



# Armstrong

## School Program 2023-2024

### Orientation Session



# Armstrong

## entertainment meets education

REGISTER NOW



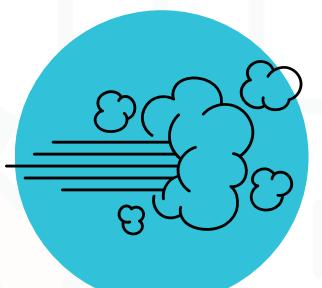
<https://armstrongedu.com/>



# Lesson Content



**Variables**



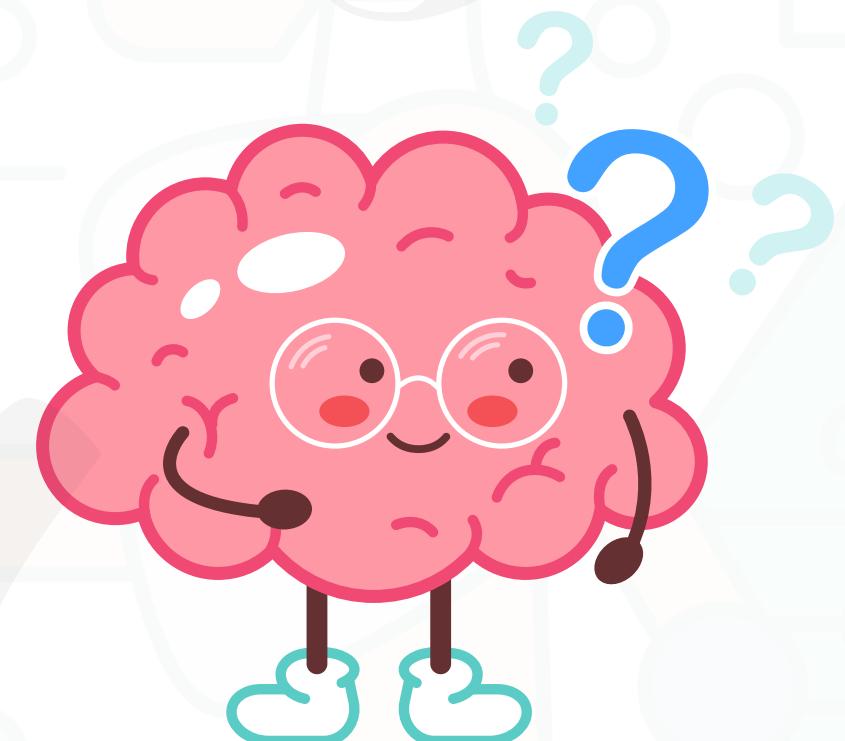
**Control speed**



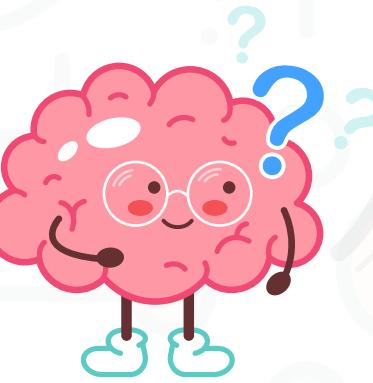
**Light dimmer**

# Think

## What is a constant?



# Think



## What is a constant?

A constant is a word or a symbol that represents a fixed value that does not change.

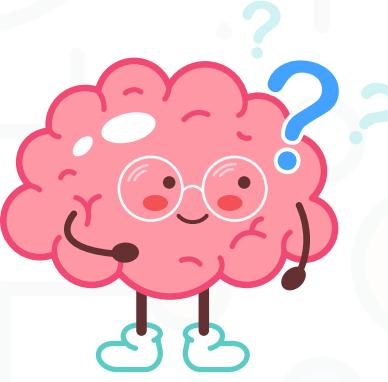
For example, the number 10 is a constant, because it always means the same thing.

# Think

What is a variable?



# Think



## What is a variable?

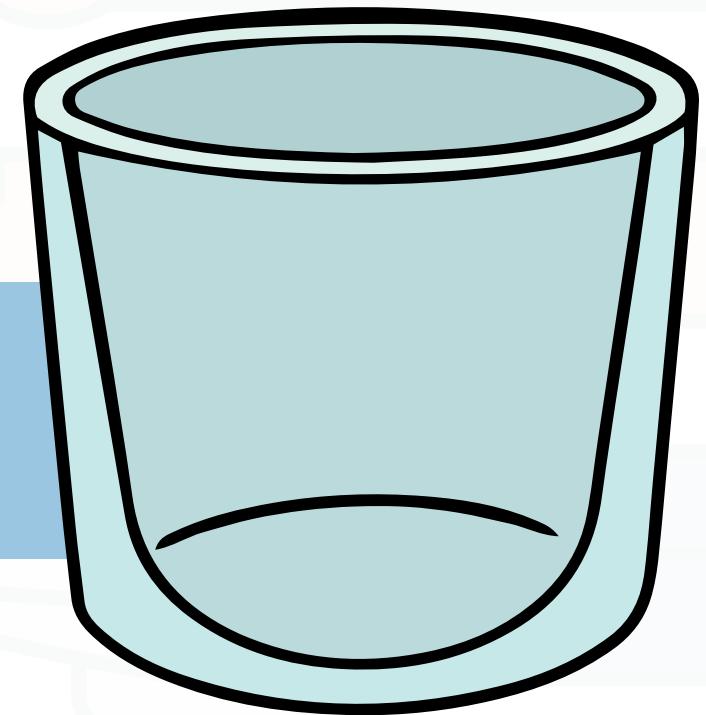


Constants are different from variables, which are words or symbols that can change their values depending on the situation.

For example, the word age is a variable, because it can mean different things for different people. The letter x is also a variable, because it can represent different numbers in different equations.

# Analogy ≡

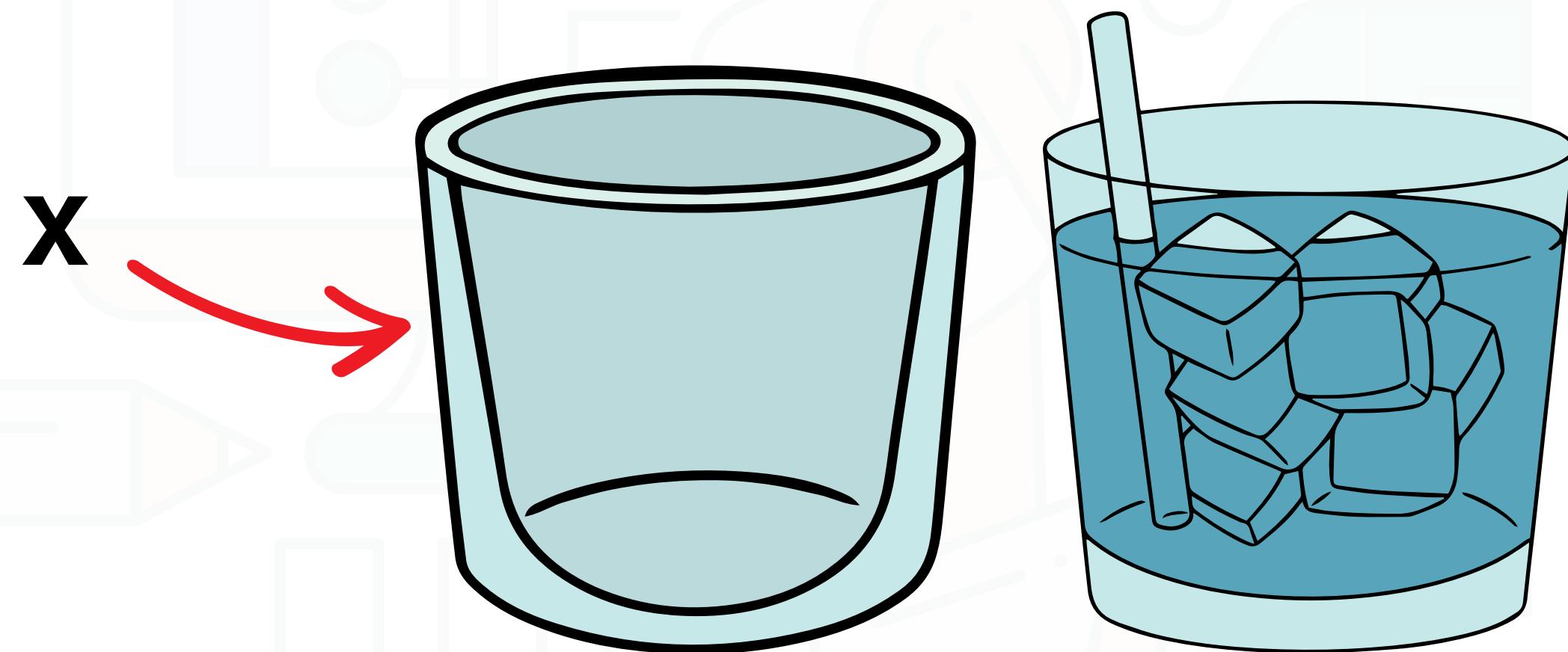
Imagine you have an empty cup.



Think about cups as containers that can hold different things, such as liquids, candies, or coins.

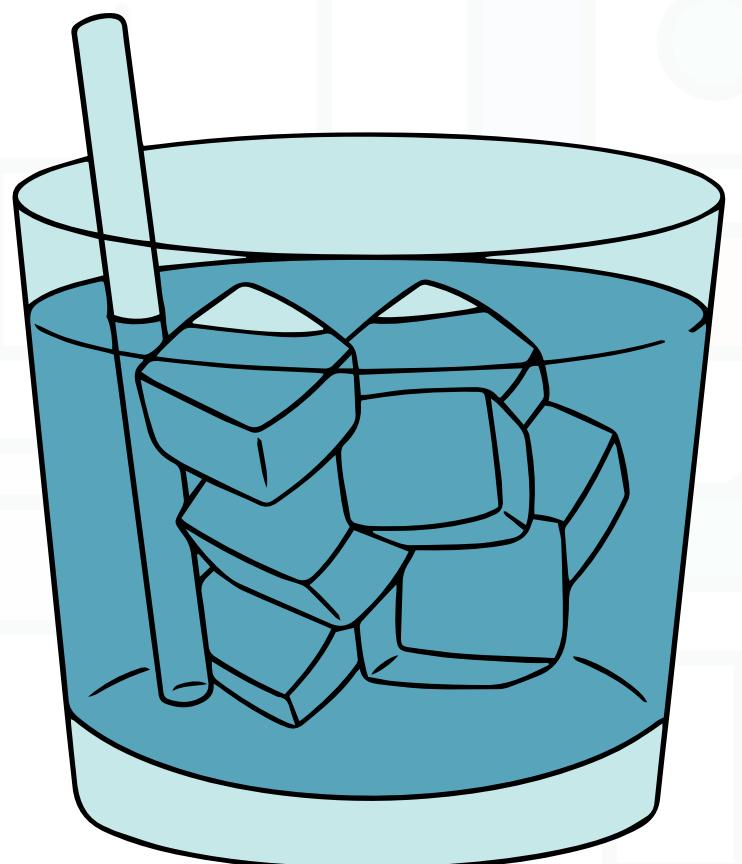
# Analogy

For example, you can have a cup named **x** and fill it with water. Then you can say that **x** is a variable that has the value of water.

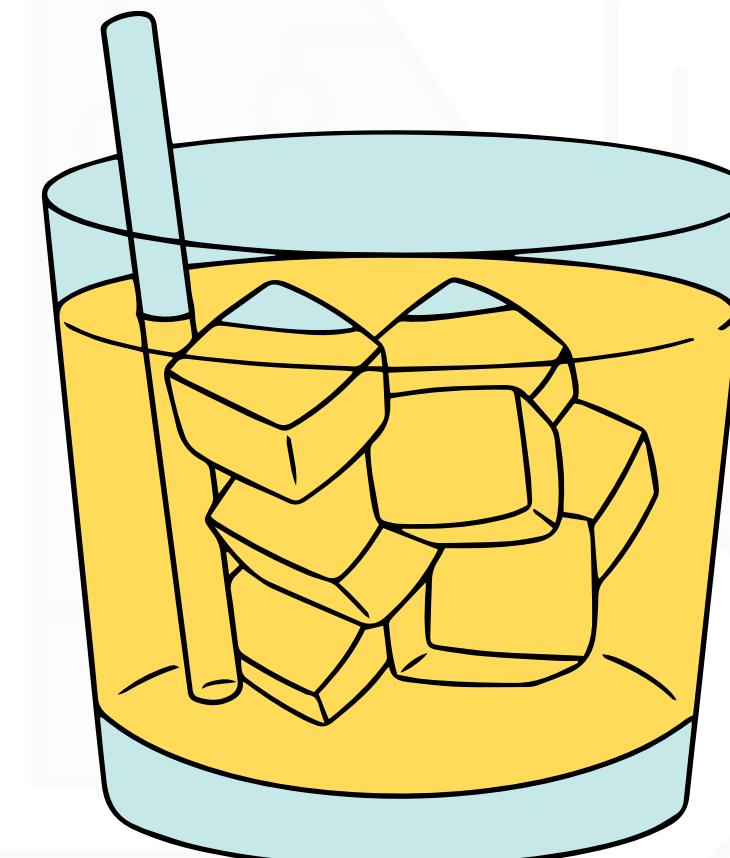
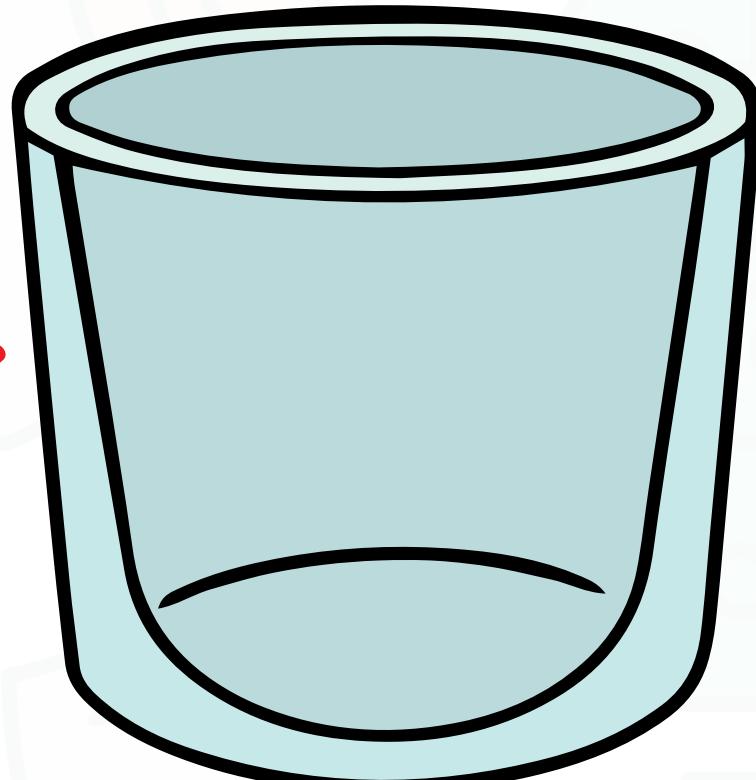


# Analogy

You can also change the value of  $x$  by emptying the cup and filling it with juice. Then you can say that  $x$  is still a variable, but now it has a different value of juice.

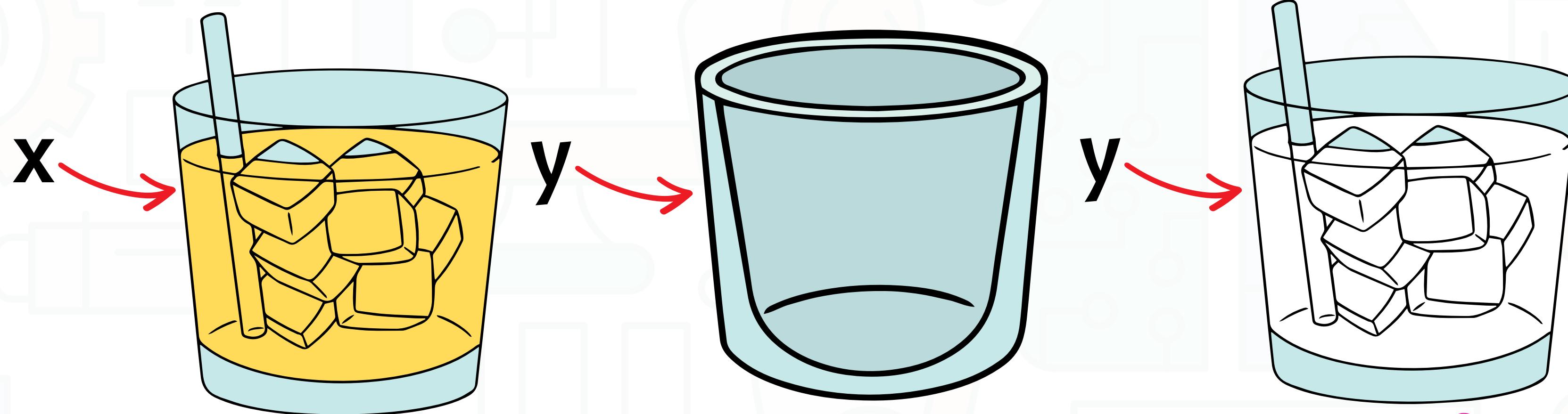


**X**



# Analogy

You can also have another cup named  $y$  and fill it with milk. Then you can say that  $y$  is another variable that has the value of milk.



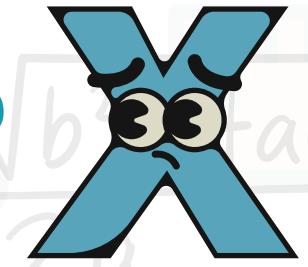
# Analogy

You can also compare the values of  $x$  and  $y$  by looking at the cups and seeing which one has more or less liquid. You can also add or subtract the values of  $x$  and  $y$  by pouring the liquids from one cup to another.

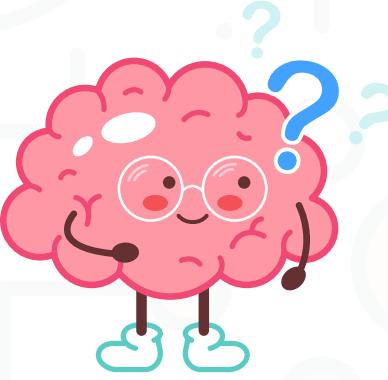


# Think

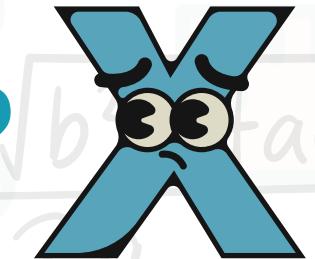
Why are variables important in coding?



# Think

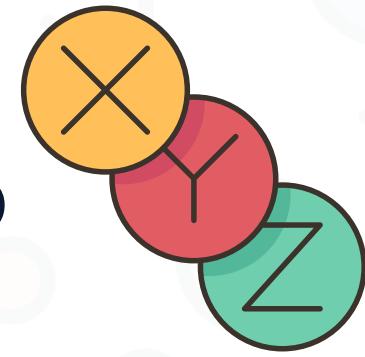


## Why are variables important in coding?



Variables are useful in coding because they can help us store and manipulate data that can vary based on the program's requirements.

# Variables



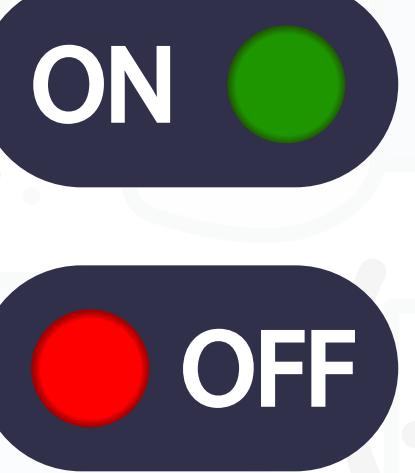
Types of variables:

1 2 3

Numbers

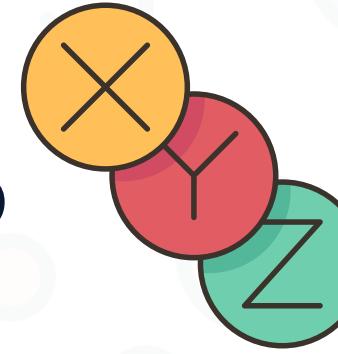
A B C

Letters



Boolen

# Variables



Numbers **123**

Whole number



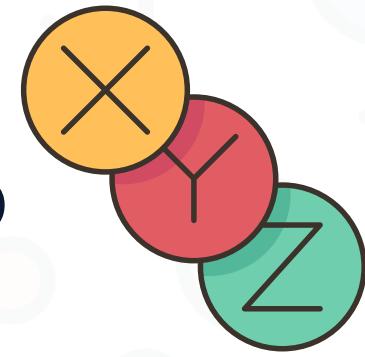
Integer

Decimal



Float

# Variables



Letters **A****B****C**

Letters

A large, stylized letter 'p' with a torn paper texture, appearing to be torn from the left side.

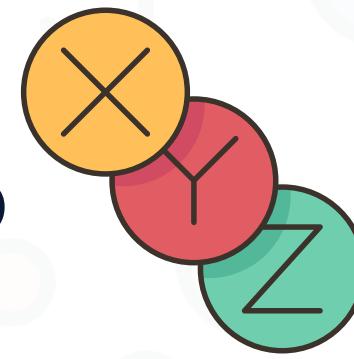
Characters

Words

The word 'smile' composed of individual letters. The 's' is yellow and blue, the 'm' is blue and white, the 'i' is red and white, the 'l' is black and white, and the 'e' is purple and white.

String

# Variables



Boolen:

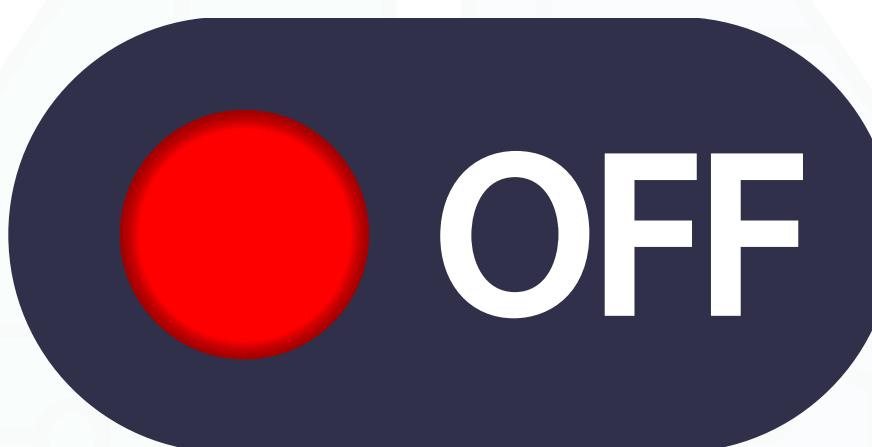


1



HIGH

0



LOW

# How to create variables



Step 1: Choose the type of Variable.

```
int myInteger = 42;
```

```
float pi = 3.14159;
```

```
bool isTrue = true;
```

```
char firstLetter = 'A';
```

# How to create variables



## Step 2: Name the Variable.

```
int myInteger = 42;
```

```
float pi = 3.14159;
```

```
bool.isTrue = true;
```

```
char.firstLetter = 'A';
```

# How to create variables



## Step 3: Giving Value to the Variable.

```
int myInteger = 42;
```

```
float pi = 3.14159;
```

```
bool isTrue = true;
```

```
char firstLetter = 'A';
```

# Let's try it on our robot

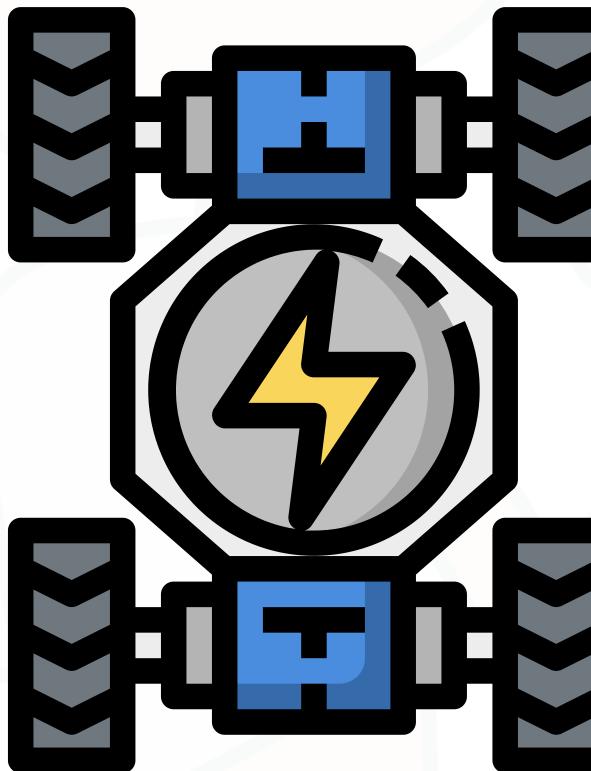


Write code to increase speed gradually

Enable -> 6

Input 1 ->11

Input2 -> 12



Enable -> 10

Input 1 -> 15

Input2 -> 16

Enable -> 5

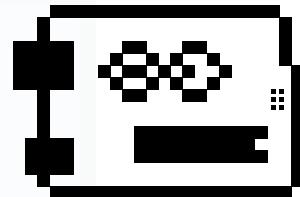
Input 1 -> 7

Input2 -> 8

Enable -> 9

Input 1 -> 13

Input2 -> 14

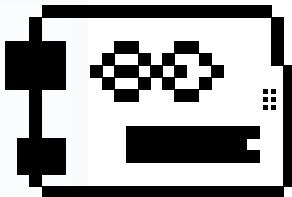


# Increase speed gradually



Step 1: Create variable called speed

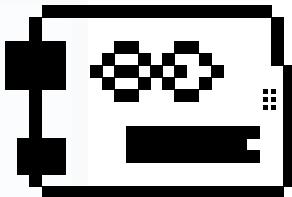
```
1 int speed = ;
```



# Increase speed gradually



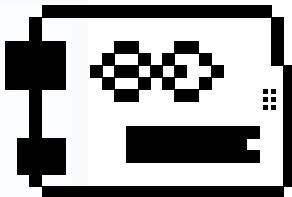
## Step 2: Import forward code



# Increase speed gradually



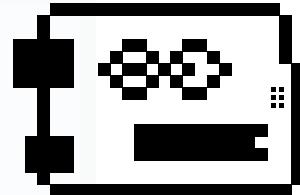
## Step 2: Import forward code



# Increase speed gradually



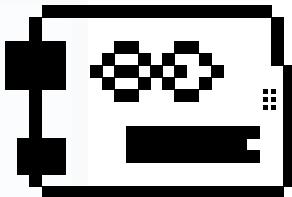
## Step 3: Set speed with 70



# Increase speed gradually



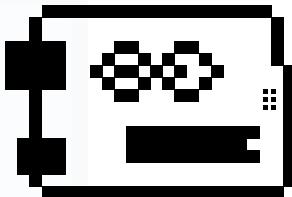
## Step 4: Set PWM pins to speed



# Increase speed gradually



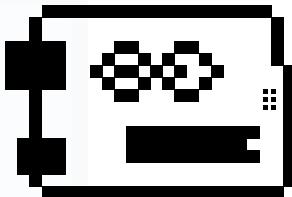
**Step 5: Increase speed by 10 every 3 seconds**



# Increase speed gradually



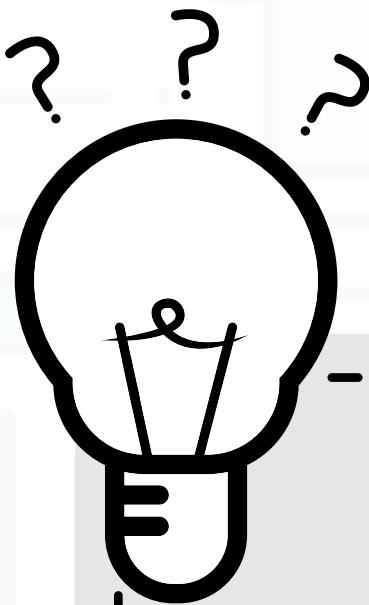
**Step 6: Repeat this 5 times**



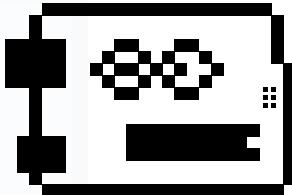
# Increase speed gradually



## Step 7: After 5 times make robot stop



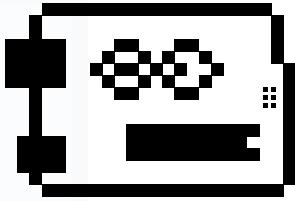
To make robot stop:  
set all motor PWM pins to 0  
set all motor pins to low



# Increase speed gradually



## Step 7: After 5 times make robot stop



# Light dimmer simulation

TIN  
KER  
CAD



# Light dimmer simulation



Step 1: Connect LED to pin 3

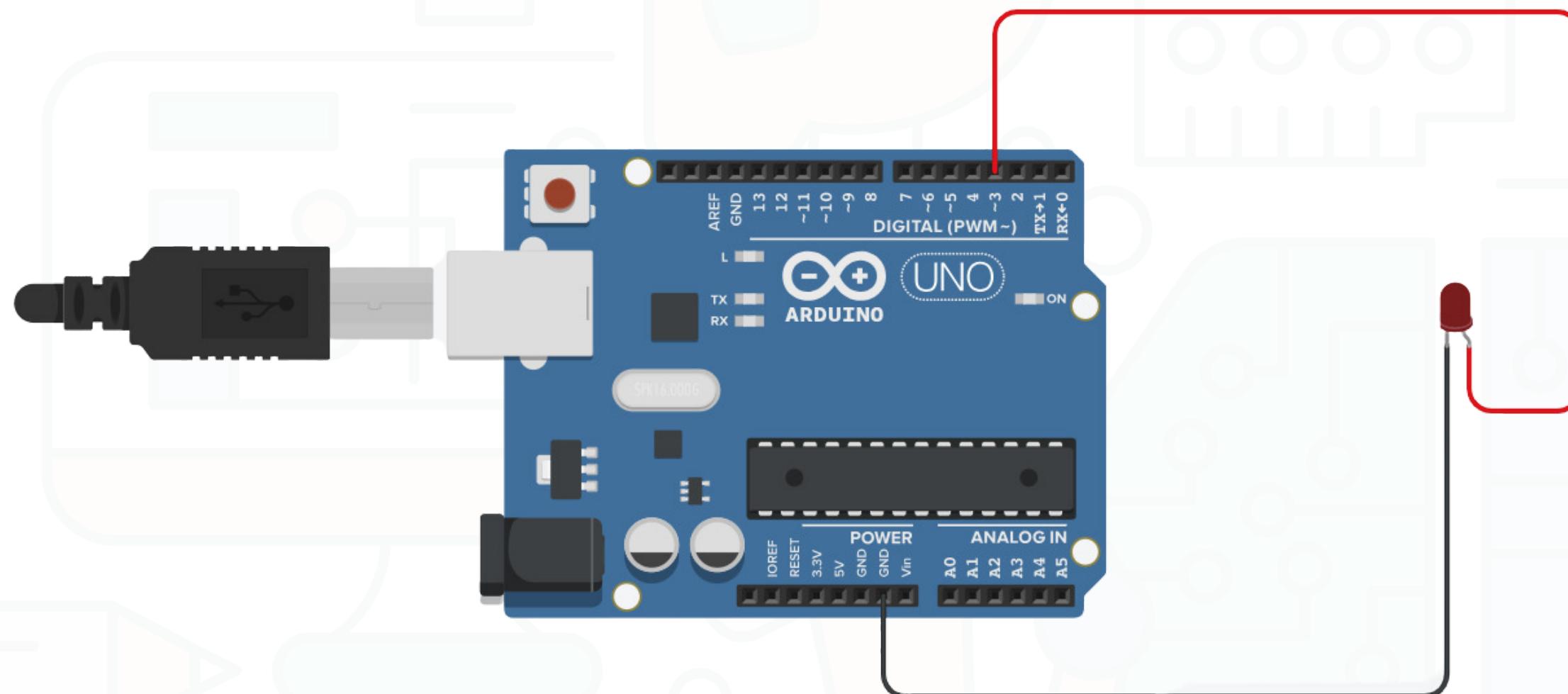
Try it by yourself



# Light dimmer simulation



Step 1: Connect LED to pin 3

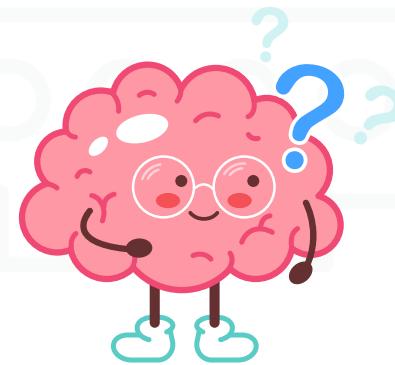


# Light dimmer simulation



**Step 2: Write code to control LED's brightness**

**Think in steps**

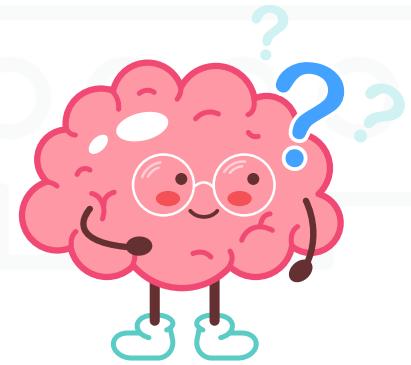


# Light dimmer simulation



Step 2: Write code to control LED's brightness

Think in steps



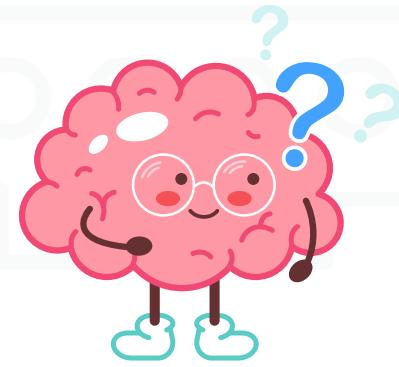
1. I want to start with a turned-off LED

# Light dimmer simulation



Step 2: Write code to control LED's brightness

Think in steps



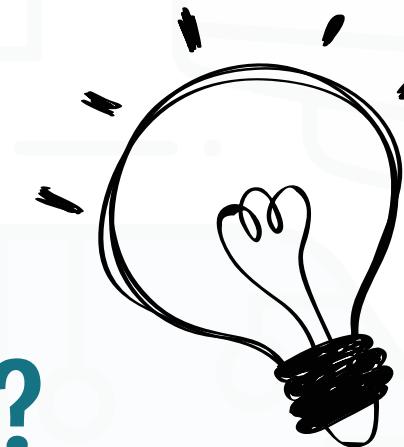
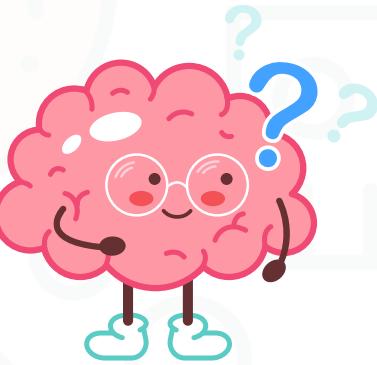
1. I want to start with a turned-off LED
2. Increase its brightness by 10 every 0.5 sec

# Light dimmer simulation



Step 2: Write code to control LED's brightness

Think



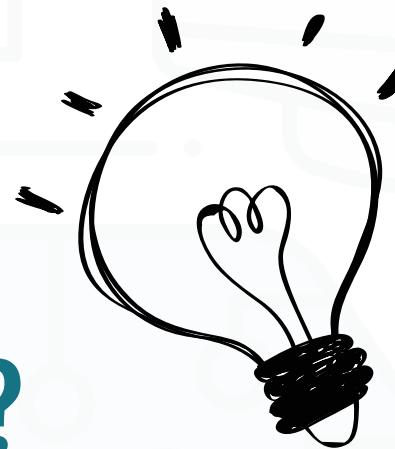
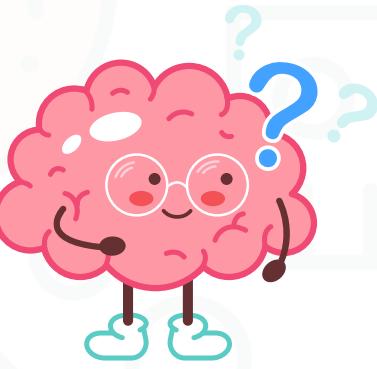
How can we control brightness?

# Light dimmer simulation



Step 2: Write code to control LED's brightness

Think



How can we control brightness?

Using a variable named brightness that can be increased or decreased over time.

# Light dimmer simulation

## Step 2.1: Create brightness variable

```
int brightness = ;
```

# Light dimmer simulation

Step 2.2: Set variable to 0

```
int brightness = 0;
```

# Light dimmer simulation

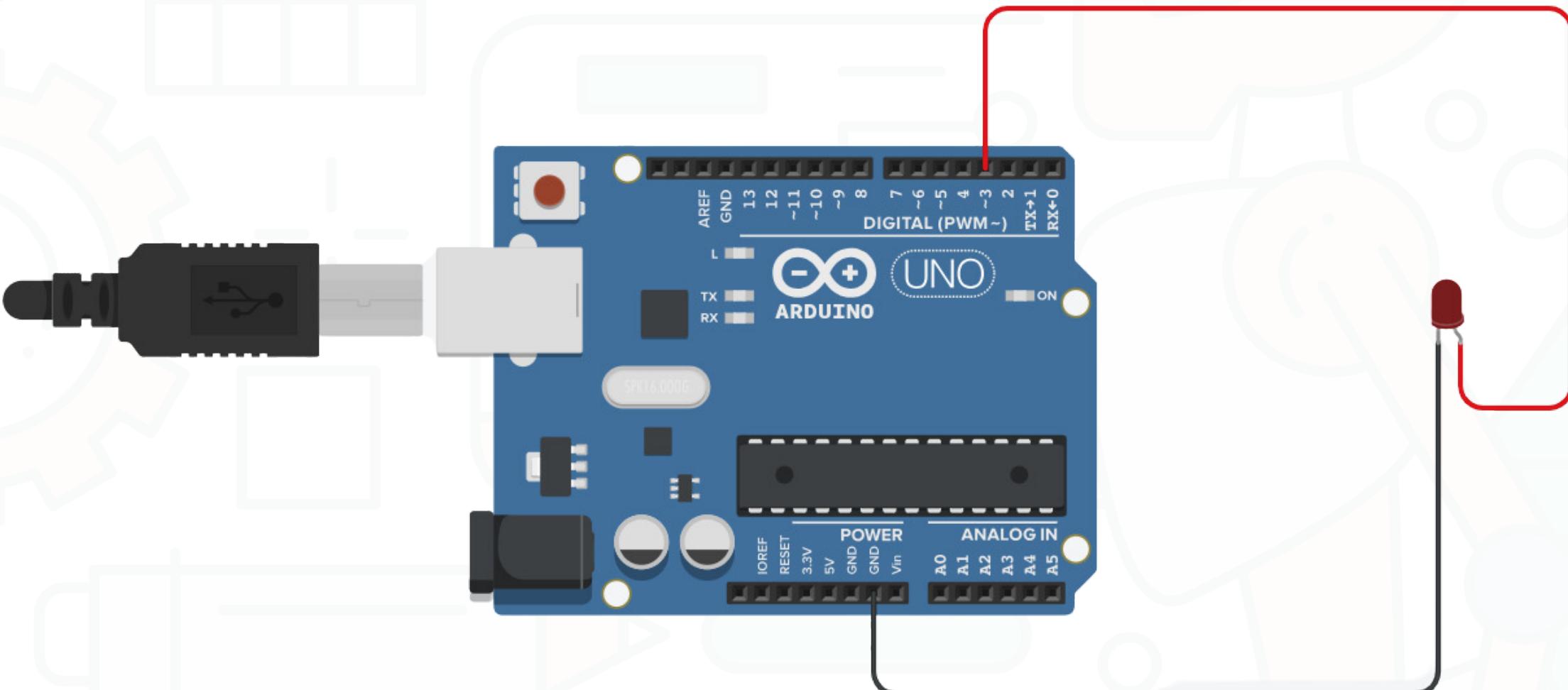
## Step 2.3: Set pin 3 to Brightness

```
3 void setup() {  
4   pinMode(3,OUTPUT);  
5 }
```

```
void loop() {  
  analogWrite(3,brightness);  
}
```

# Light dimmer simulation

## Step 2.4: Wait 0.5 sec and increase brightness by 10



```
void loop() {  
    analogWrite(3, brightness);  
    brightness += 10;  
  
    if (brightness > 255) {  
        brightness = 0;  
    }  
  
    delay(100);  
}
```

