



Armstrong

School Program 2023-2024

Orientation Session



Armstrong

entertainment meets education

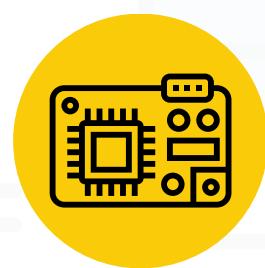
REGISTER NOW



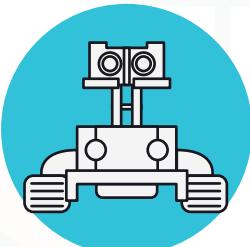
<https://armstrongedu.com/>



Lesson Content



Introduction to Robotics



Robot Parts



Robots Classification



Getting started with mblock

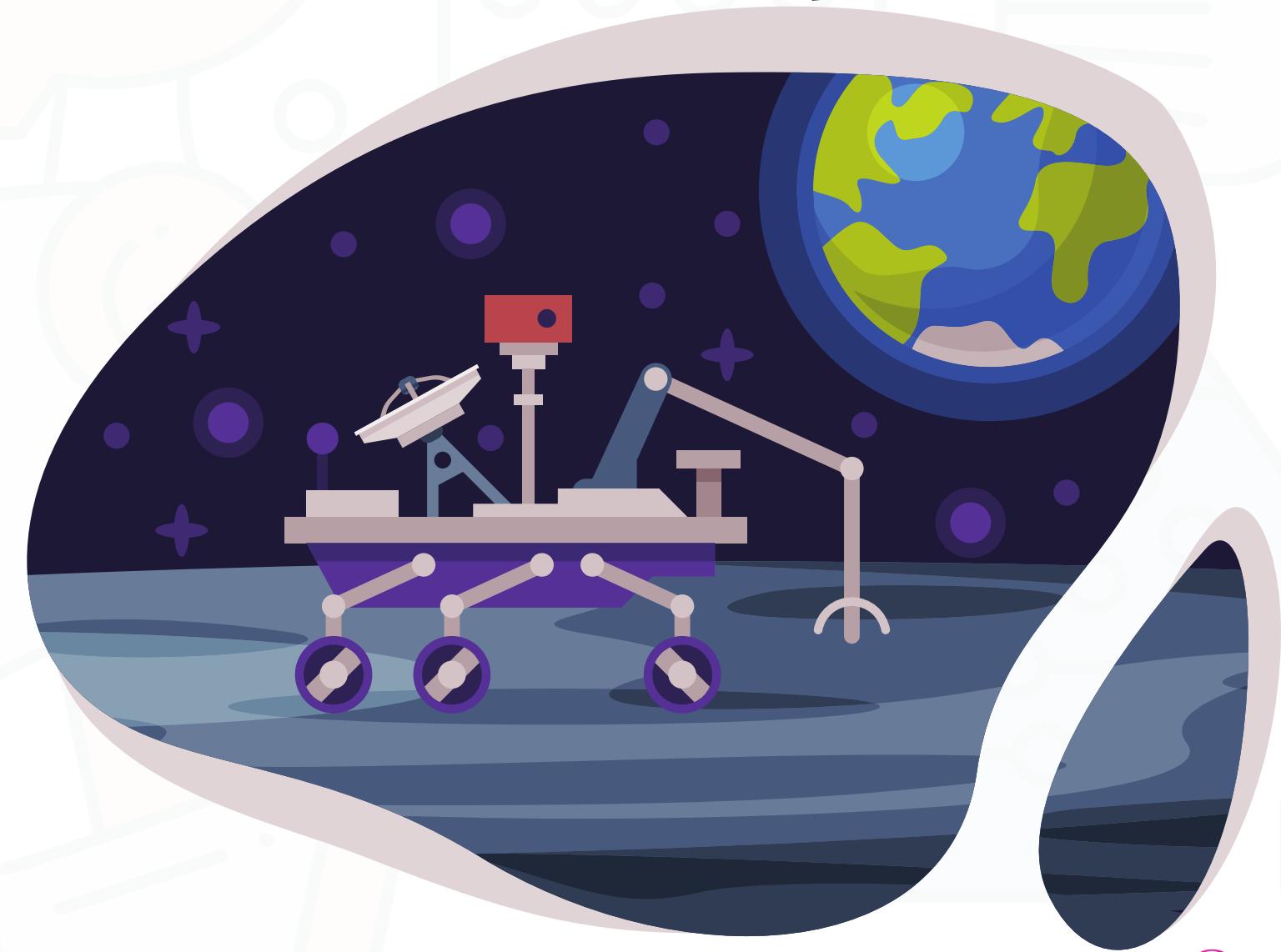
Introduction

What is a robot?

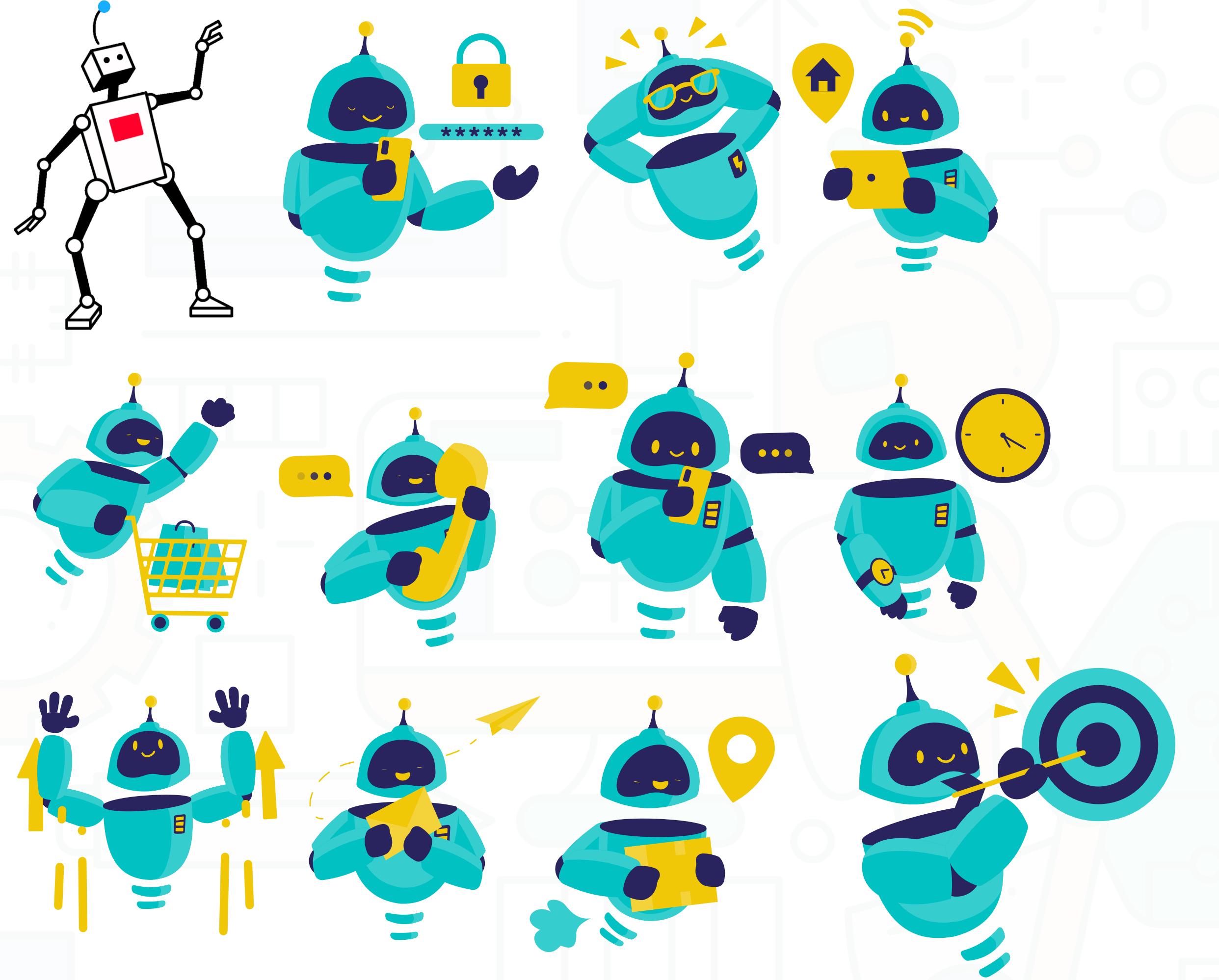
A robot is an electromechanical device that can sense its environment, process it, and react to its environment.

What is robotics?

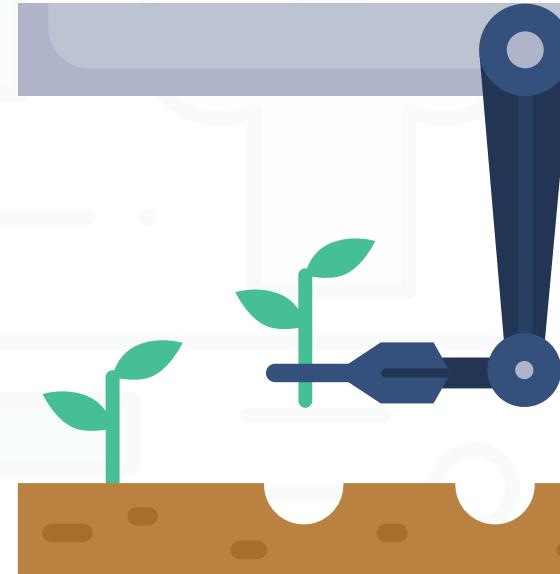
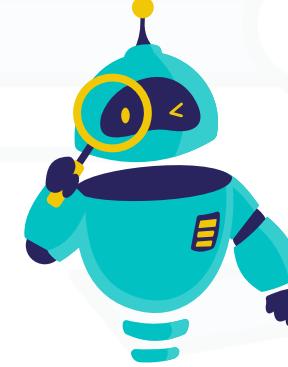
Robotics is the engineering of robots. It combines **mechanics**, **electronics**, and **software**.



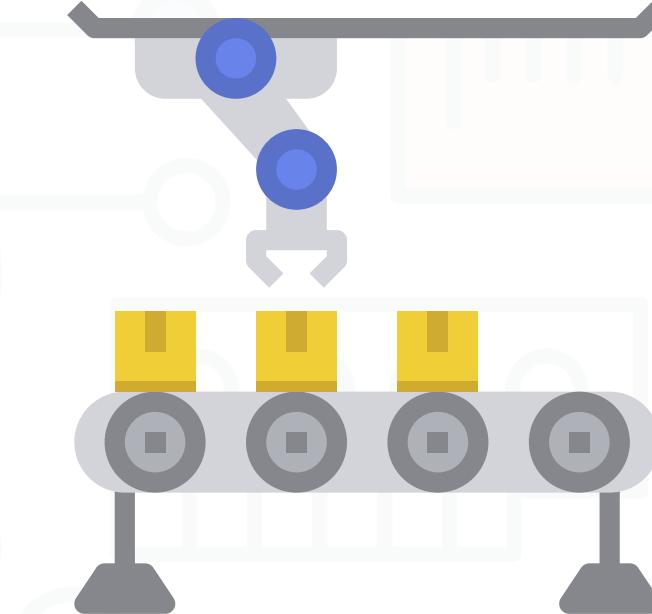




Uses of Robots



Agriculture



Manufacturing

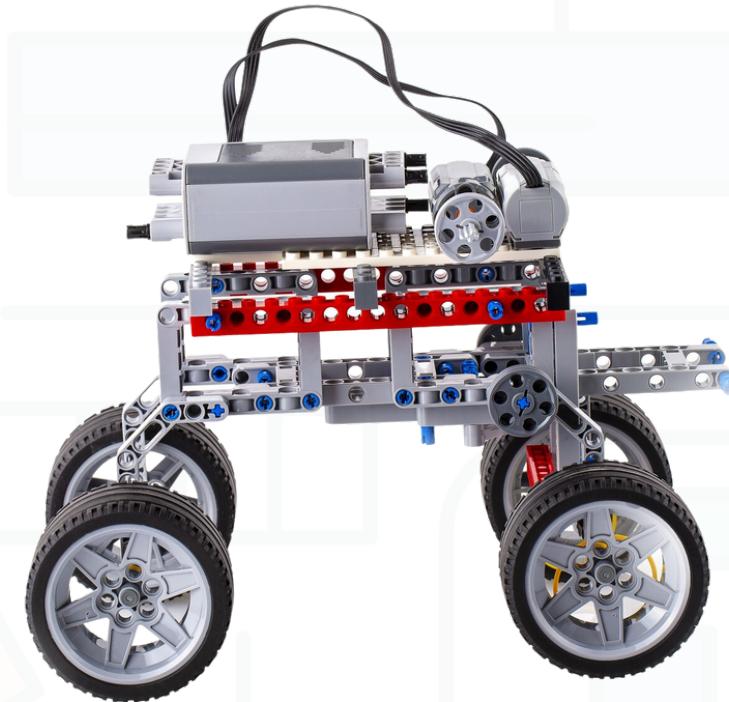


Medical

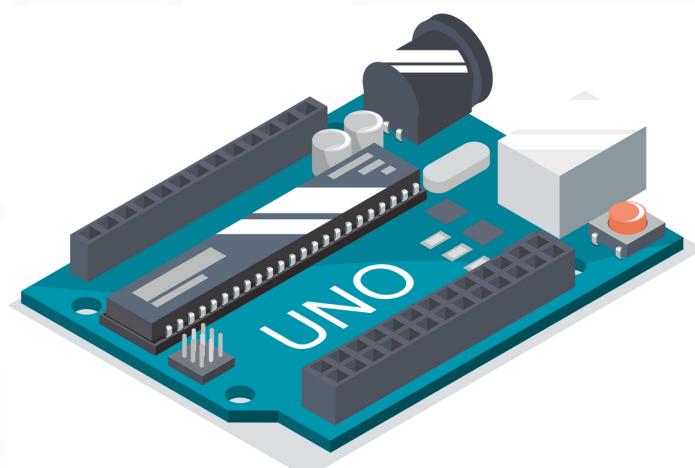


Military

Robot Parts



Chassis



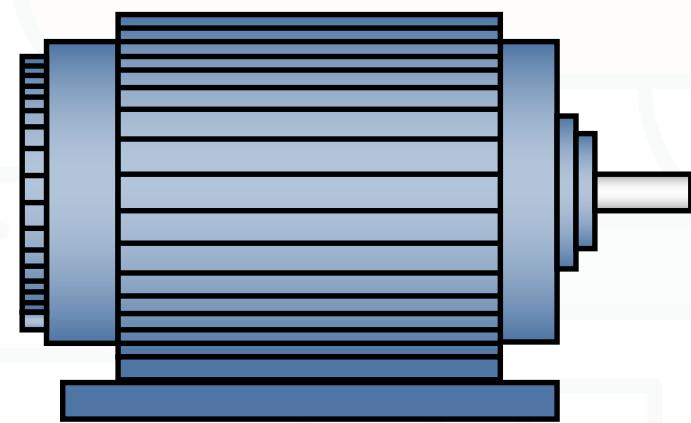
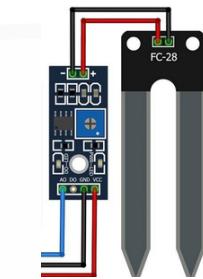
Microcontroller



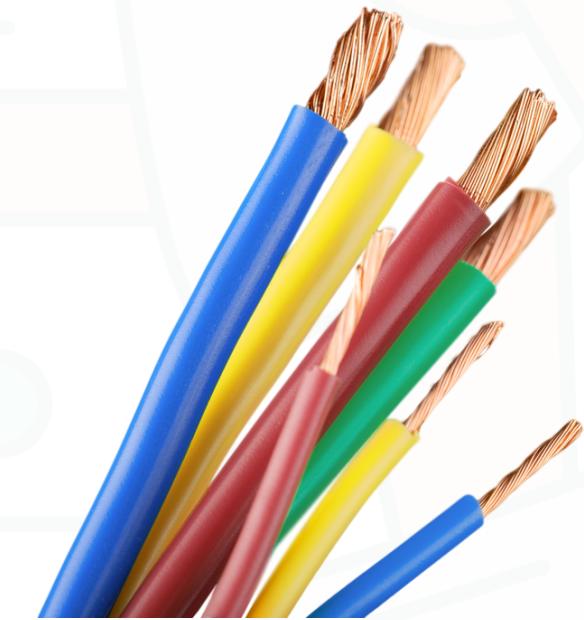
Batteries



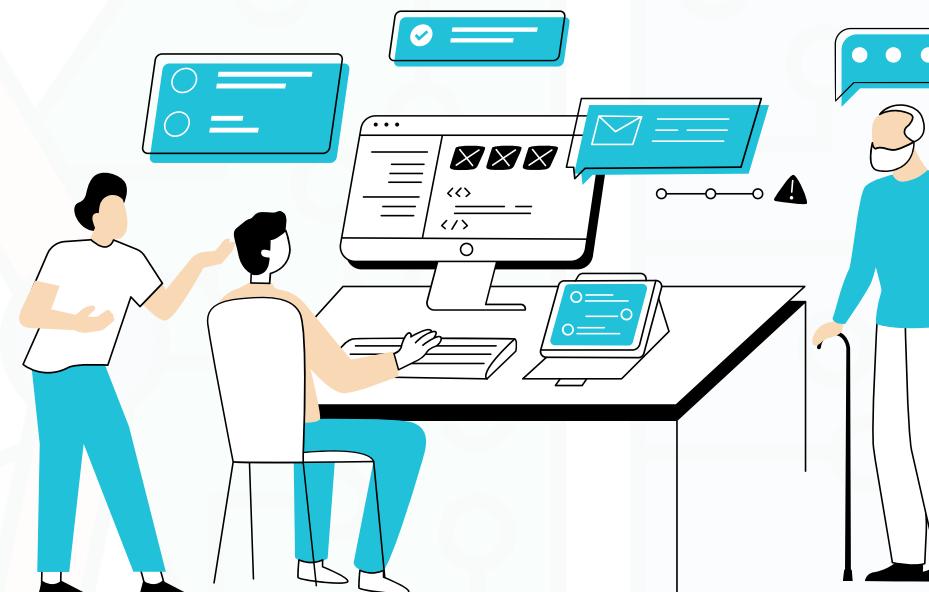
Sensors



Motors

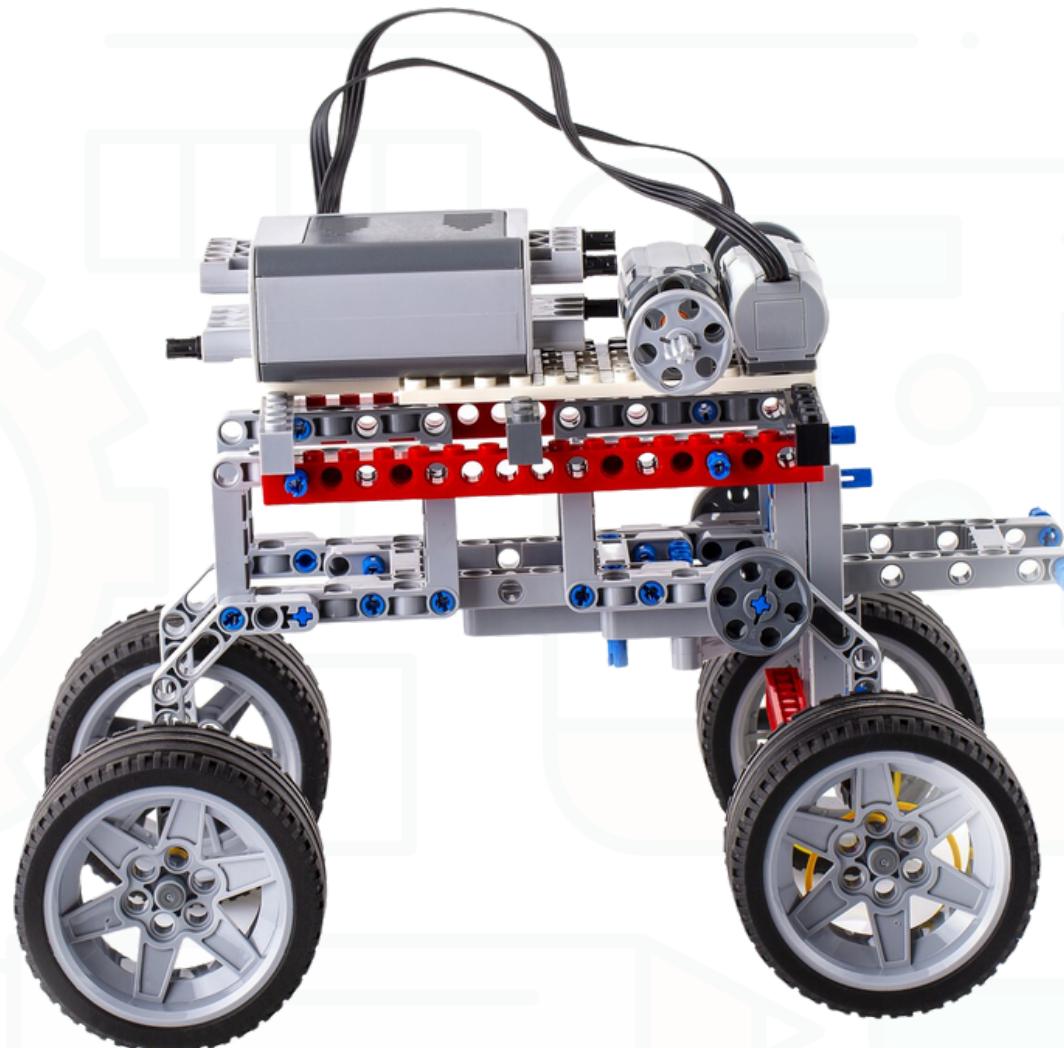


wires

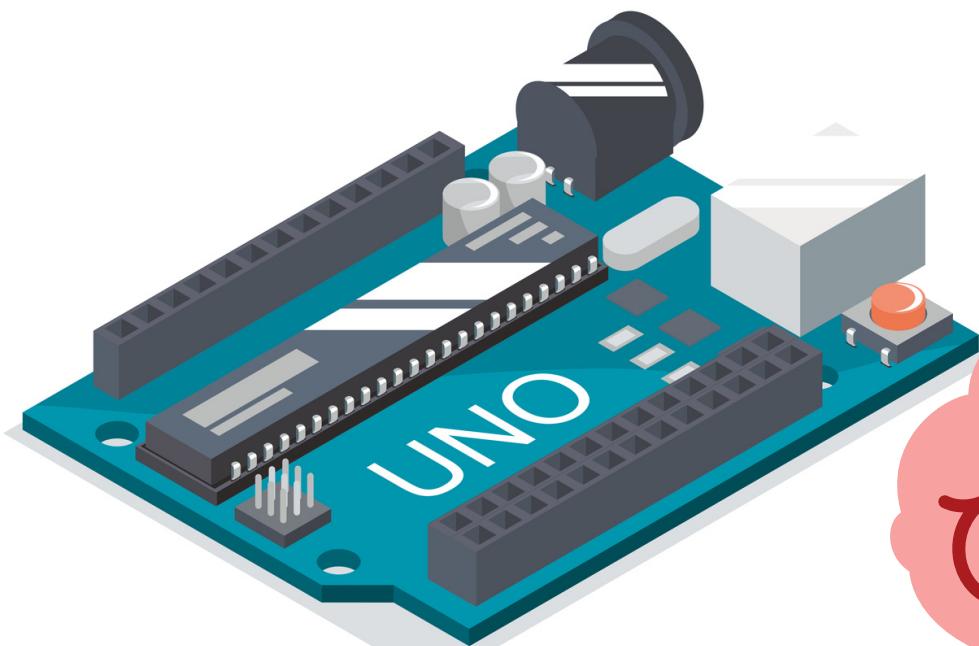
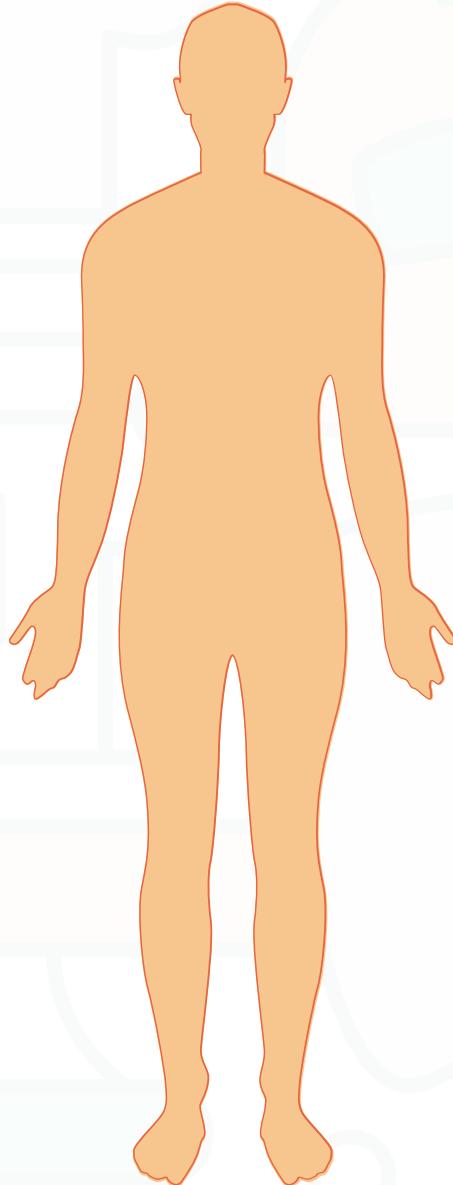


Software

Robot Parts vs Human Body parts



chassis



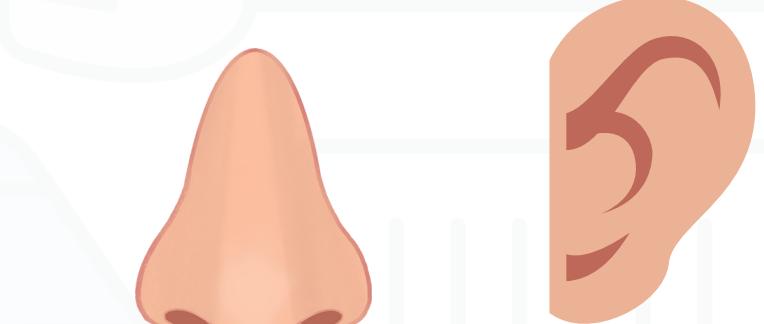
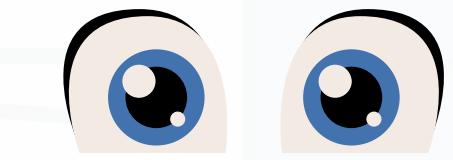
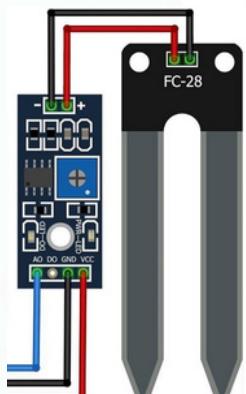
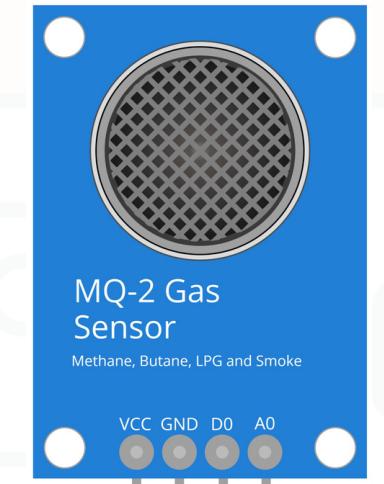
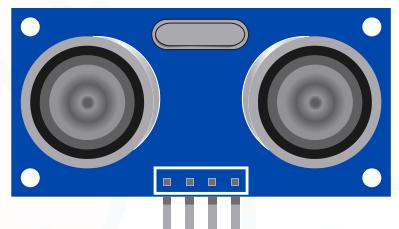
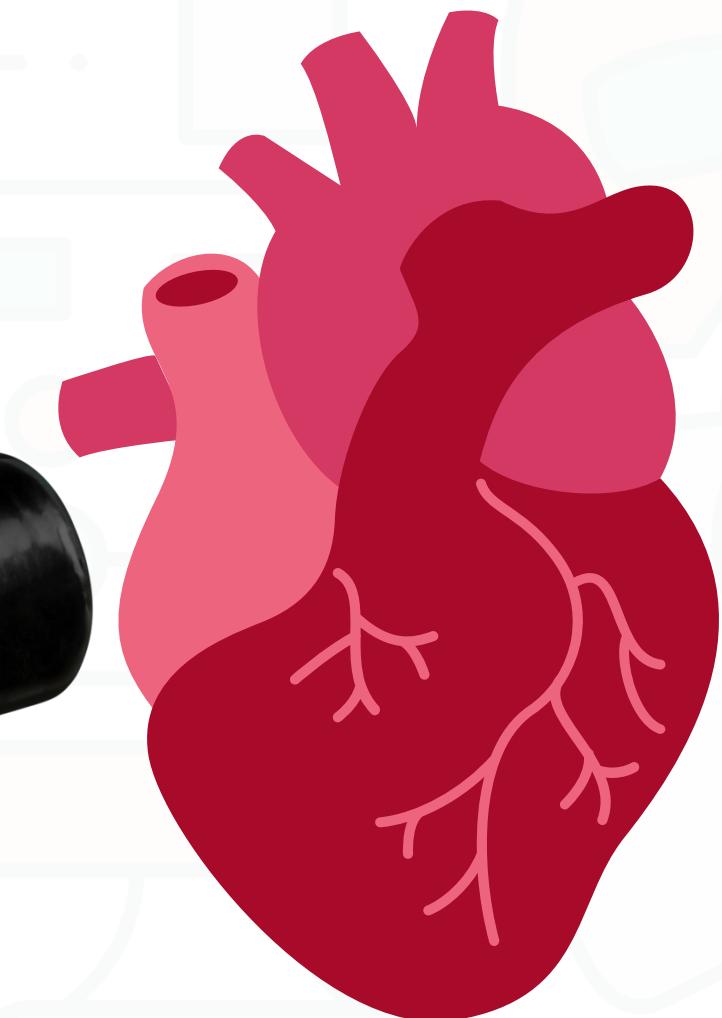
Microcontroller



Robot Parts vs Human Body parts

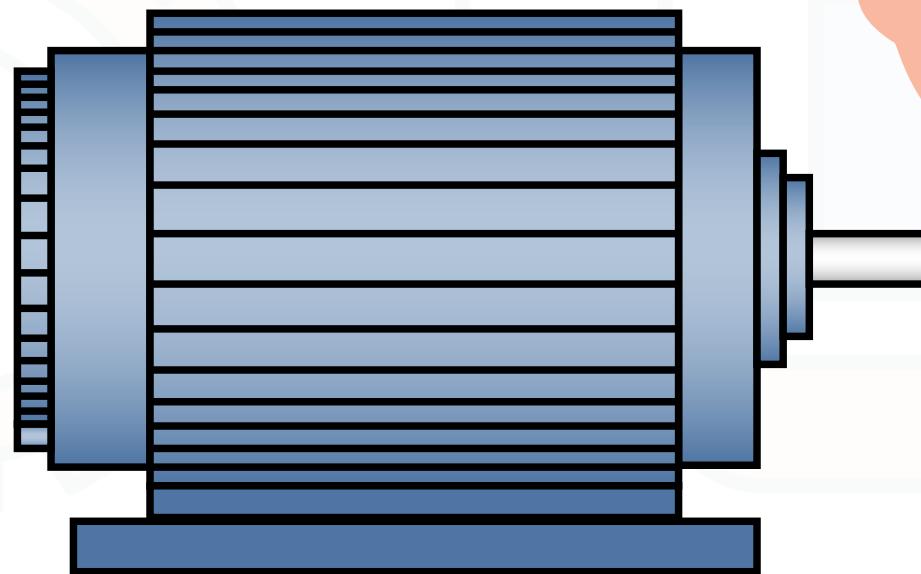


Batteries

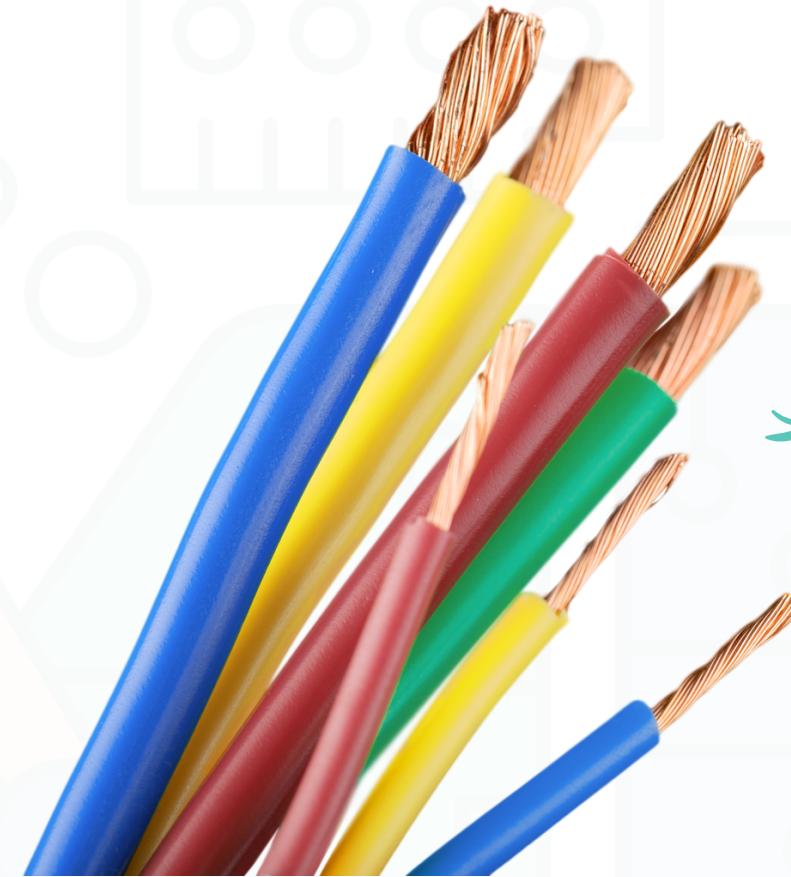


Sensors

Robot Parts vs Human Body parts



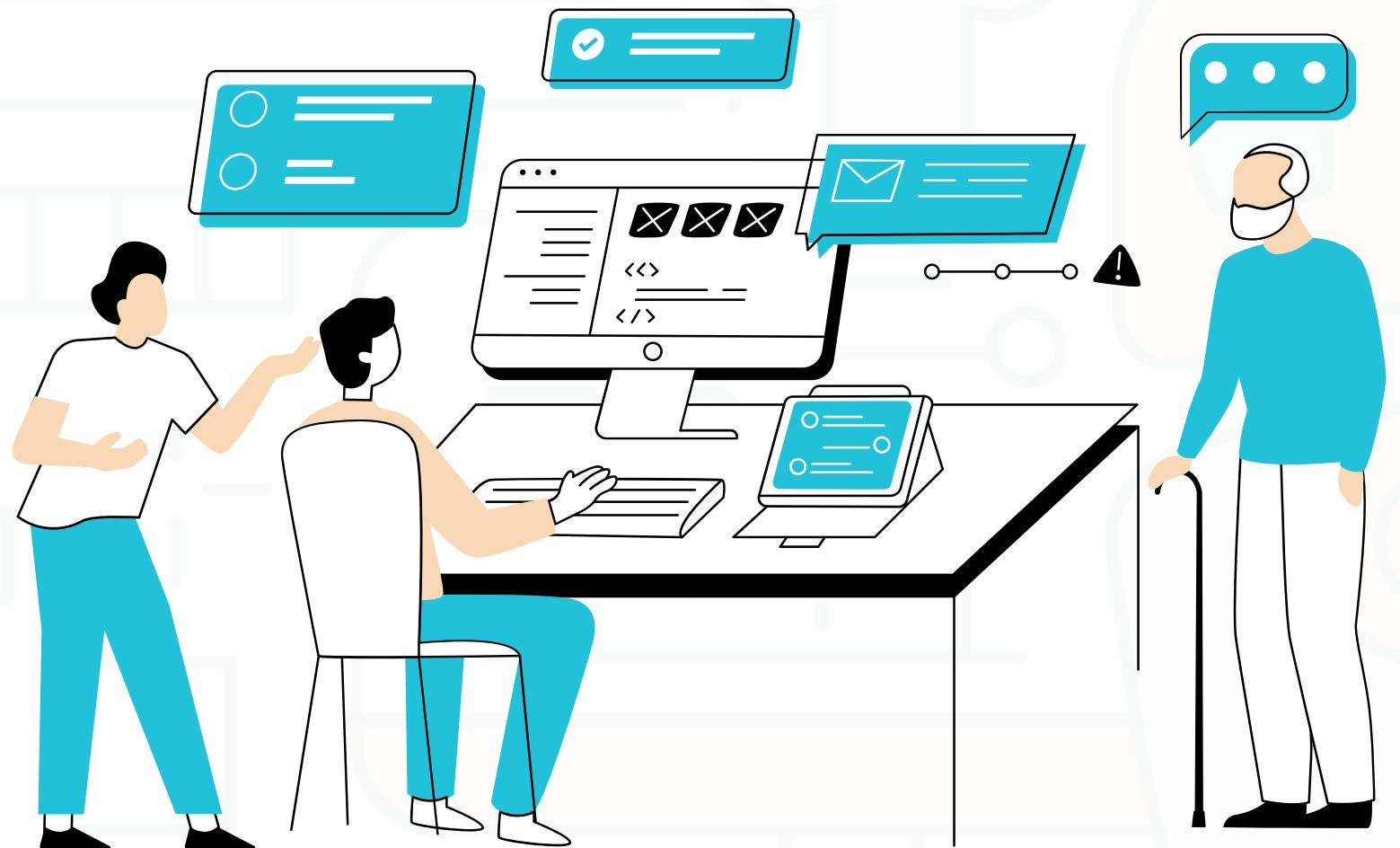
Motors



wires

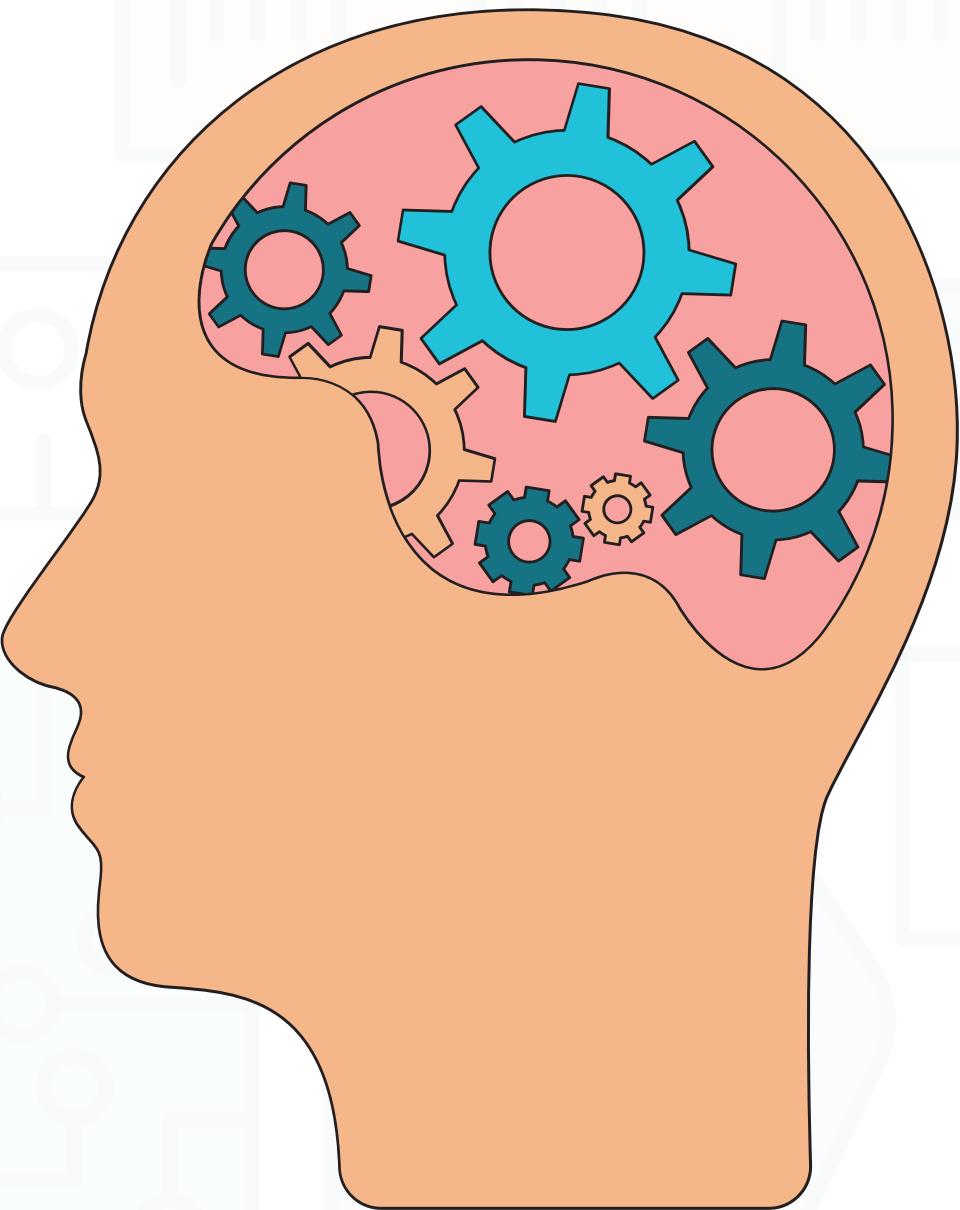


Robot Parts vs Human Body parts

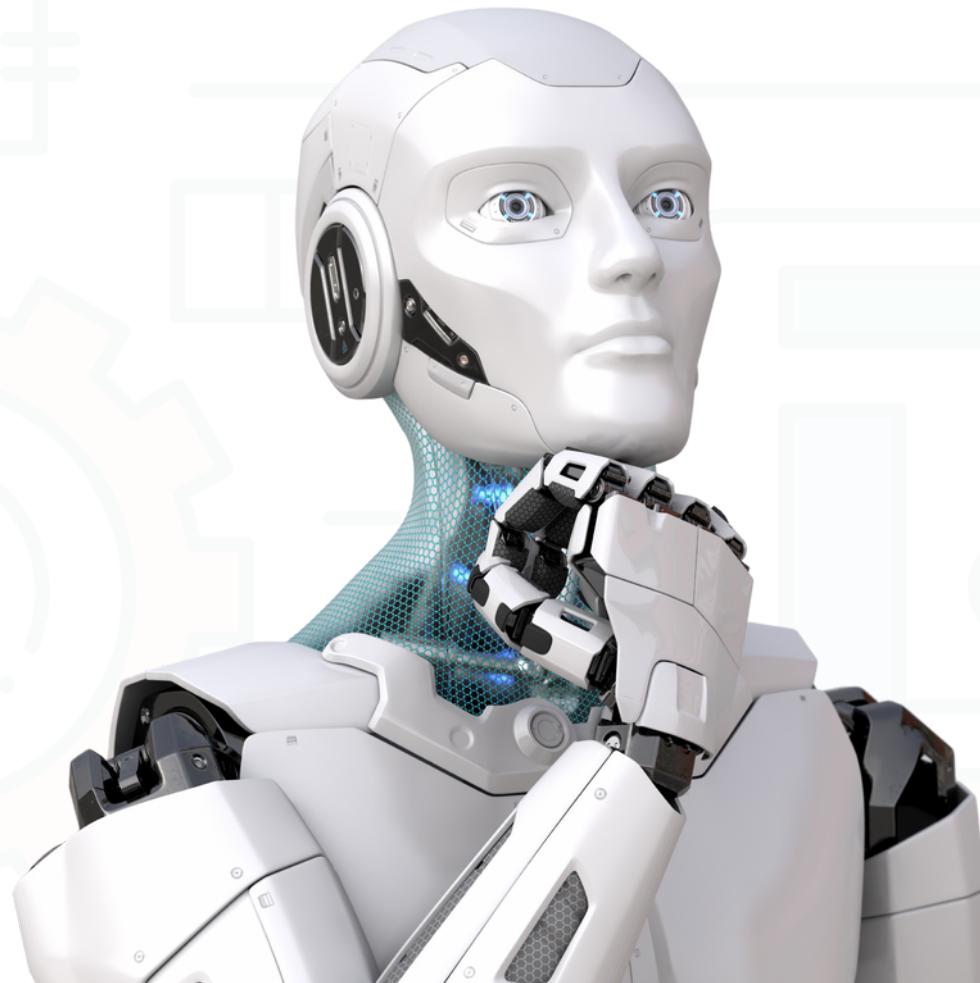


Software

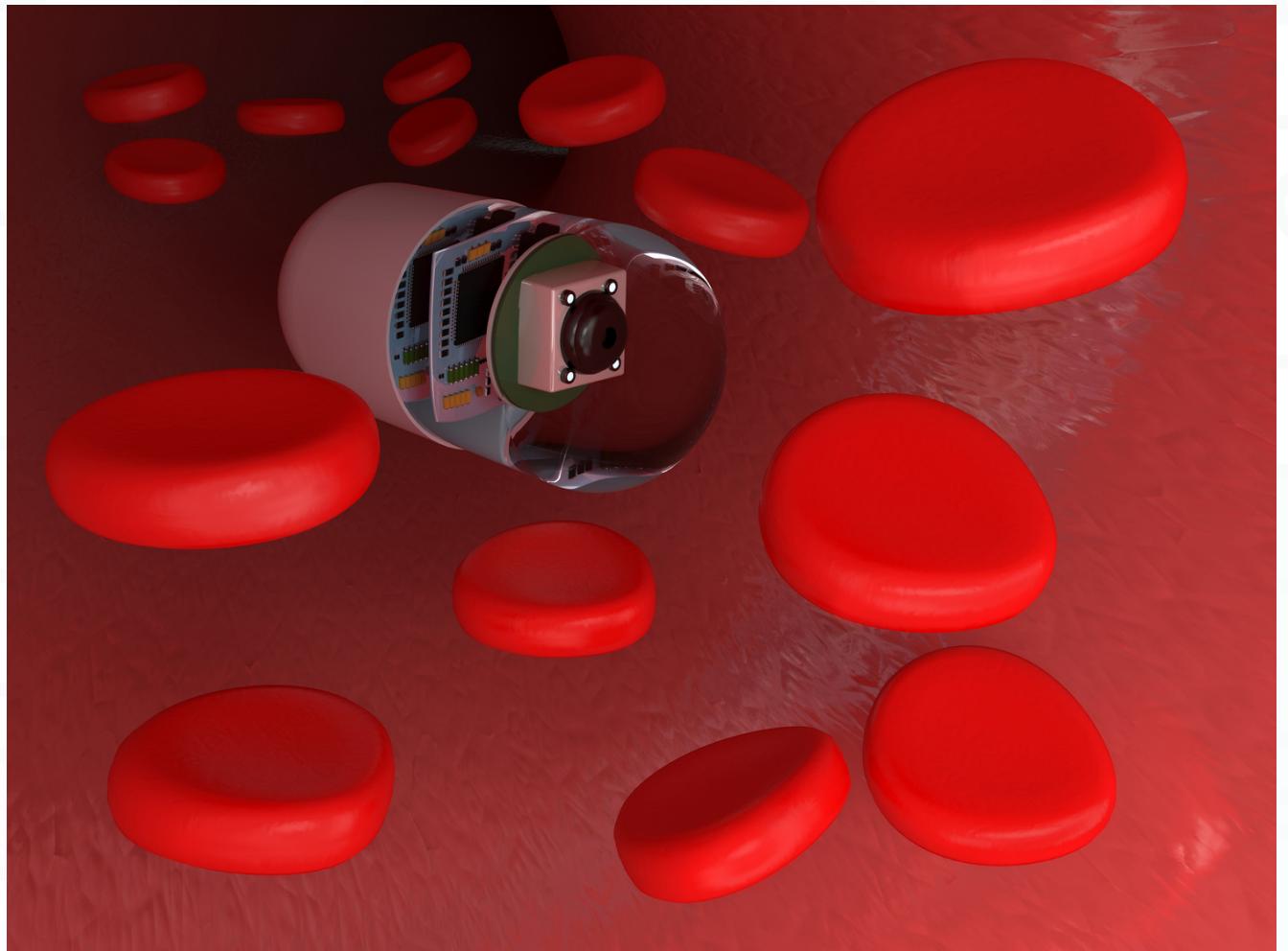
Cognition



Types of Robots



Humanoid

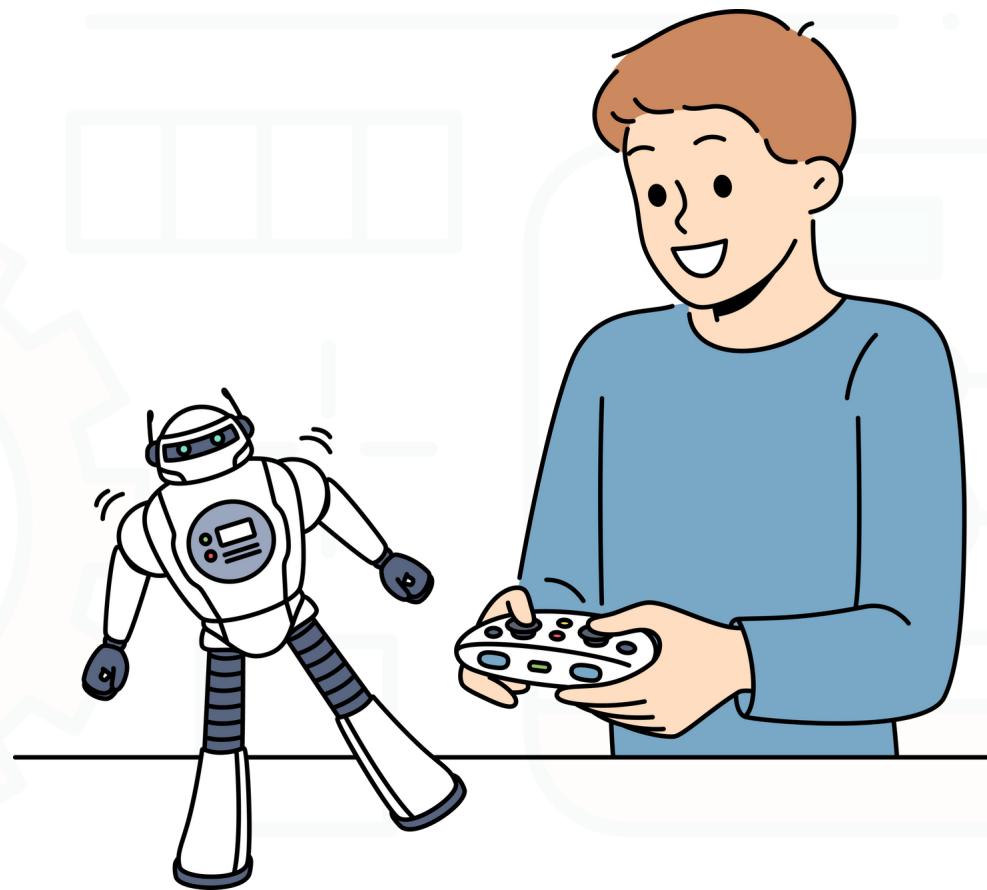


Nano

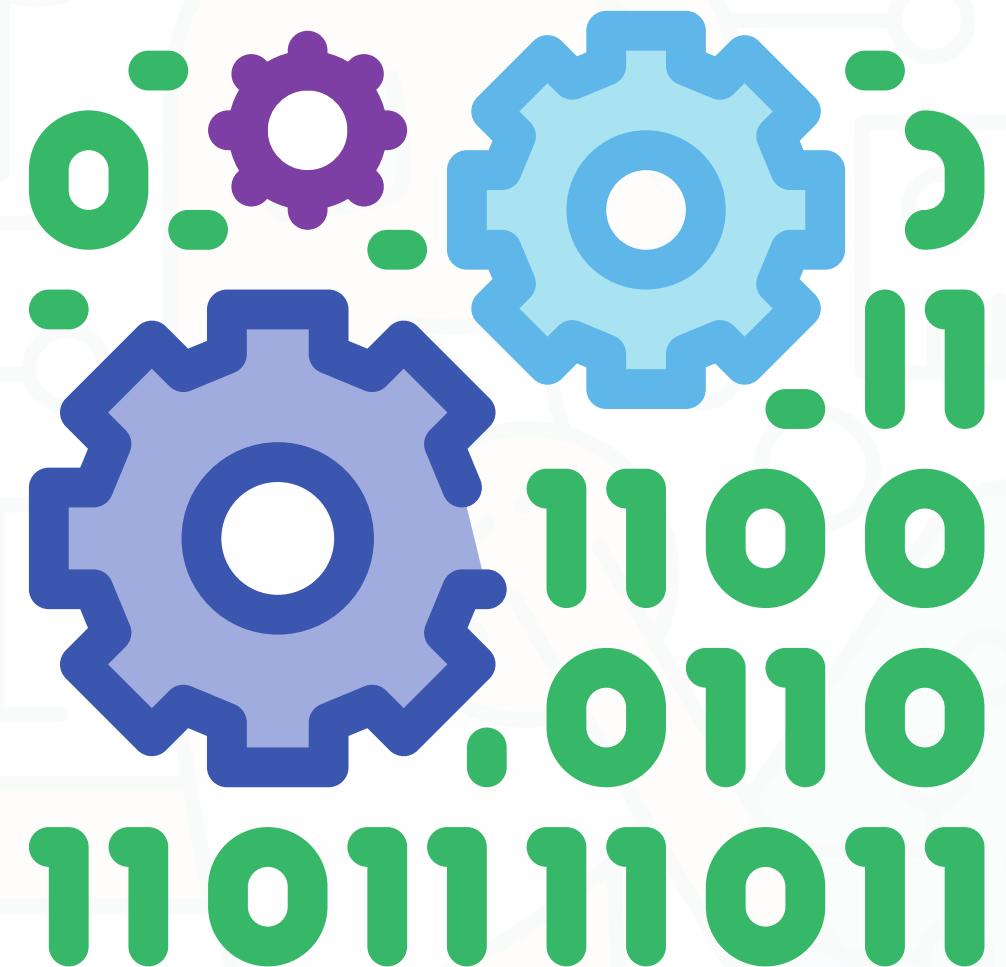


Enormous

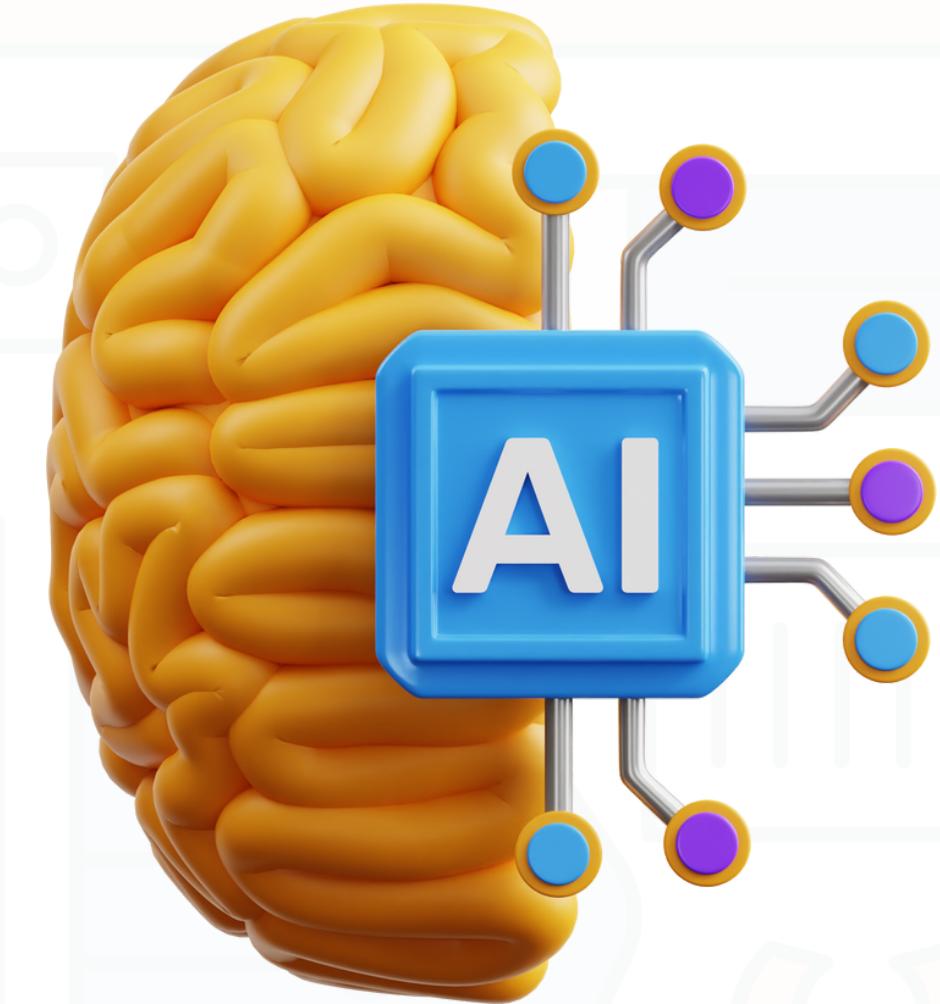
Robotics classification



Operated Robots

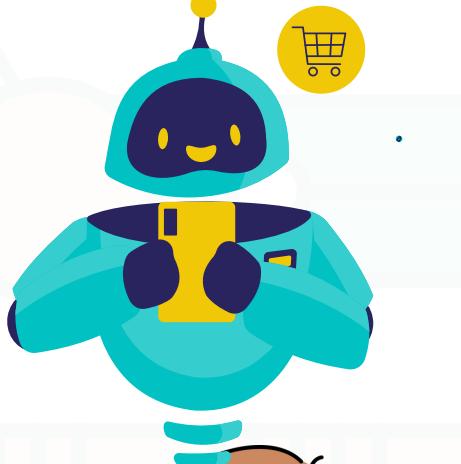


**Pre-programmed
Robots**



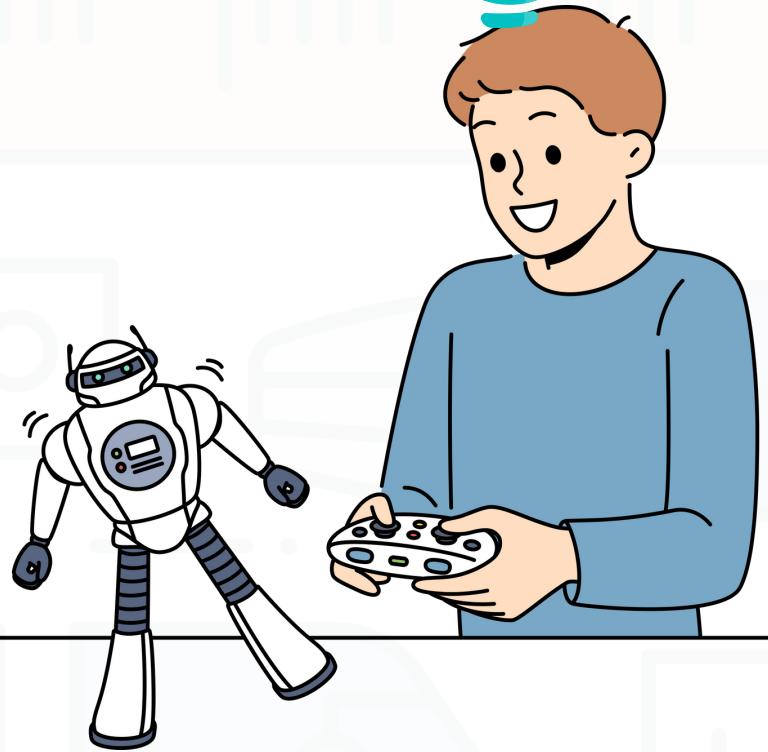
Autonomous

Robotics classification



Operated Robots

- 1. Operate under human control.**
- 2. Work in extreme geographical conditions.**
- 3. Doing dangerous and complex tasks.**
- 4. Use a wired or wireless connection to control from a safe distance.**

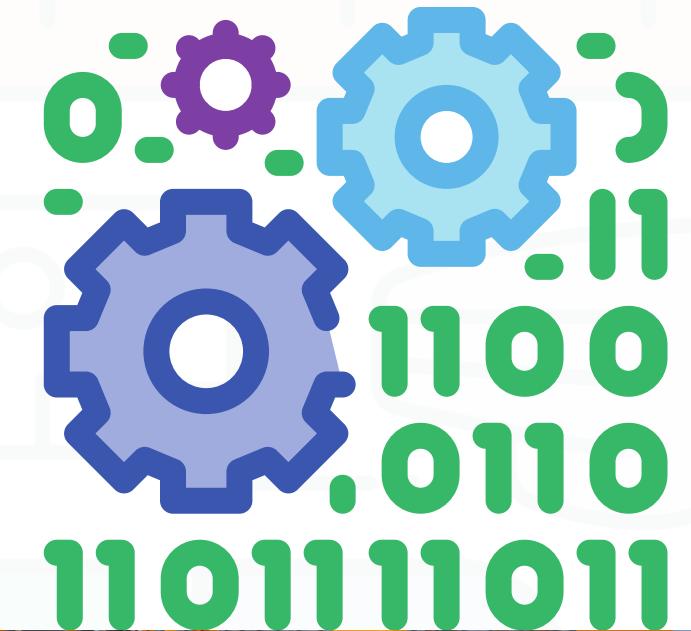


Robotics classification



Pre-programmed Robots

1. Perform a specific task in a specific way.
2. Work in a controlled environment.
3. Doing simple and monotonous tasks.
4. Doesn't require human intervention or assistance during their work.

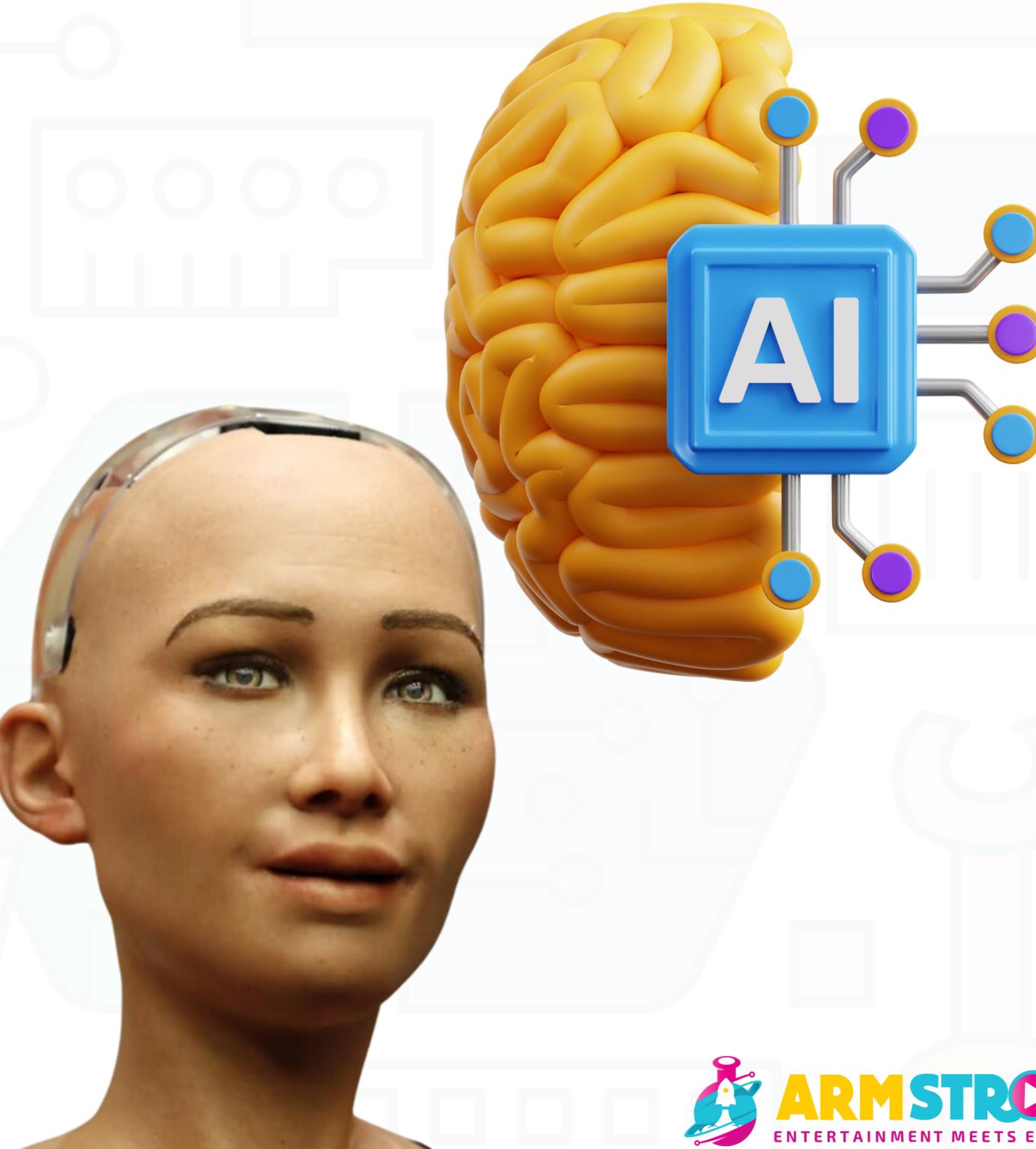


Robotics classification

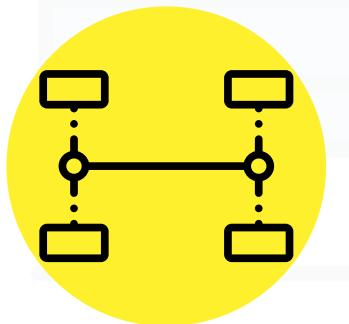


Autonomous Robots

1. Operate itself as they see fit using AI.
2. Work in an open environment.
3. Do tasks that require human expertise.
4. Work without human intervention or even any pre-orders.



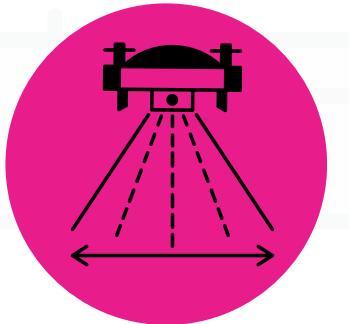
LAB Kit



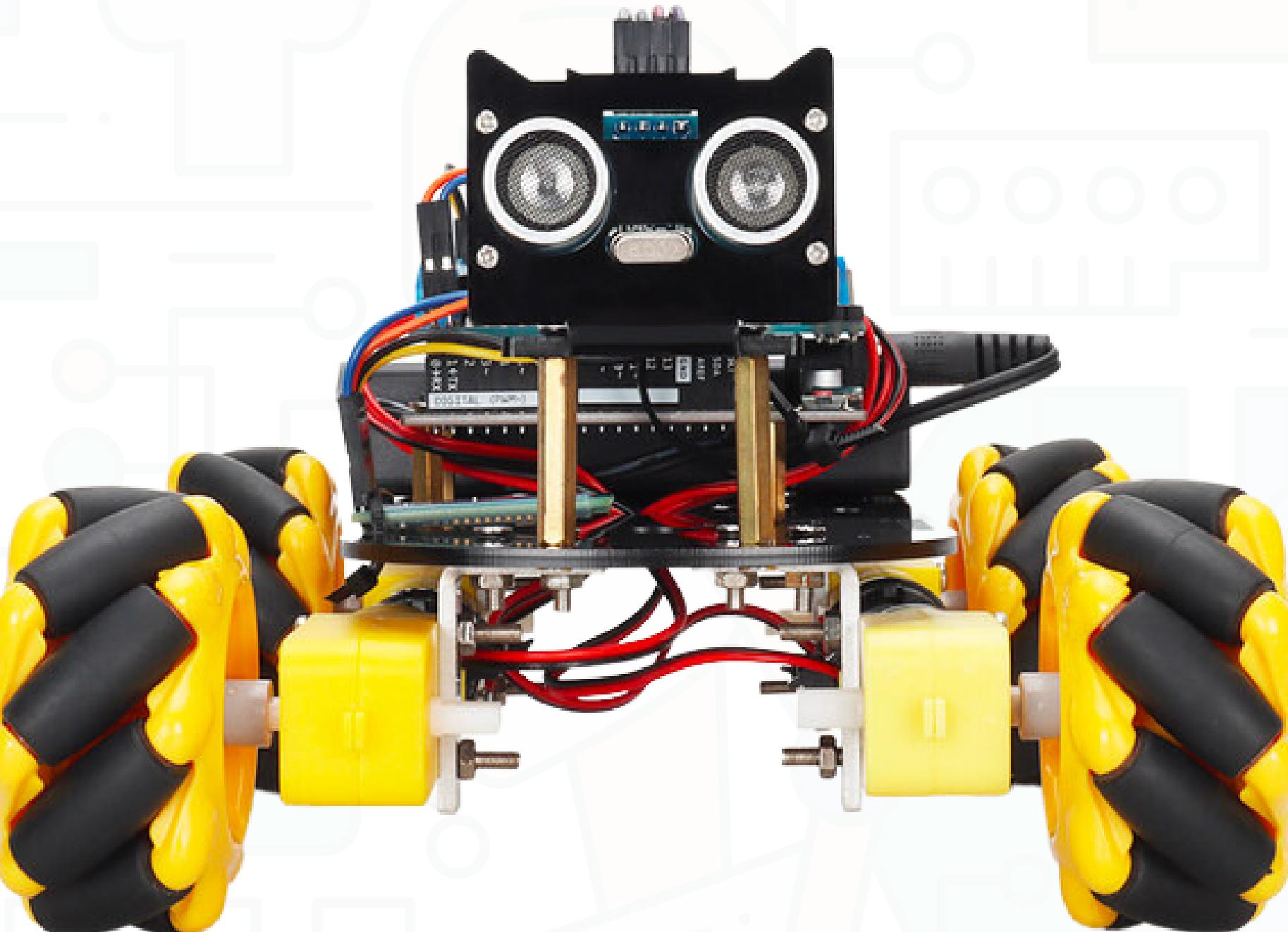
4WD



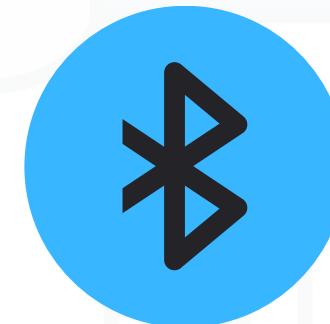
Omni-Wheels



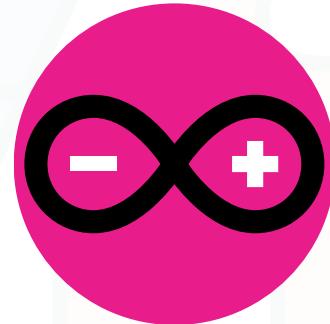
Sensors



Rechargeable



Wireless Control



Arduino
compatible

Intro to Arduino

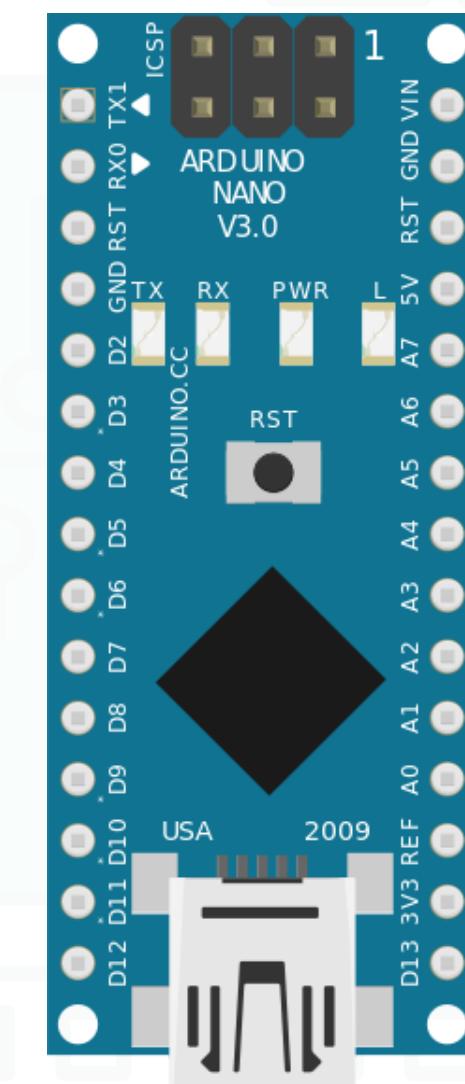
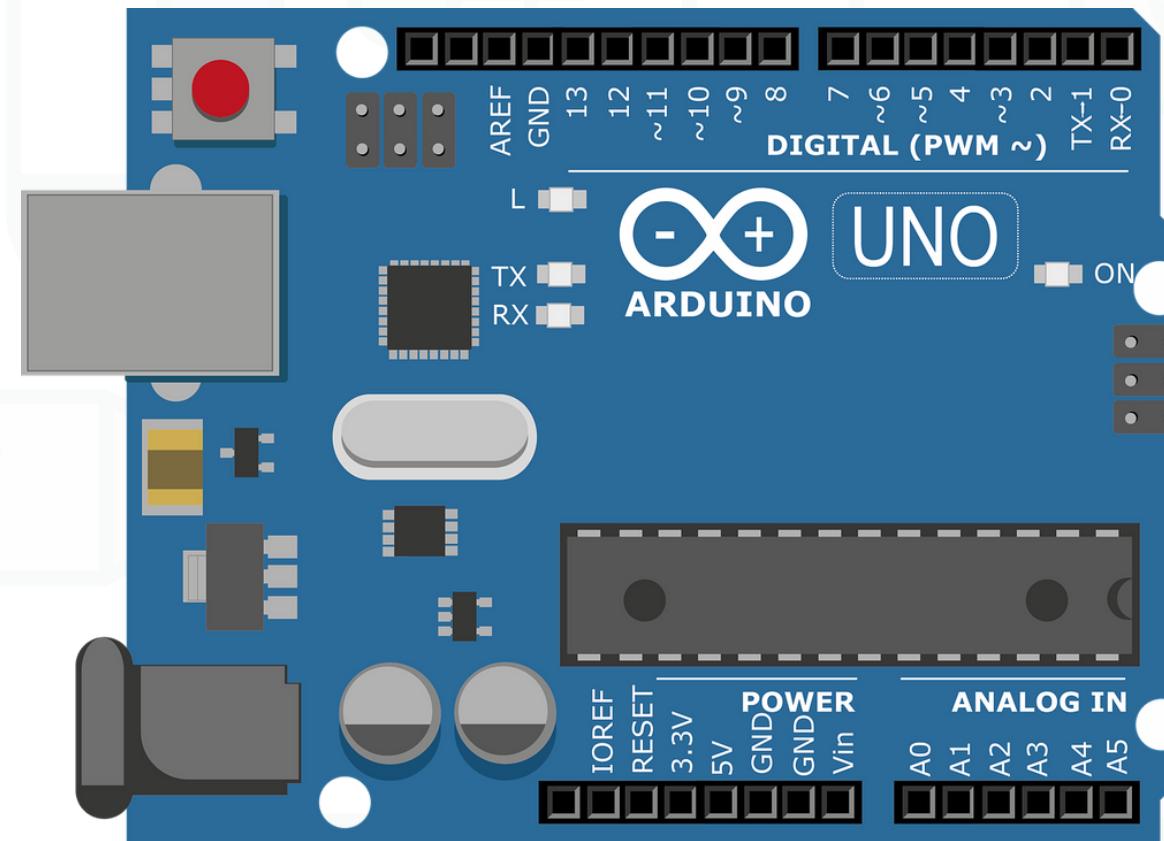


What is a microcontroller?

A microcontroller is a computer-on-a-chip optimized to control electronic devices.

An **Arduino board** is a microcontroller that has a user-friendly Ide.

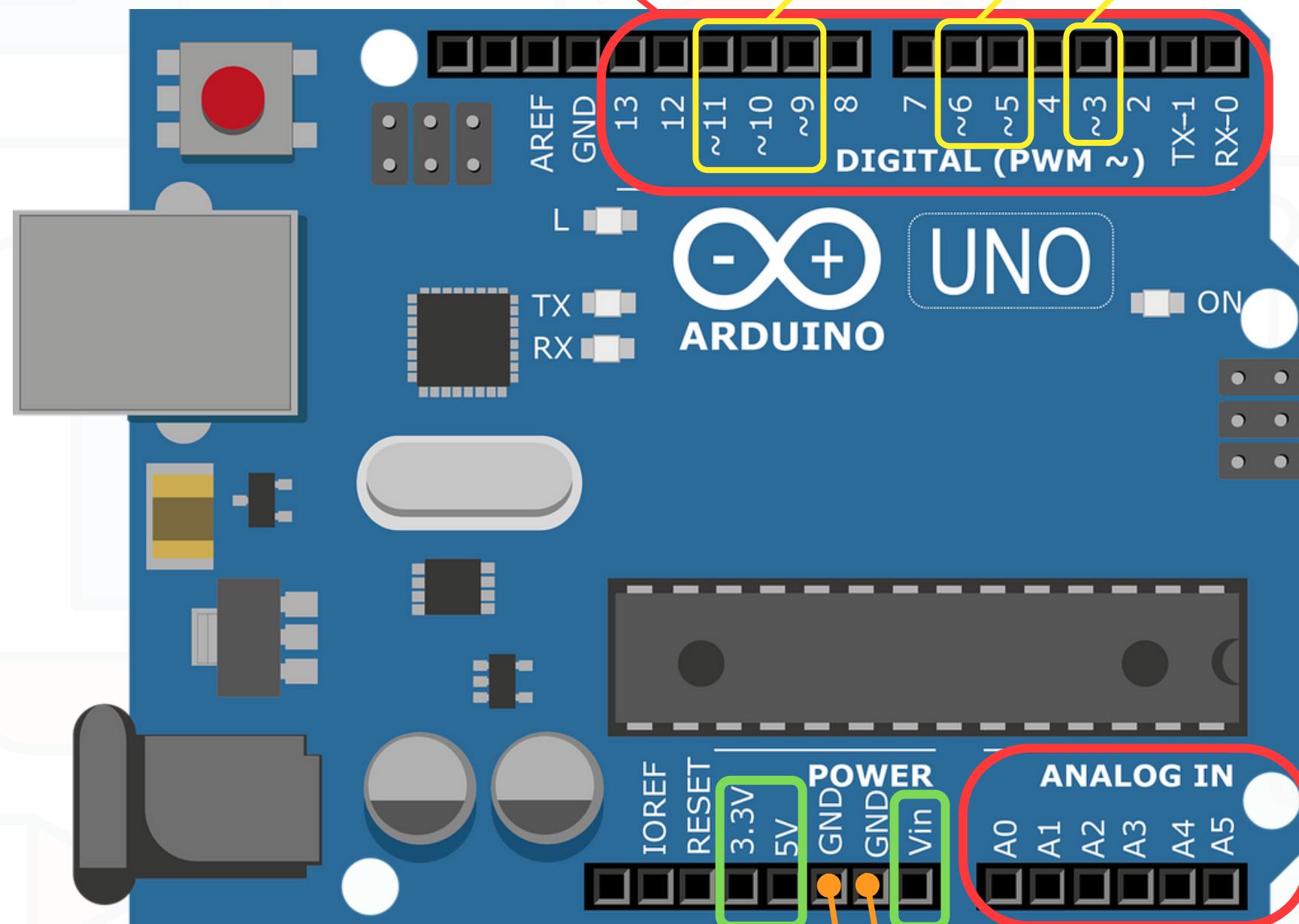
Arduino uno vs Arduino nano



Intro to Arduino



Digital input/output pins ← → Analog output pins



Output voltage pin

Input voltage pin

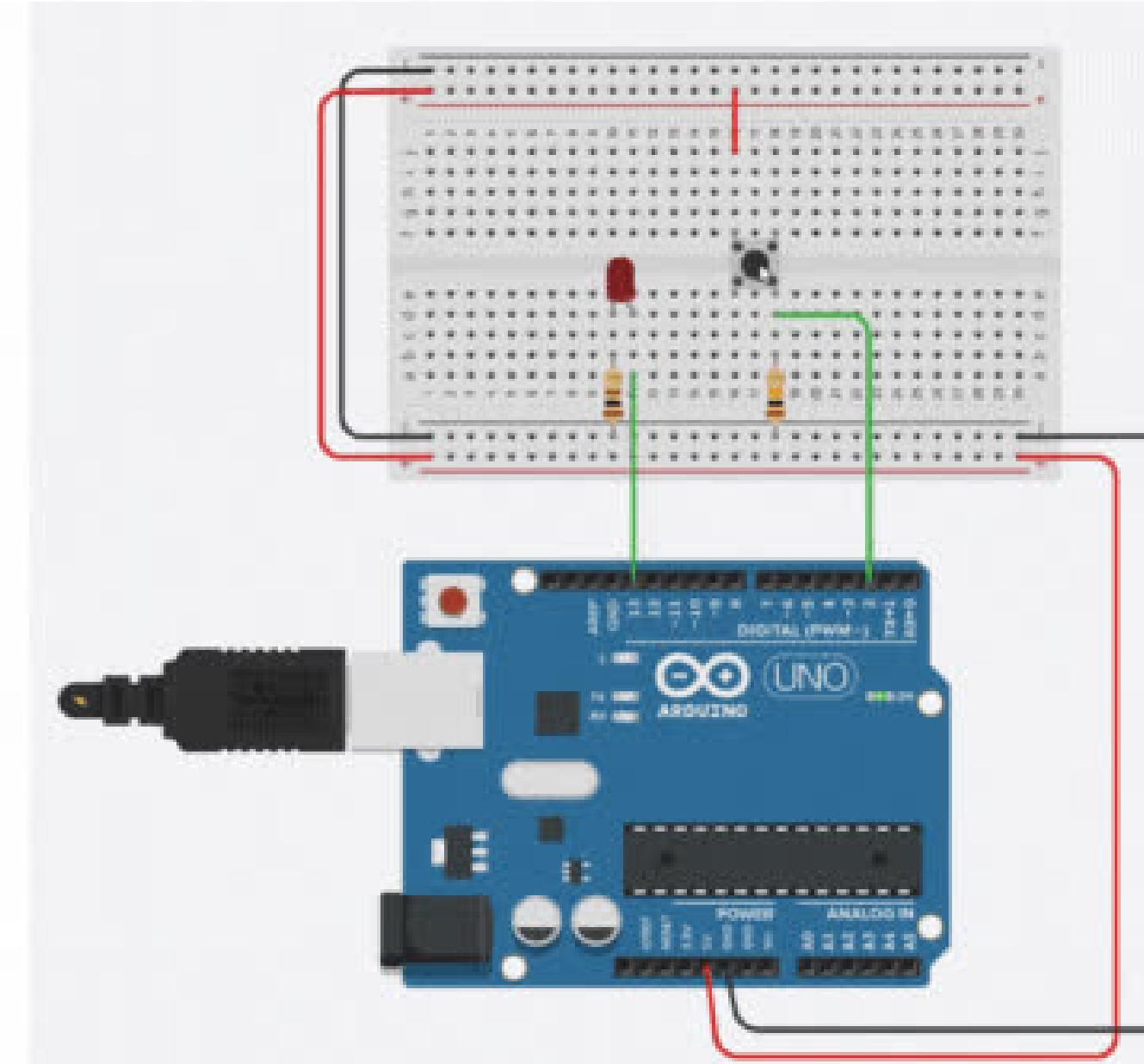
Ground

HIGH & LOW



HIGH= on, LOW=off

In programming a pin is **HIGH** when it outputs electricity (led on) and **LOW** when it doesn't output electricity.



Getting started with IDE



Step 1: Download Arduino IDE.

Downloads



Arduino IDE 2.1.1

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

[SOURCE CODE](#)

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

DOWNLOAD OPTIONS

Windows Win 10 and newer, 64 bits
Windows MSI installer
Windows ZIP file

Linux AppImage 64 bits (X86-64)
Linux ZIP file 64 bits (X86-64)

macOS Intel, 10.14: "Mojave" or newer, 64 bits
macOS Apple Silicon, 11: "Big Sur" or newer, 64 bits

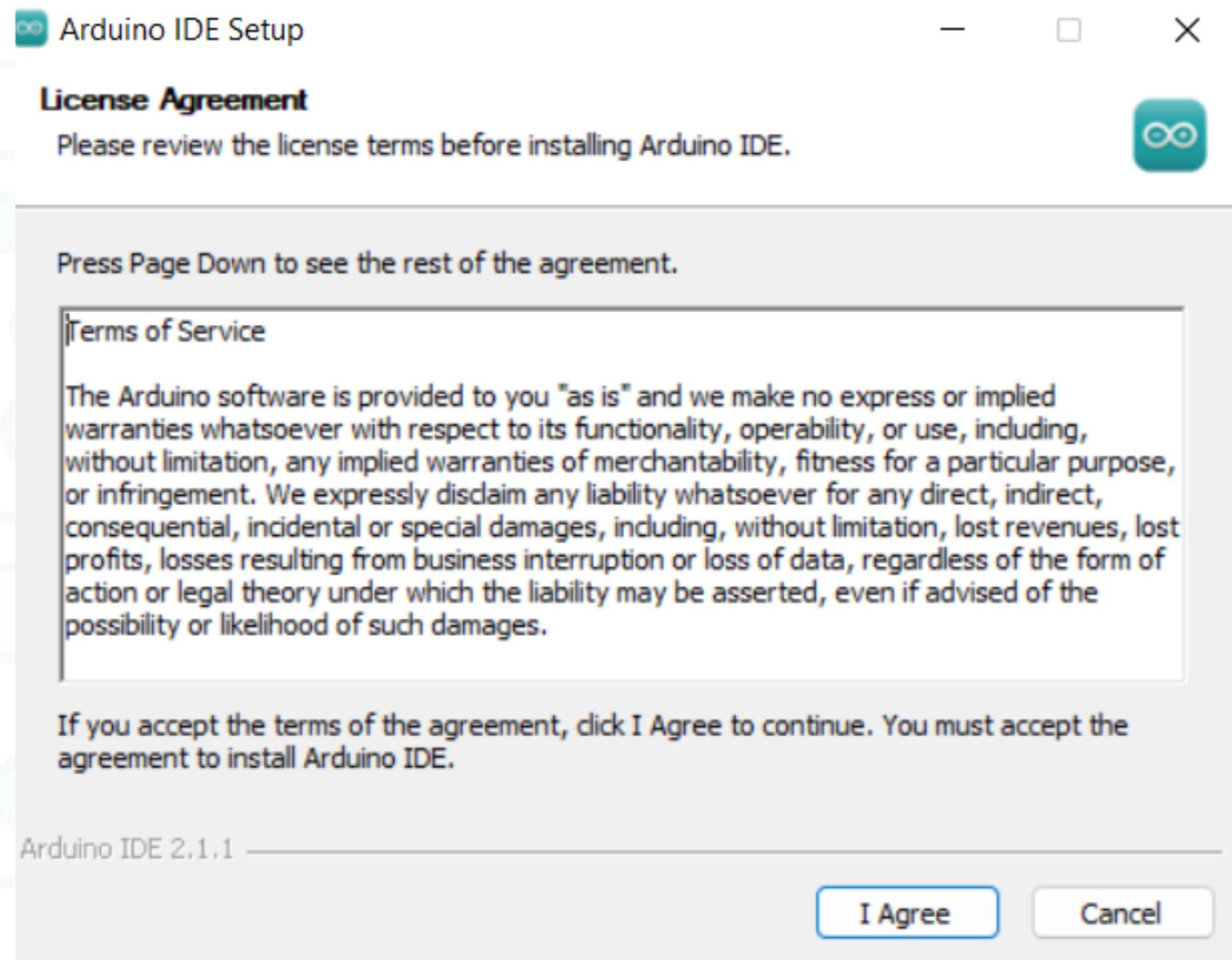
[Release Notes](#)

Choose Your operating system

Getting started with IDE



Step 2: Install Arduino IDE.



Getting started with IDE



Step 3: Set up.

The image shows the Arduino IDE interface. On the left, the code editor displays a sketch named "sketch_aug15a.ino". The code contains two functions: "setup()" and "loop()". The "setup()" function is intended for one-time setup code, and the "loop()" function is for repeated main code. Lines 1 through 10 are highlighted with a red oval, and a red arrow points from the text "Lines of code" to this area. Another red arrow points from the text "Here you can write your code" to the code editor area. On the right, a yellow-bordered callout box shows the "File" menu with options like "New Sketch", "Open...", and "Save".

File Edit Sketch Tools Help

Select Board

sketch_aug15a.ino

```
1 void setup() {
2     // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7     // put your main code here, to run repeatedly:
8
9 }
10
```

Lines of code

Here you can write your code

File Edit Sketch Tools Help

- New Sketch Ctrl+N
- New Cloud Sketch Alt+Ctrl+N
- Open... Ctrl+O
- Open Recent
- Sketchbook
- Examples
- Close Ctrl+W
- Save Ctrl+S
- Save As... Ctrl+Shift+S
- Preferences... Ctrl+Comma
- Advanced
- Quit Ctrl+Q

Getting started with IDE



Step 4: Let's code.

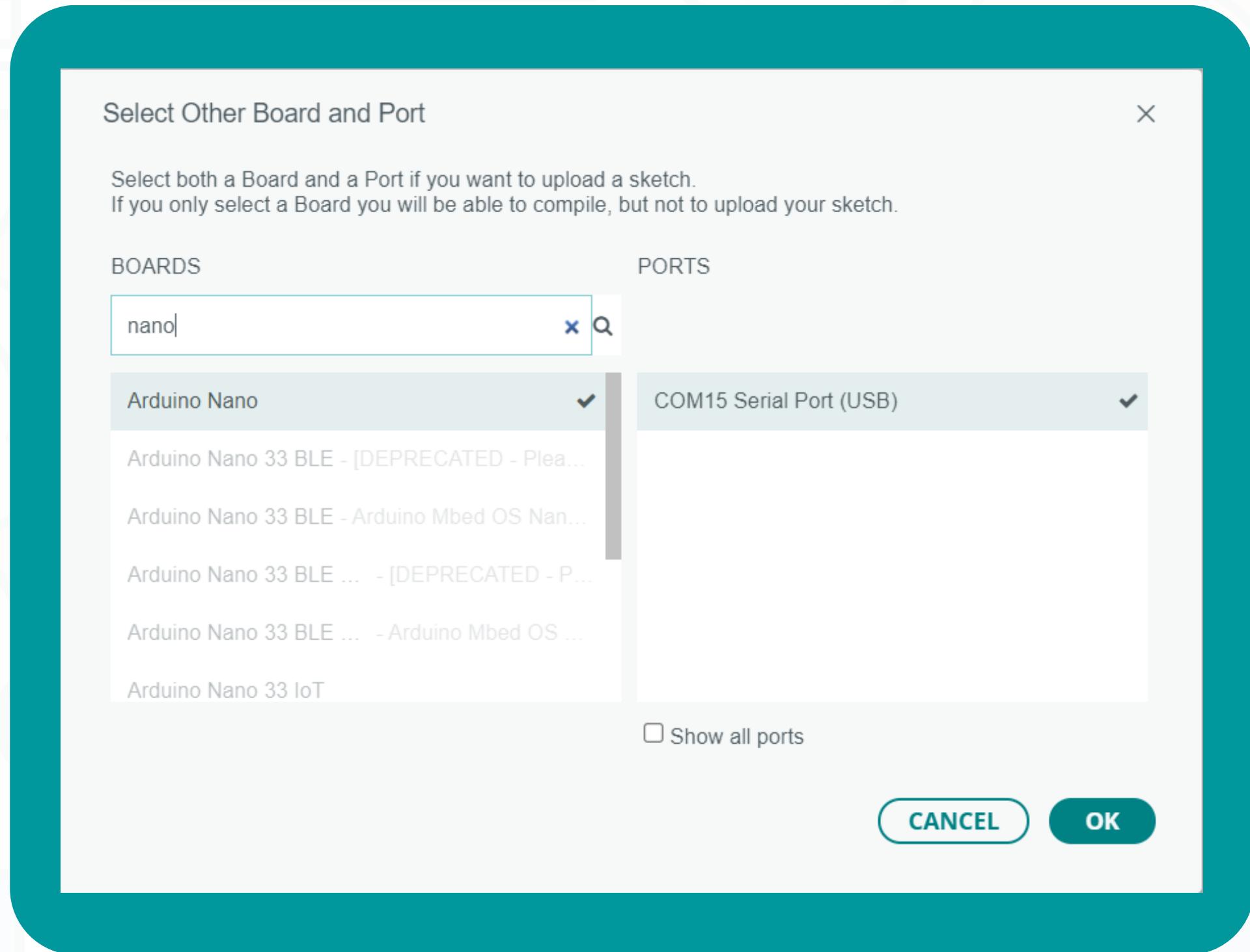
```
1 void setup() {  
2     // put your setup code here, to run once:  
3  
4 }
```

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

Getting started with IDE



Step 5: Upload code.



1

Select Your Arduino and the port that it's connected to

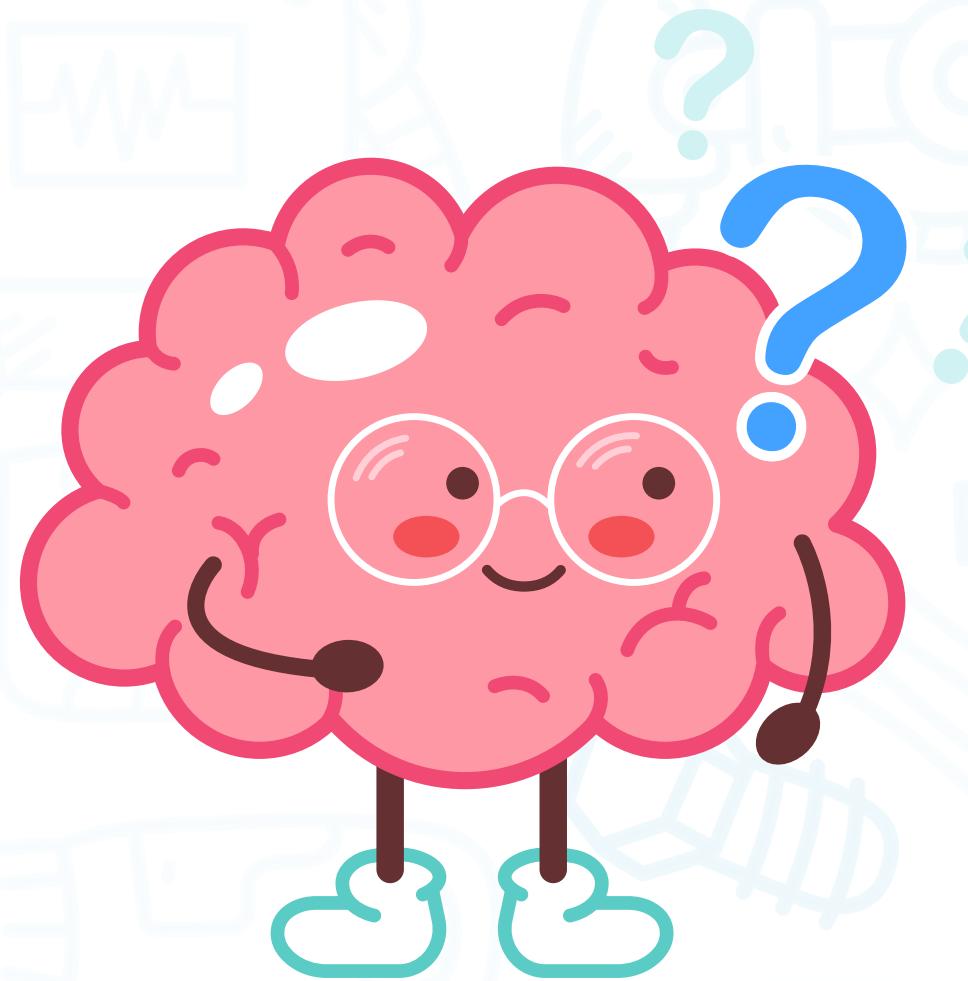
2



Verify Upload

Think

How can we write to make the built in led in the arduino blink?

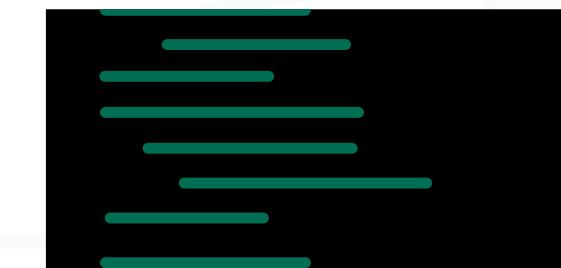


Code

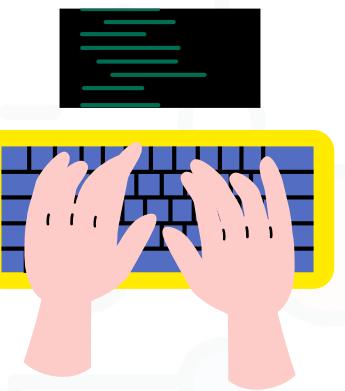
Write a code to make the built in led in the arduino blink.

Try it by yourself

Hint: Built-in led connected to pin 13

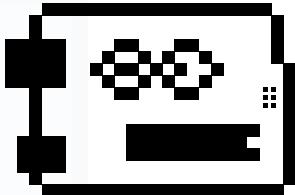


Code



Step 1: Set the built in LED "HIGH" for 1sec

```
void setup() {  
    // initialize digital pin LED_BUILTIN as an output.  
    pinMode(LED_BUILTIN, OUTPUT);  
}
```



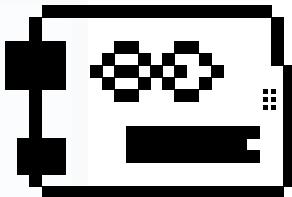
Code



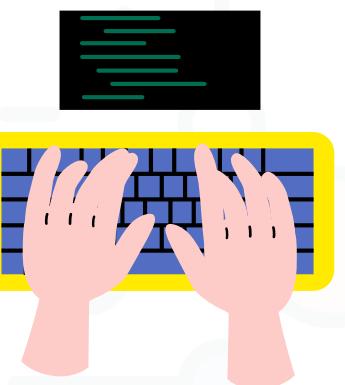
Step 1: Set the built in LED "HIGH" for 1sec

```
void setup() {  
    // initialize digital pin LED_BUILTIN as an output.  
    pinMode(LED_BUILTIN, OUTPUT);  
}
```

```
void loop() {  
    digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)  
    delay(1000); // wait for a second
```



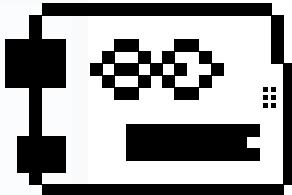
Code



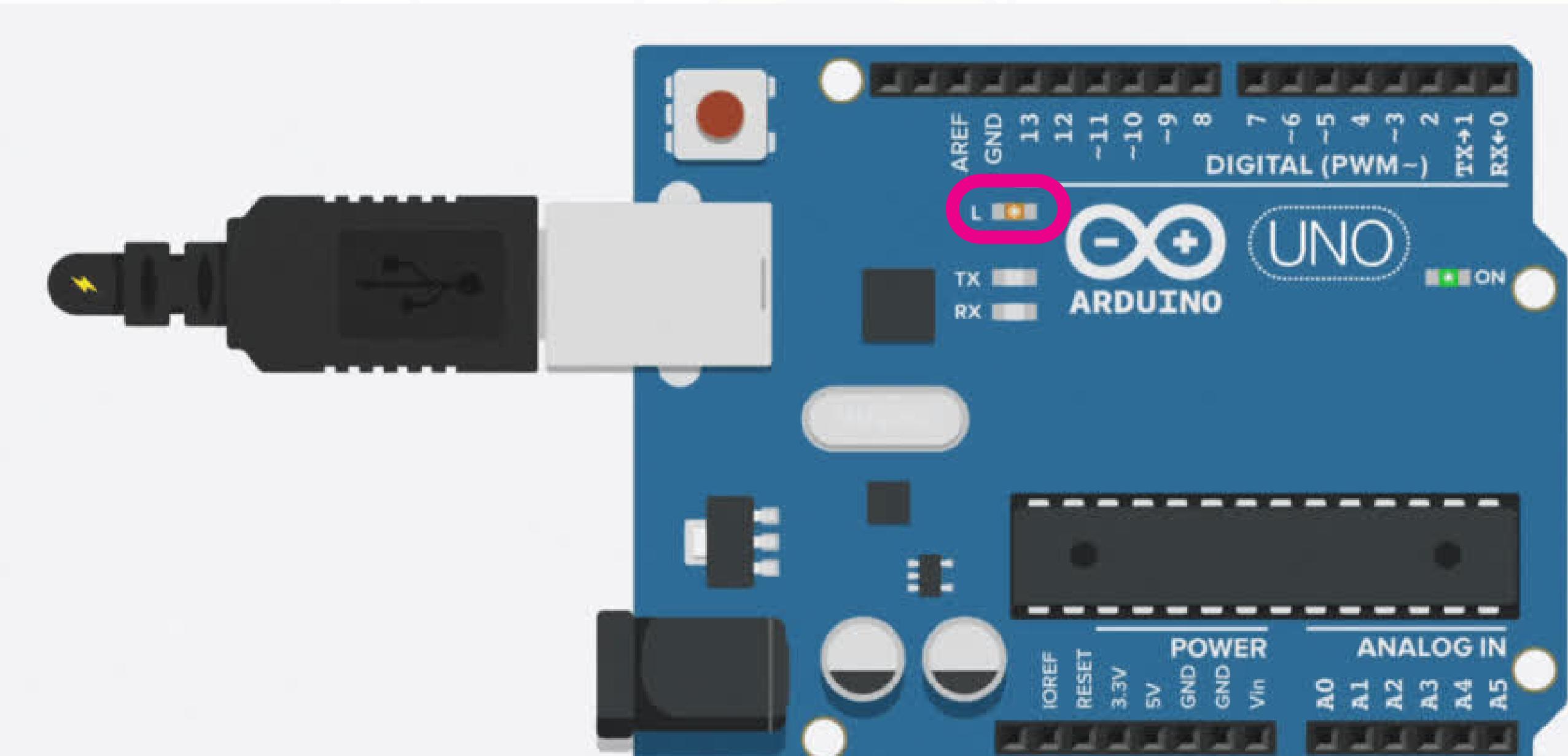
Step 2: Set the built in LED "LOW" for 1sec

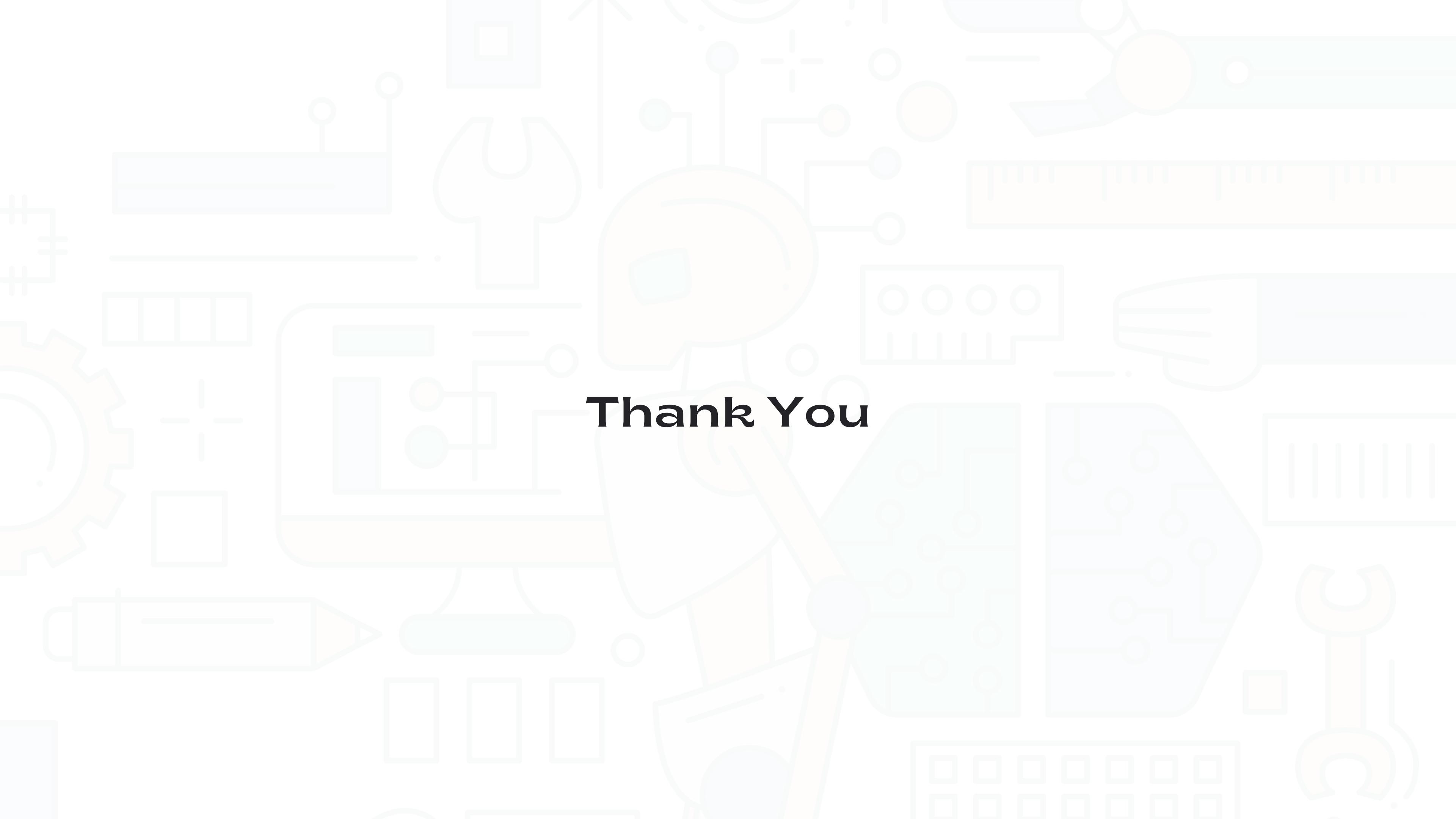
```
void setup() {  
    // initialize digital pin LED_BUILTIN as an output.  
    pinMode(LED_BUILTIN, OUTPUT);  
}
```

```
void loop() {  
    digitalWrite(LED_BUILTIN, HIGH);    // turn the LED on (HIGH is the voltage level)  
    delay(1000);                      // wait for a second  
    digitalWrite(LED_BUILTIN, LOW);     // turn the LED off by making the voltage LOW  
    delay(1000);                      // wait for a second  
}
```



Code





Thank You