



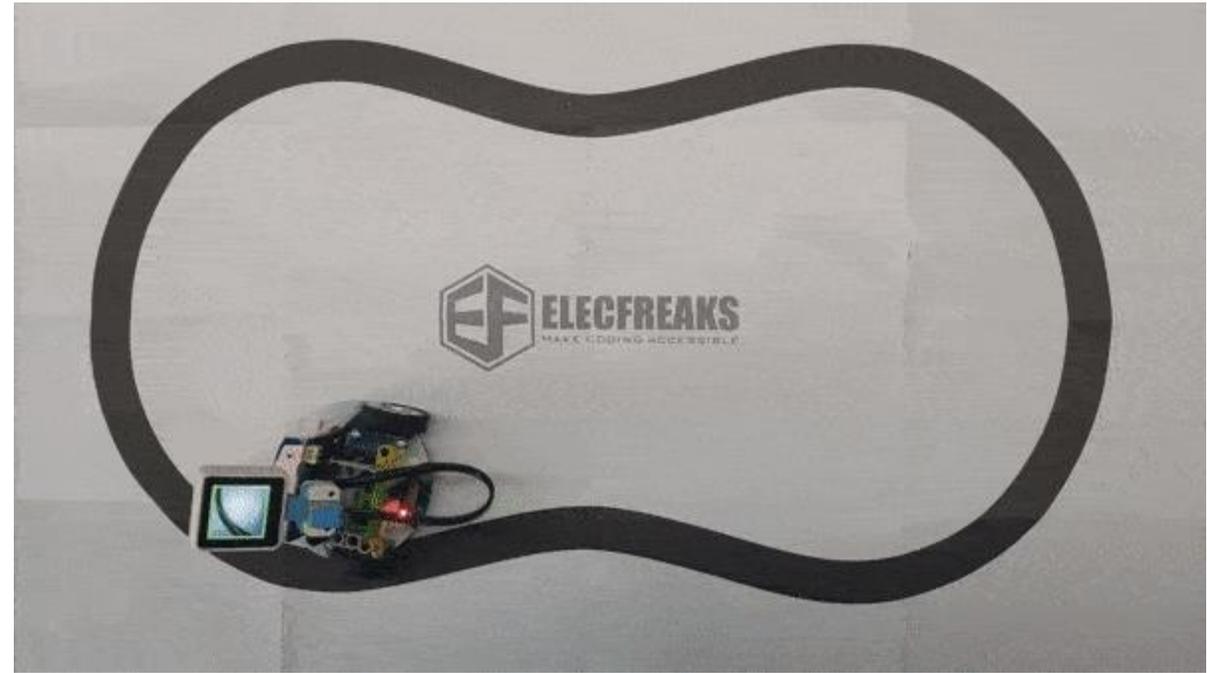
The Cutebot

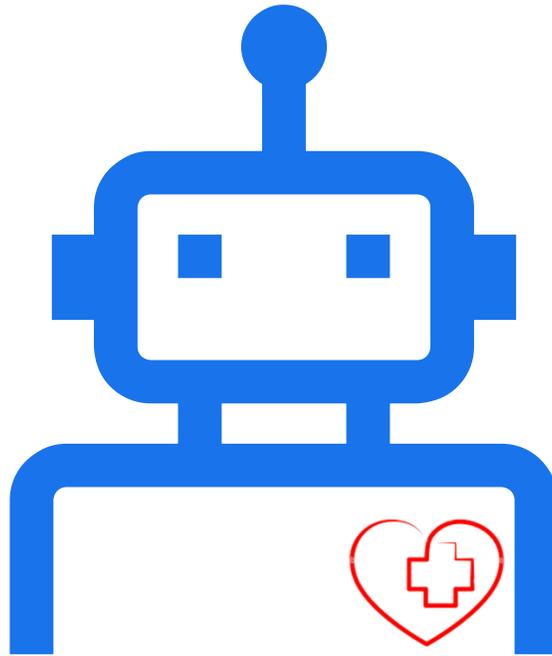


A CAR WITH CUTENESS

What's the goal?

- https://youtu.be/-XvU_iAPCUA
- We will learn how to be safe with technology
- We will learn what the electronics do, and how to put it together
- We will learn the key vocabulary and terms in coding
- You will learn how to code the car to move on a track, then by a remote control
- You will have lots of fun!





How do we get there?



FIRST WE NEED TO LEARN HOW TO BE SAFE WITH OUR
TECHNOLOGY.



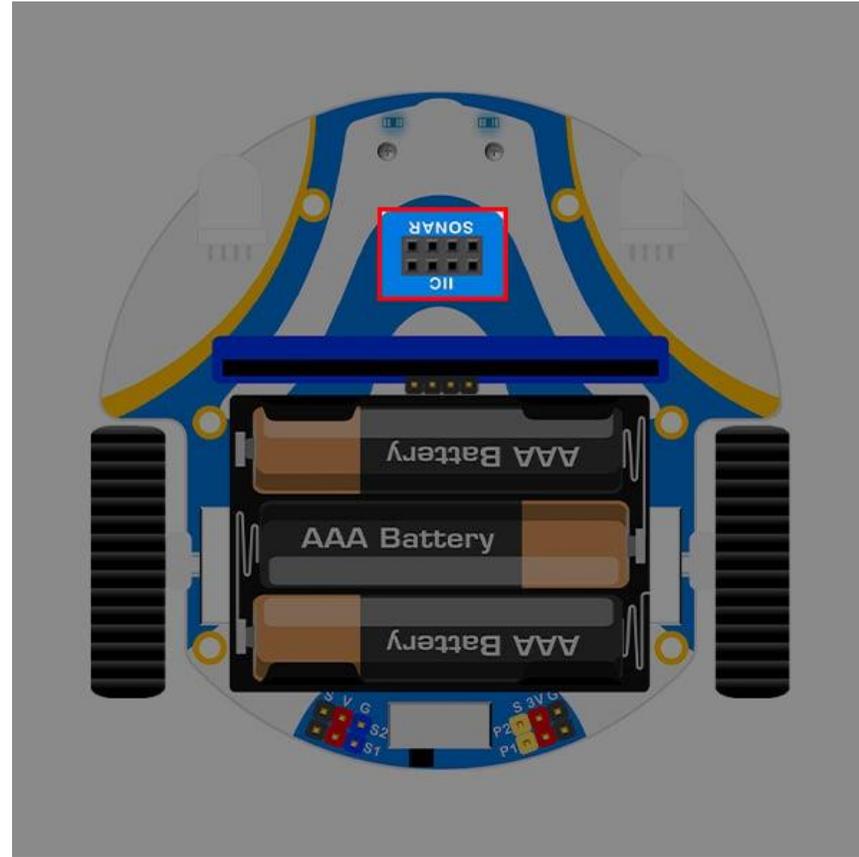
- Water and technology don't mix
- It's not a real car, it cannot carry extra weight
- Please handle all technology as gently as you would handle a baby
- The buttons and switches will break if pushed too hard
- Do not turn off the Microbit when transferring code
- If you break your things, you will not get a new one
- Do not block the cars path, pick it up instead

**Keep me
safe please!**



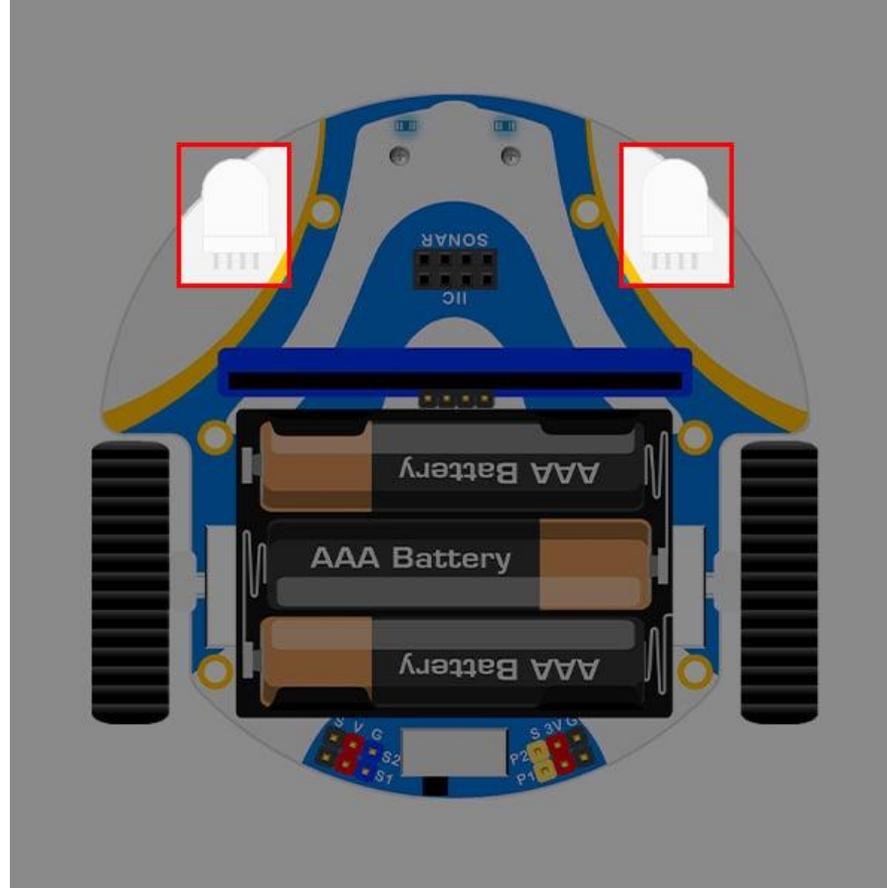
Our Cutebot

- These boxes connect the sensors to the car.
- Make sure you attach the sensor in the front 4 boxes
- These “eyes” can check distances ahead of it



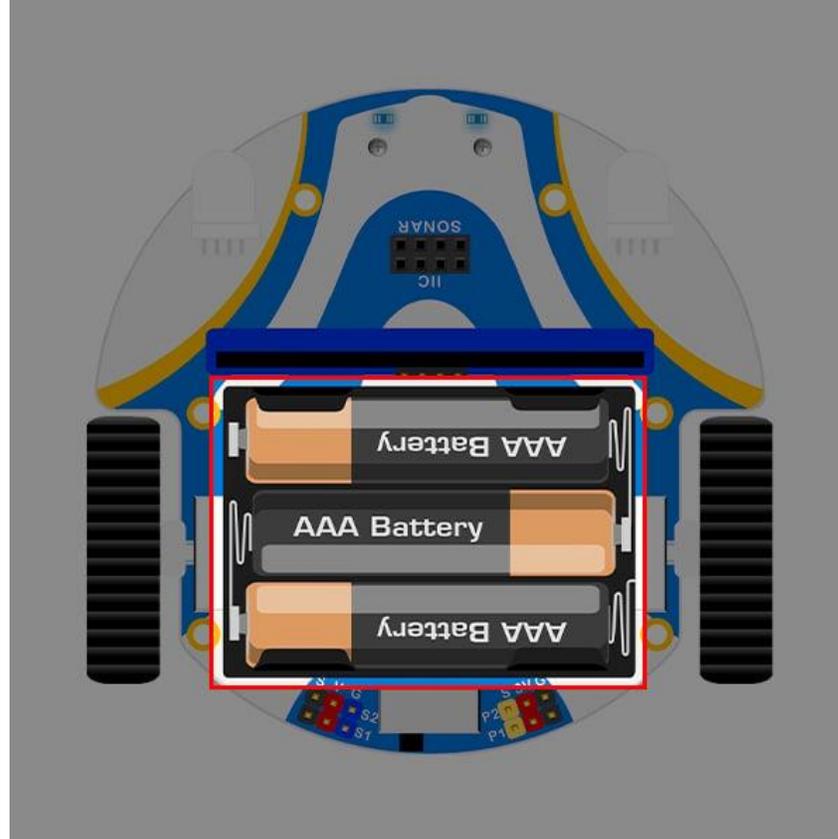
Our Cutebot

- These RGB lights can turn on
- RGB means they can be red, green, and blue, and can mix those colors. Try programming them to turn on later!



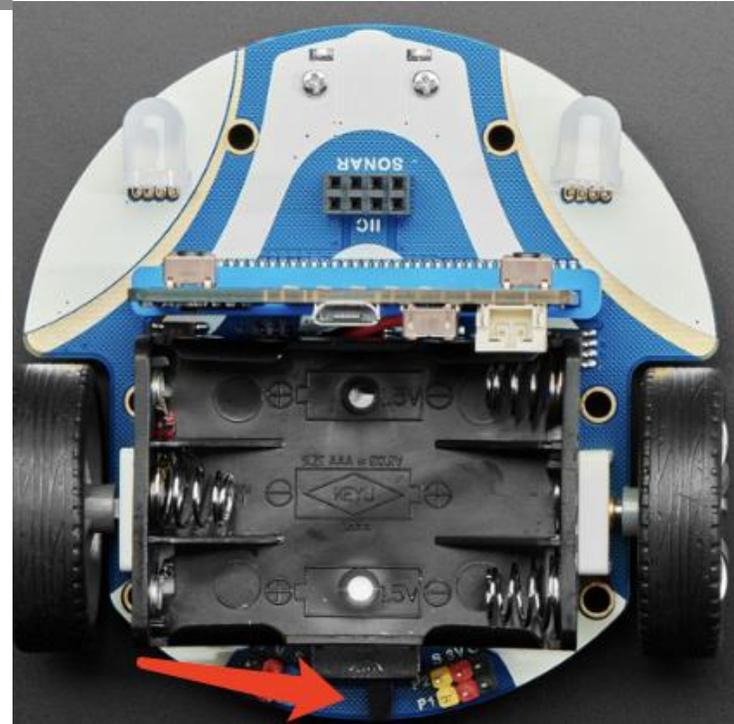
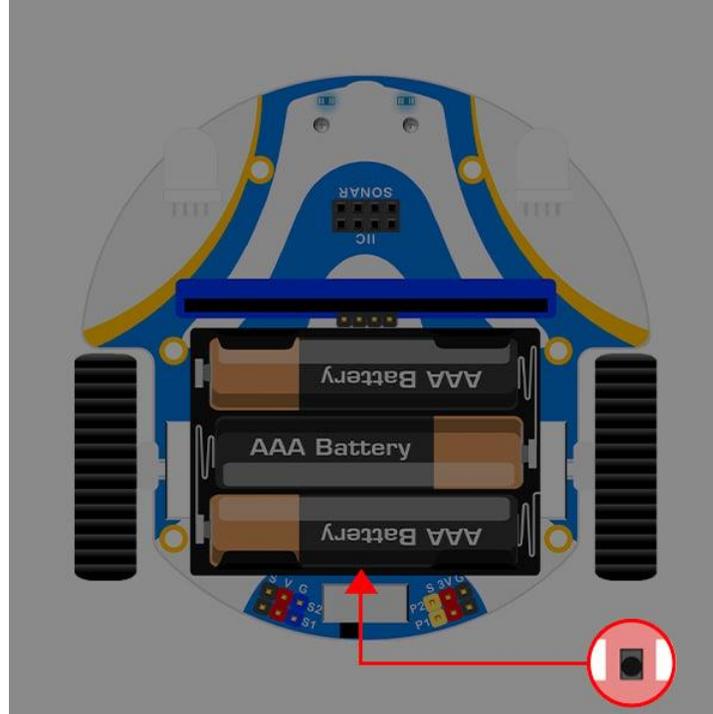
Our Cutebot

- This is where we give the Cutebot power
- It will take 3 AAA batteries. Make sure you turn the Cutebot off to save electricity!



Our Cutebot

- The power switch is hiding behind the batteries
- Turn it off immediately if the code is not working correctly



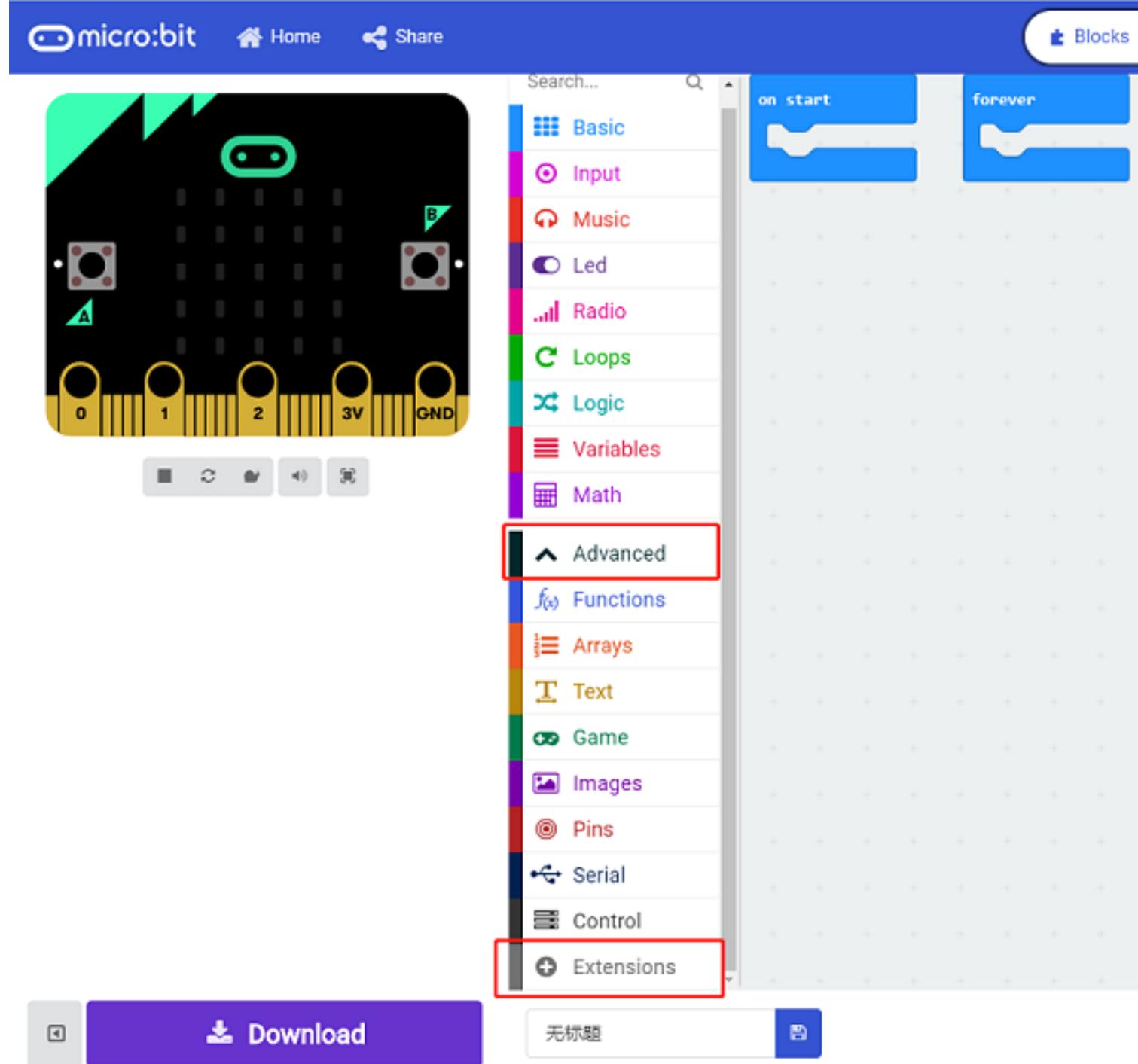
Our Cutebot

- Under the Cutebot are sensors that can detect black lines under it
- There is also a wheel that helps it turn in all directions



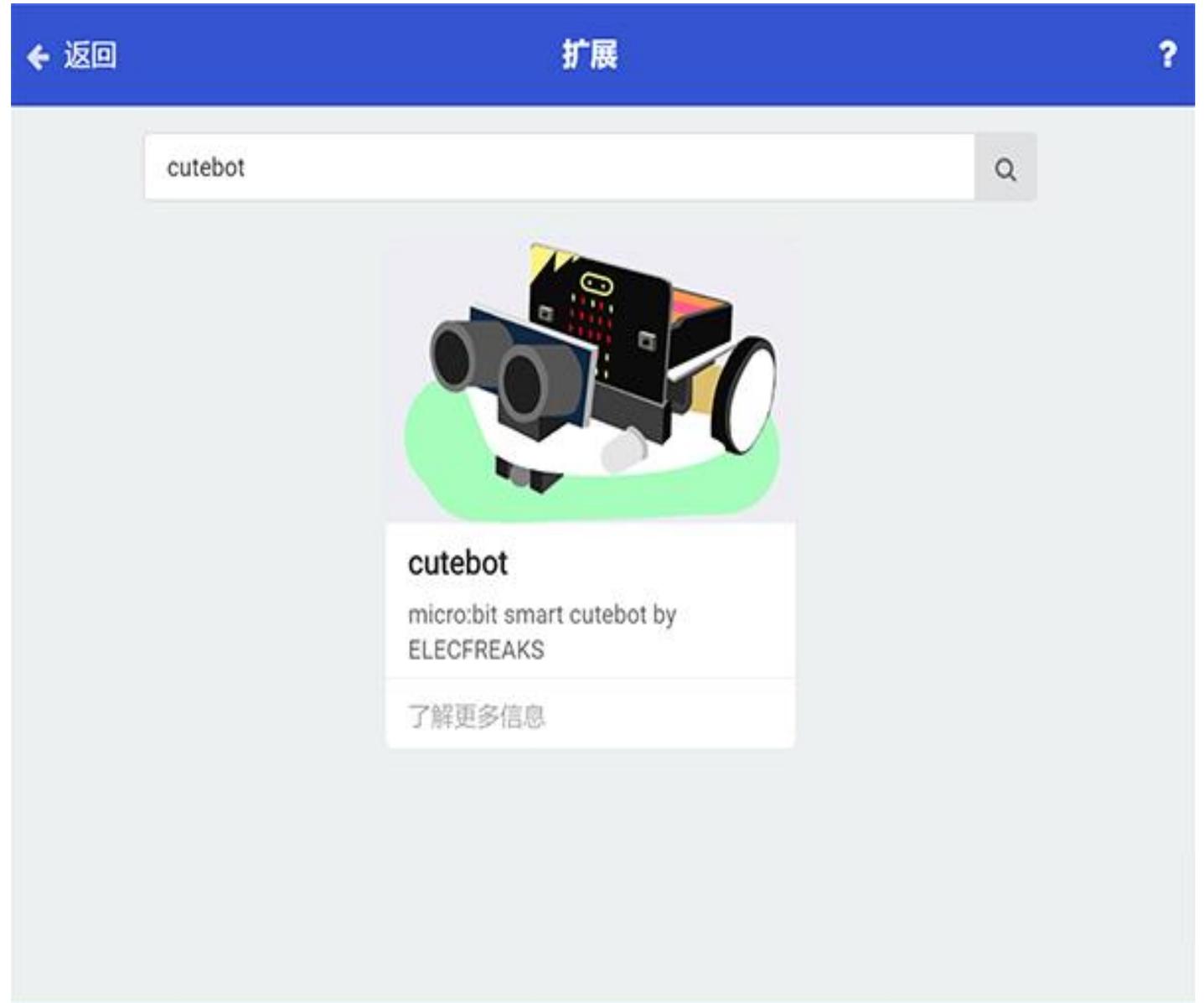
Coding the car

- First we will need to add the expansion for the Cutebot controls
- First click on advanced, then extensions



Coding the car

- Search for Cutebot and add the extension



Coding the car

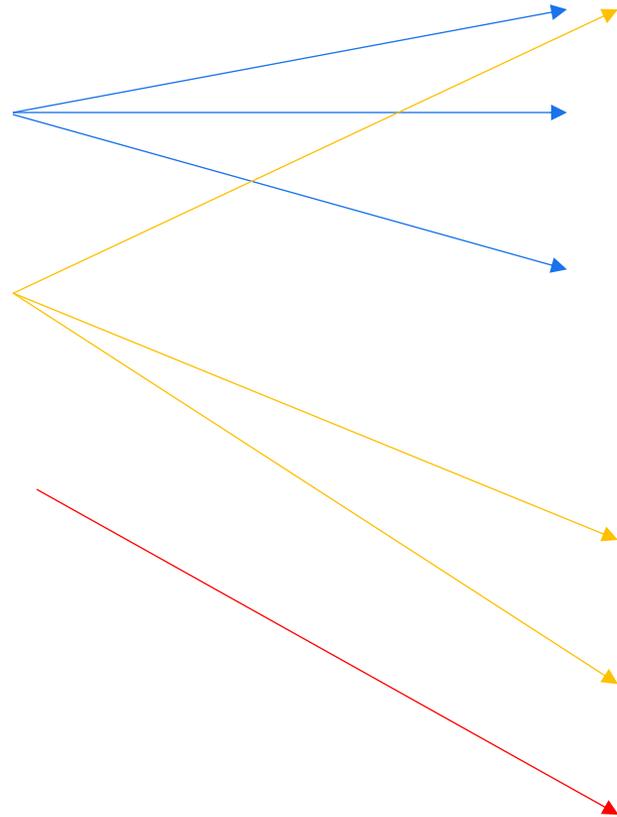
- You will now see all the coding options
- Can you guess what some will do?

The image shows the Scratch IDE interface. On the left, the 'CuteBot' extension is selected in the 'Extensions' menu. The main workspace displays a script for controlling a car. The script consists of the following blocks:

- Set left wheel speed **100** % right wheel speed **-100** %
- Go Forward at speed **50** % for **5** seconds
- Go straight at full speed
- Reverse at full speed
- Turn left at full speed
- Turn right at full speed
- Stop car immediatly
- Set LED headlights Right_RGB color **Red**
- Set LED headlights Right_RGB color R: **0** G: **0** B: **0**
- Turn off all LED headlights
- Tracking state is **...**
- Left line sensor Found
- On Left line Found
- HC-SR04 Sonar unit **cm**
- Set servo **S1** angle to **90** °
- On IR receiving
- IR Button Menu is pressed

Coding the car

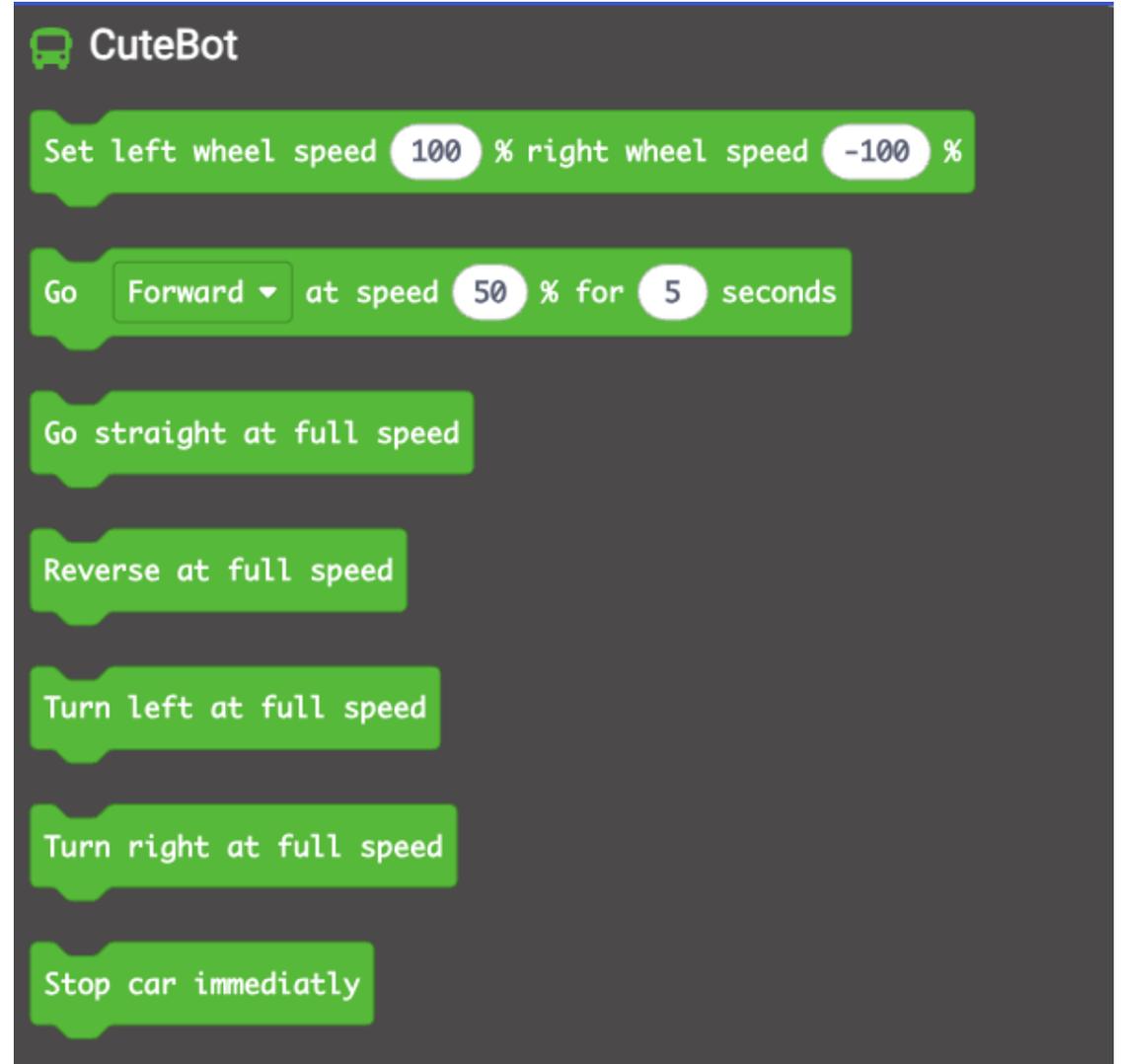
- Going forward!
- Turning!
- Stopping!



```
CuteBot  
Set left wheel speed 100 % right wheel speed -100 %  
Go Forward at speed 50 % for 5 seconds  
Go straight at full speed  
Reverse at full speed  
Turn left at full speed  
Turn right at full speed  
Stop car immediatly 立即地
```

Coding the car

- Can you make your car go forward, then stop?
- What codes would you need?
- Extra challenge! Could you make it come back to where it started?



Coding the car

- To follow the line, we need the sensors to be turned on and know what to do
- We can use tracking 追踪 for this
- Can you tell me why we slow down one wheel when one side doesn't see the line?



```
forever
  if Tracking state is [0] then
    Set left wheel speed 50 % right wheel speed 0 %
  +
  if Tracking state is [1] then
    Set left wheel speed 0 % right wheel speed 50 %
  +
  if Tracking state is [2] then
    Set left wheel speed 50 % right wheel speed 50 %
  +
```

Coding the car

- To follow the line, we need the sensors to be turned on and know what to do
- We can use tracking 追踪 for this
- Can you tell me why we slow down one wheel when one side doesn't see the line?



```
forever
  if Tracking state is [0] then
    Set left wheel speed 50 % right wheel speed 0 %
  +
  if Tracking state is [1] then
    Set left wheel speed 0 % right wheel speed 50 %
  +
  if Tracking state is [2] then
    Set left wheel speed 50 % right wheel speed 50 %
  +
```

Coding the car

- We can also make the car stop when it is too close to something
- We can also make the car reverse if the object gets to close



```
forever
  set sonar to HC-SR04 Sonar unit cm
  if sonar < 20 and sonar > 2 then
    Set left wheel speed 0 % right wheel speed -50 %
    pause (ms) pick random 100 to 200
  else
    Go straight at full speed
```

Coding the remote control

- We can use the radio send functions to send different commands to the car
- These commands will be used for the Microbit in the remote control

```
on start
  radio set group 1
```

```
on button A pressed
  radio send number 1
```

```
on button B pressed
  radio send number 2
```

```
on button A+B pressed
  radio send number 3
```

Coding the car

- These commands will allow the car to react when the buttons are pressed on the remote control
- You can change the cars icon to any one you want

```
on start
  show icon [grid icon]
  radio set group 1
```

```
on radio received receivedNumber
  if receivedNumber = 1 then
    turn left at full speed
  else if receivedNumber = 2 then
    turn right at full speed
  else if receivedNumber = 3 then
    go straight at full speed
```

Coding more things!

- If you are interested in more ways to code your car, you can check out the developer's website [here!](#)
- Thank you for watching and keep that Cutebot going!

