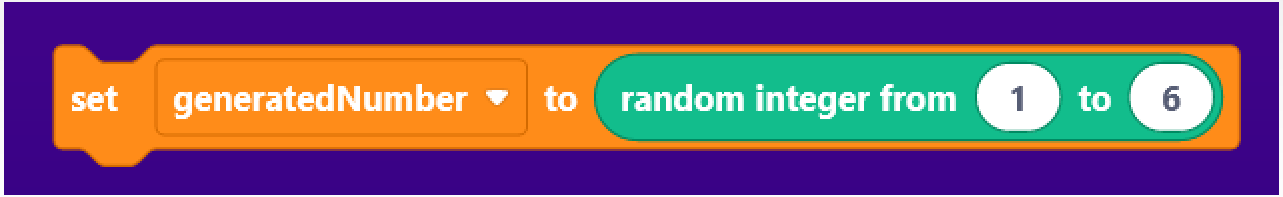
✓ Okay



Challenge 1: LED Dice

Task: Create a program that randomly generates a number from 1-6, then displays the corresponding number of LEDs.

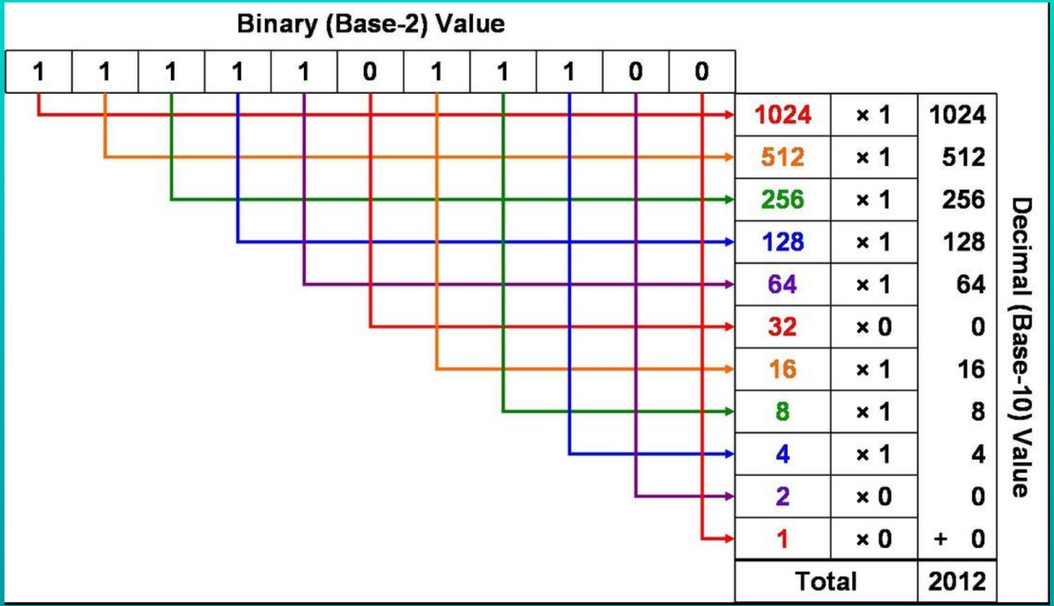
Tip: To generate a random number, try to use the “random integer from __ to __ block” and saving it’s output as a variable.



set generatedNumber ▾ to random integer from 1 to 6

Bonus: What is Binary?

In most situations, it is easiest for humans to utilize the Base-10 number system. However, computers are able to more easily understand Base-2 (also known as Binary).



Binary (Base-2) Value

1	1	1	1	1	0	1	1	1	0	0

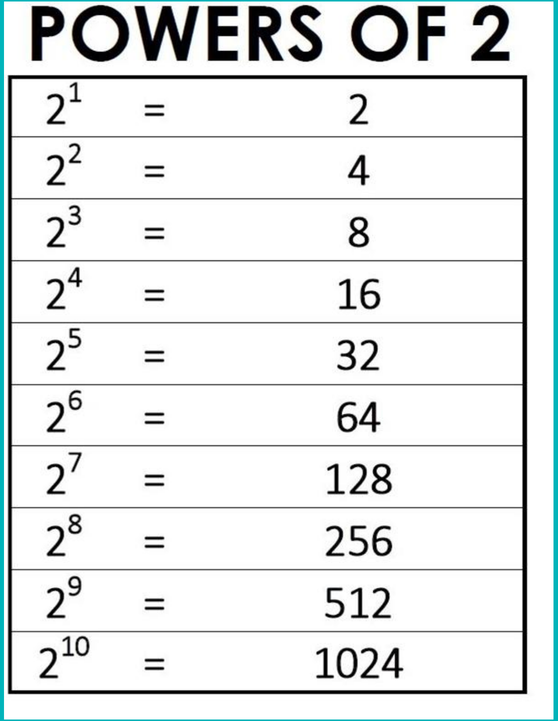
Decimal (Base-10) Value

1024	× 1	1024
512	× 1	512
256	× 1	256
128	× 1	128
64	× 1	64
32	× 0	0
16	× 1	16
8	× 1	8
4	× 1	4
2	× 0	0
1	× 0	+ 0
Total		2012

How to convert Base-10 to Binary

Ex: Convert 7 from Base-10 to Binary

1. Start by finding the greatest power of 2 that is still less than the number you are converting from.
2. Mark a '1' in the place value slot of the binary number you are building corresponding to the exponent of the power of 2 you found.
3. Subtract the found power of two from your original number, and repeat from step one until your remaining number is 0.



POWERS OF 2

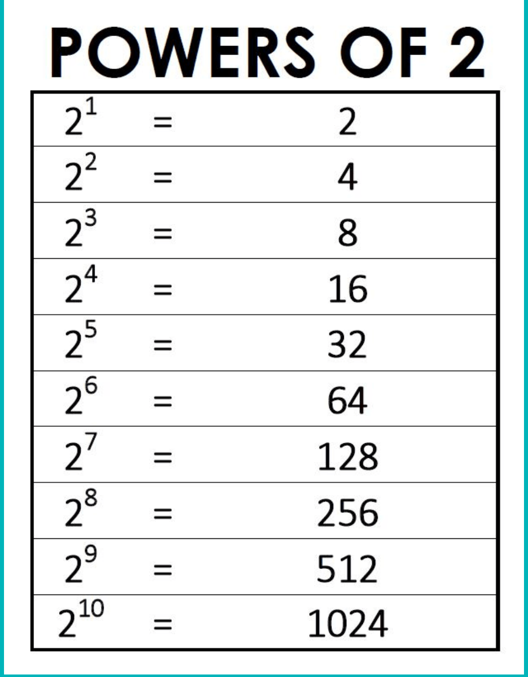
2^1	=	2
2^2	=	4
2^3	=	8
2^4	=	16
2^5	=	32
2^6	=	64
2^7	=	128
2^8	=	256
2^9	=	512
2^{10}	=	1024

Note: $2^0 = 1$

Practice

Convert the following numbers from Base-10 to Binary:

- 3
- 8
- 2
- 0
- 10
- 16
- 18
- 21



POWERS OF 2

2^1	=	2
2^2	=	4
2^3	=	8
2^4	=	16
2^5	=	32
2^6	=	64
2^7	=	128
2^8	=	256
2^9	=	512
2^{10}	=	1024

Challenge 2: Binary Counter (hard)

Goal: Create a 6-digit binary counter using 6 LEDs.

More Details: Display the binary equivalent of a Base-10 number, which increments every second.

Credits

Thank you to Benjamin Arbit and Manish Reddy for helping with the content and formatting of this slide show.