



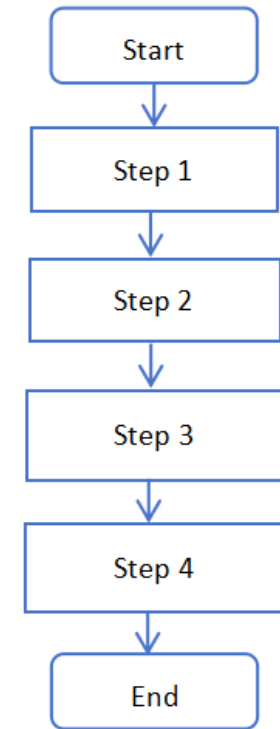
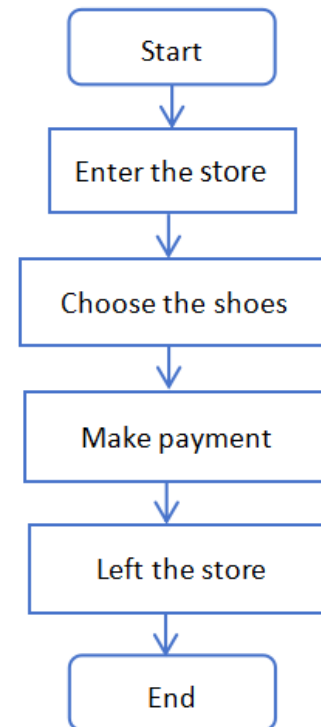
# Course Review

## 1. Sequential structure of a program

Sequential structure programming is the simplest form of program design. You just need to write statements in the order needed to solve the problem. The execution order is top-down, following the sequence of the statements.

### Flowchart Description

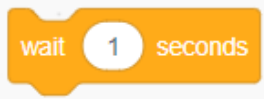
Sean walked into the shoe store, found a pair of running shoes that he liked and fit, paid for them, and finally left the store.



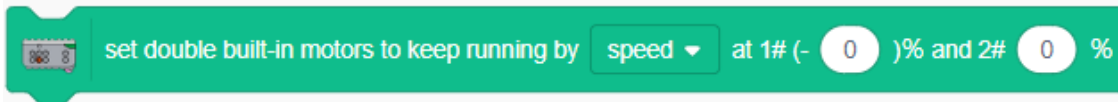


# Course Review

## 2. Module Explanation



You can make the Superbot robot wait for a certain amount of time.



Setting the speed to 0 will make the Superbot robot stop.



# Course Review

3. Control the car's running distance by adjusting the duration or number of rotations of the wheels.

To maintain a consistent distance, you need to reduce the time.

$$\text{Distance} = \text{Speed} \times \text{Time}$$

The equation is annotated with a green arrow pointing up from 'Speed' and a green arrow pointing down from 'Time', indicating that to maintain a constant distance, speed and time are inversely related.

Increase the speed.

A Scratch script starting with a yellow 'when clicked' block. It is followed by a green 'set double built-in motors to keep running by speed at 1# (- 50)% and 2# 50%' block. Below that is an orange 'wait 1 seconds' block. The final block is a green 'set double built-in motors to keep running by speed at 1# (- 0)% and 2# 0%' block. Red dashed lines connect the '50' values in the first block to the '100' values in the second script, and the '1' value in the wait block to the '0.5' value in the second script.

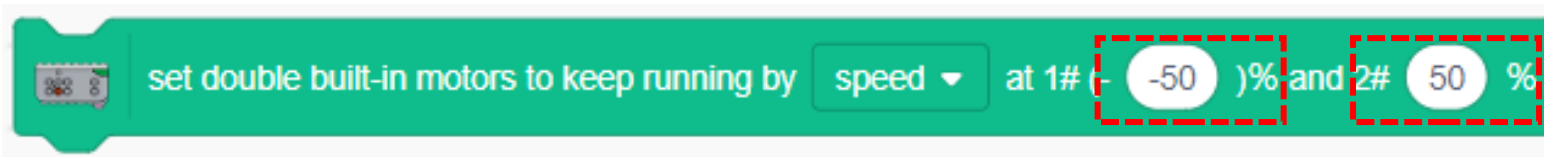
A Scratch script starting with a yellow 'when clicked' block. It is followed by a green 'set double built-in motors to keep running by speed at 1# (- 100)% and 2# 100%' block. Below that is an orange 'wait 0.5 seconds' block. The final block is a green 'set double built-in motors to keep running by speed at 1# (- 0)% and 2# 0%' block. Red dashed lines connect the '50' values in the first script to the '100' values in this script, and the '1' value in the first script's wait block to the '0.5' value in this script.

To ensure the distance traveled remains consistent, you need to reduce the time.

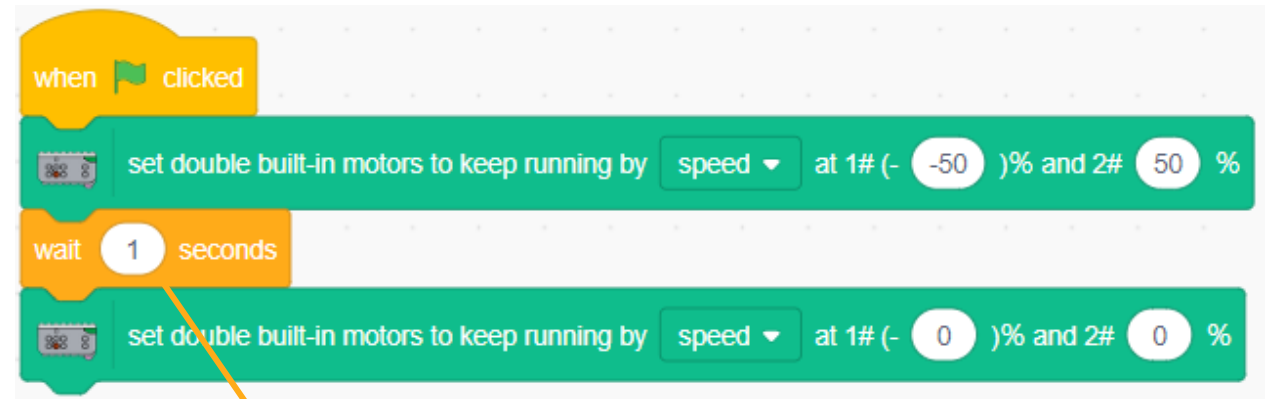
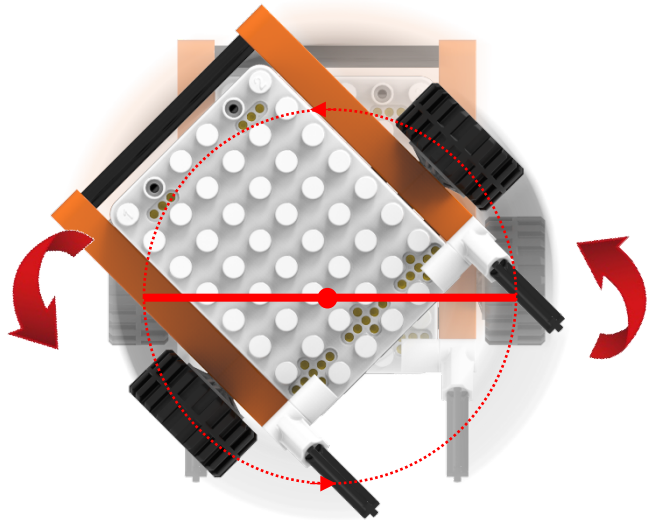


# Course Review

4. The robot turns (one forward and one backward).



Set one motor to move forward and the other motor to move backward, allowing the robot to rotate around the center of the two wheels.



With a constant turning speed, you can adjust the amount of the turn by controlling the time.



# Course Review

## 5. Finish the competition

Adjust the program to enable the robot to perform a U-turn and return.

The image shows three Scratch code blocks. The first block is a green 'set double built-in motors to keep running by speed' block with 'at 1# (-50)% and 2# 50%' selected. The second block is an orange 'wait 1 seconds' block. The third block is a green 'set double built-in motors to keep running by speed' block with 'at 1# (0)% and 2# 0%' selected. An orange arrow points from the '1 seconds' value in the second block to an orange callout box.

With a constant turning speed, you can adjust the amount of the turn by controlling the time.



**Please use your own kit.**

**Do not put any parts in your mouth.**

**Please clean up after use.**

**Please raise your hand if you have any questions.**



# INTRODUCTION

---





# Scenarios

Obstacle racing is a type of car racing where participants must drive their cars over obstacles on the course to reach the finish line. During the race, competitors must avoid hitting any obstacles, which tests their driving skills.



Kids:  
How can you use programming to control the car to navigate around obstacles?  
What are the different ways to turn?

Let's embark on our exploration journey through the "Obstacle Racing"!

# Obstacle Racing

AI Courses

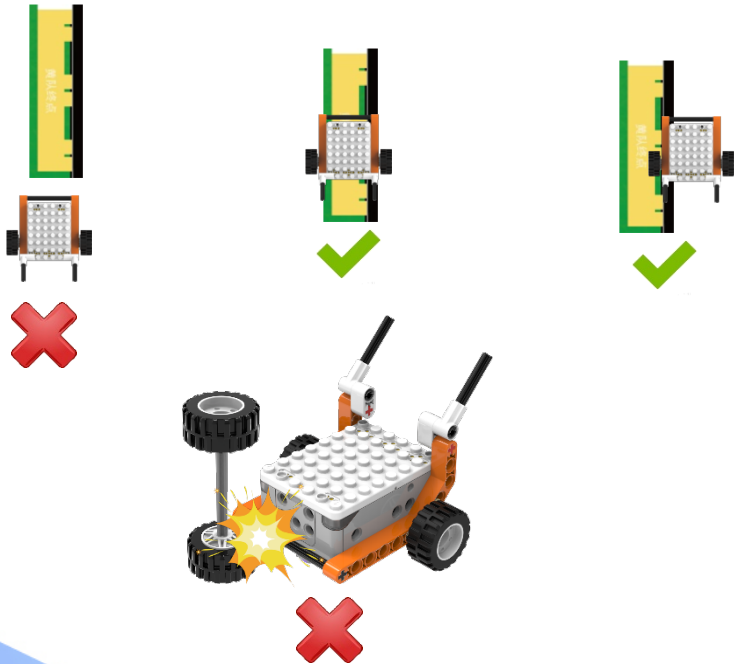




# Scenarios

## Competition rules:

1. The contestant's car starts from the starting point and must reach the designated finish line without hitting any obstacles.
2. Let's see who can complete the course in the shortest time!





# Scenarios

## Question:

Kids, do you know:

How to use programming to control the car  
to navigate around obstacles?

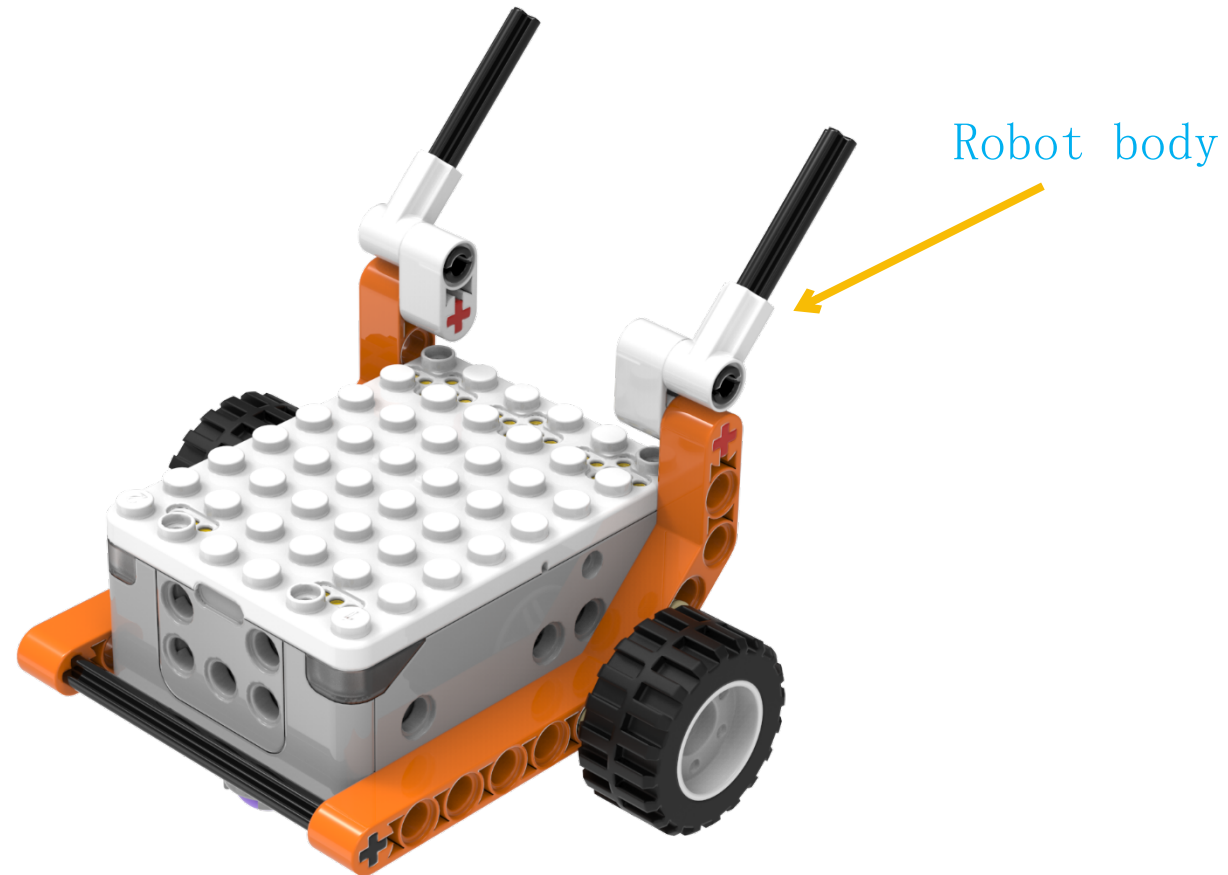
What are the different ways to turn?





# Scenarios

Today, each of you is a junior engineer. Let's work together to complete the obstacle race!



# ASSEMBLY

---





# The Final Model

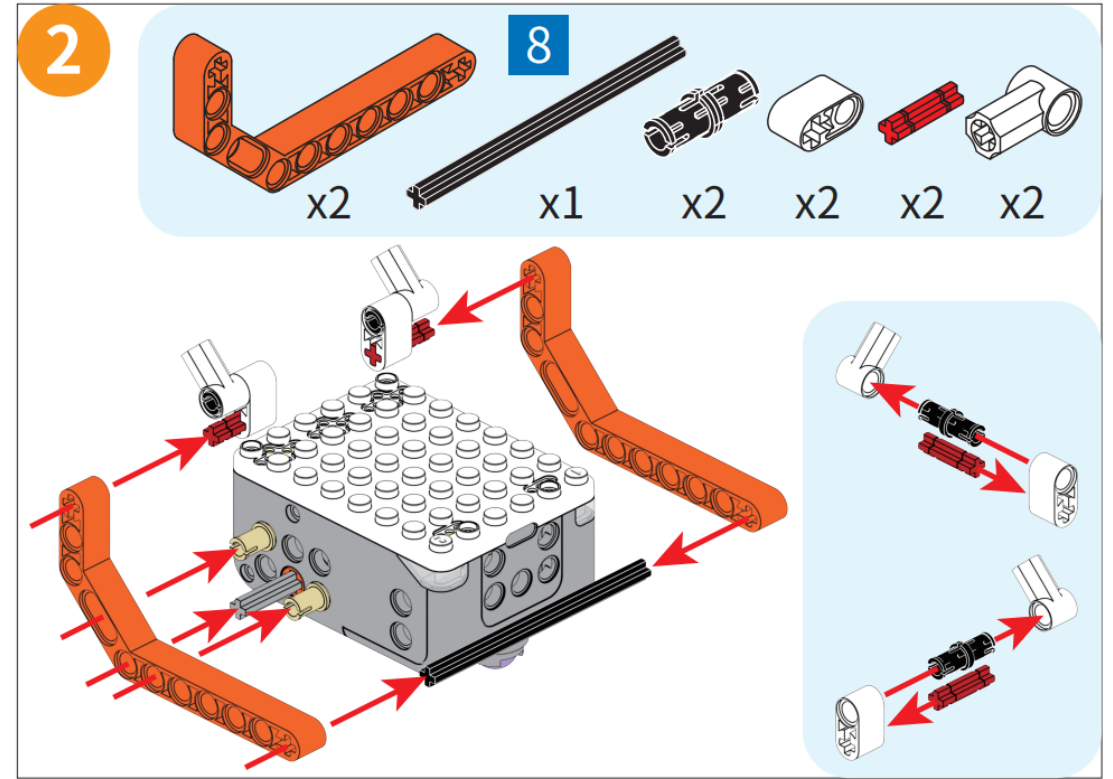
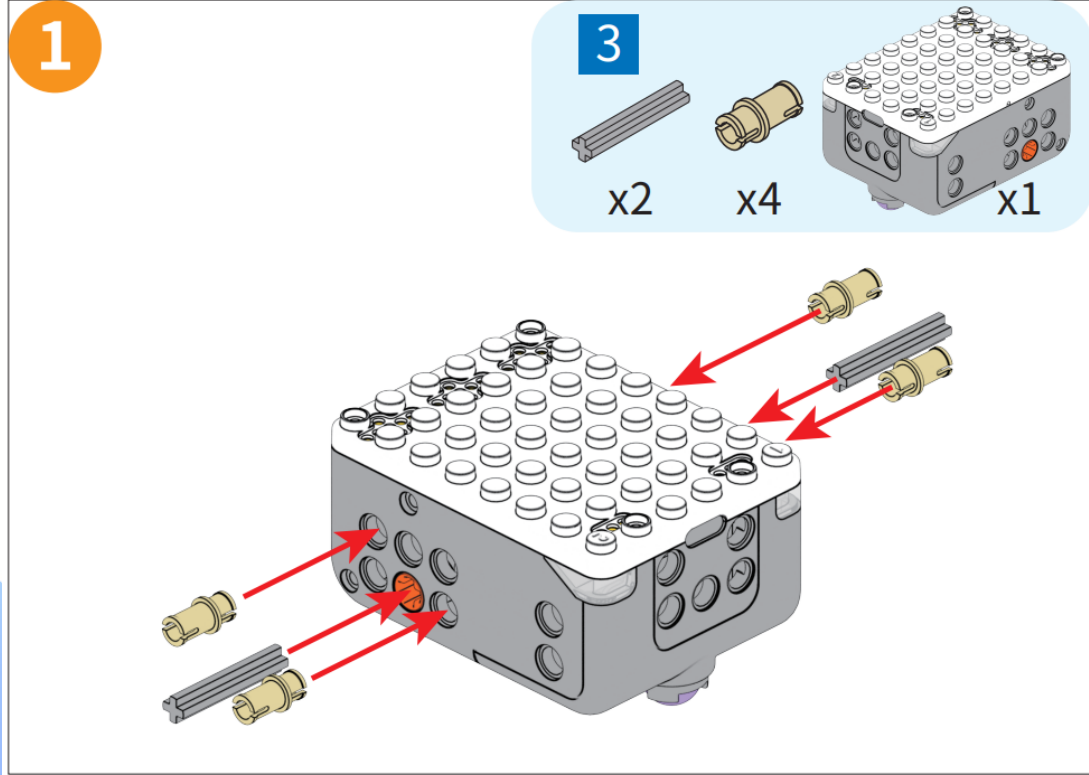
Building the components and assembling them with the robot's main body



Robot body



# Assembly





# The Final Model

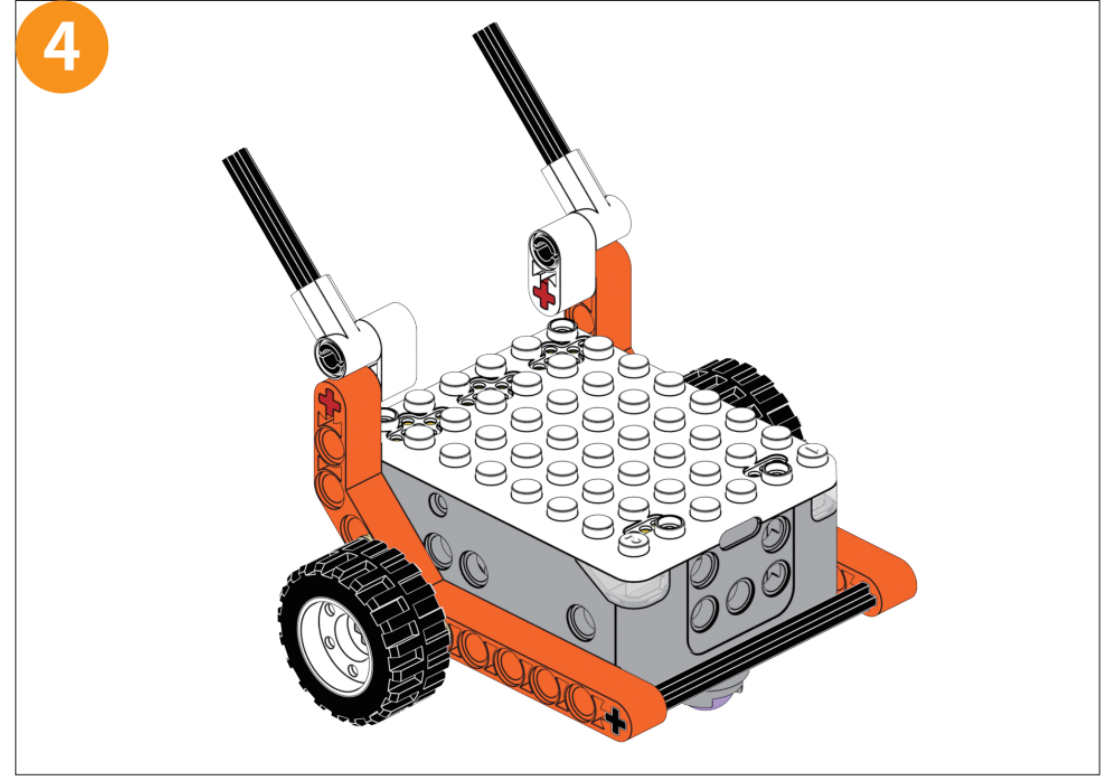
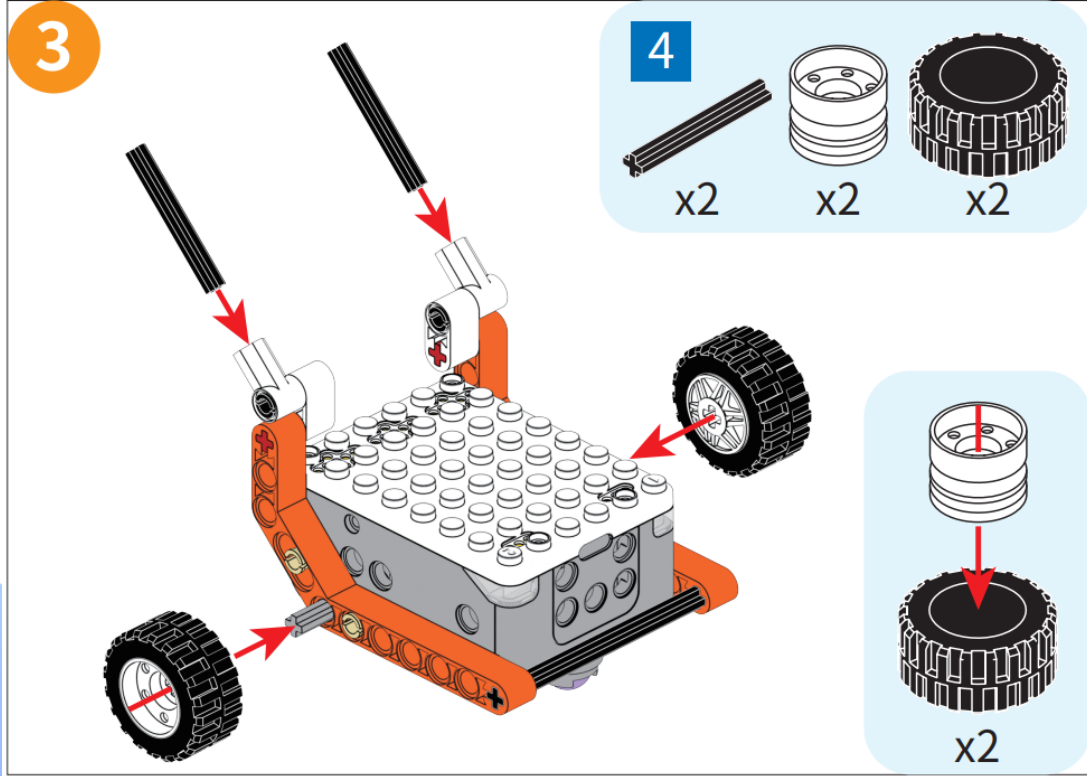
Assemble the wheels



Wheels



# Assembly



# PROGRAMMING

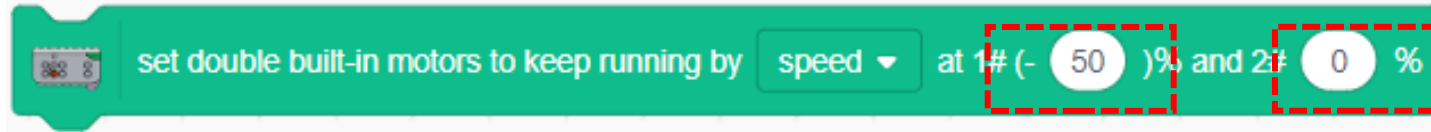
---



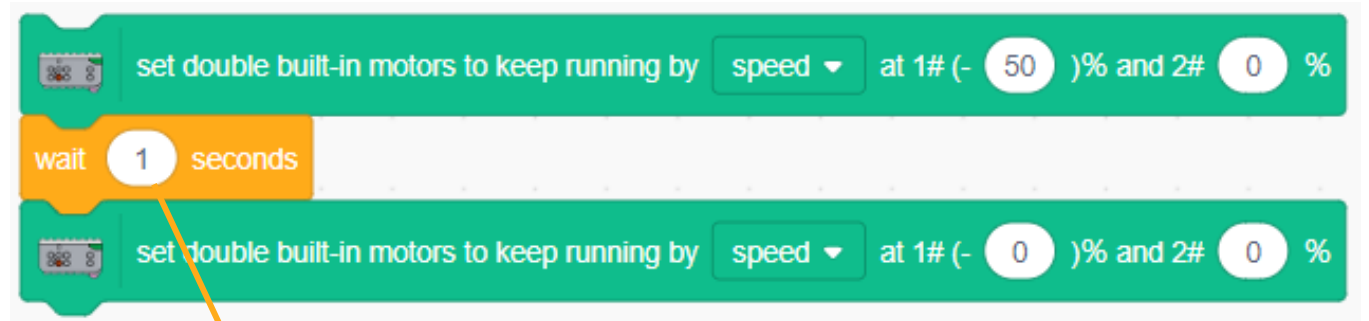
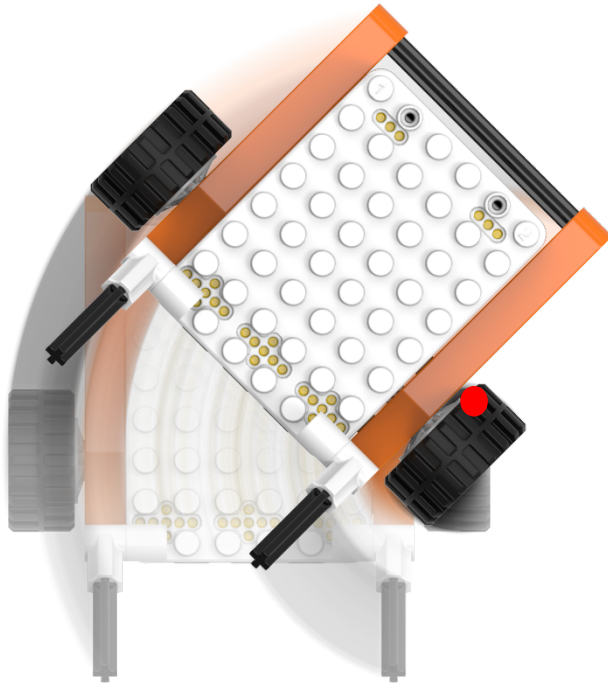


# Introductions

The robot turns (one move and one stop).



Set one motor to move forward and the other motor to stop, allowing the robot to turn around the stationary wheel.

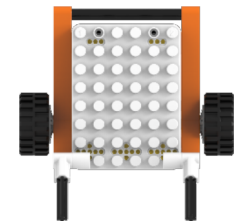
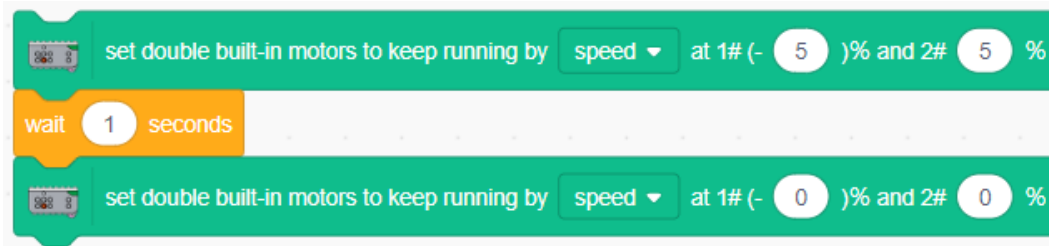
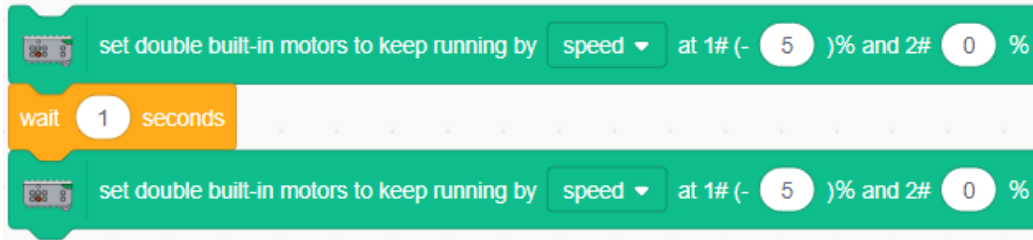
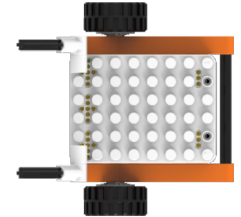
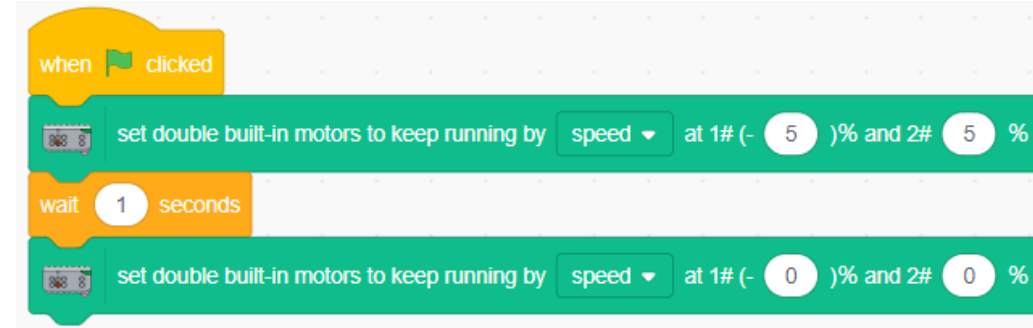
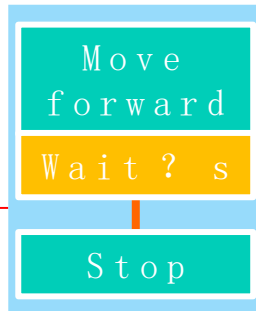
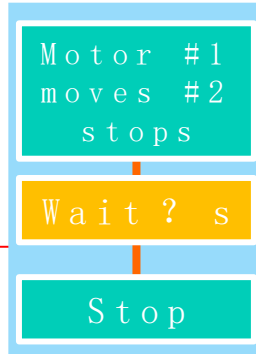
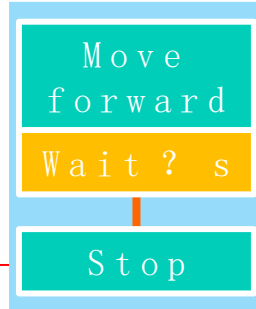
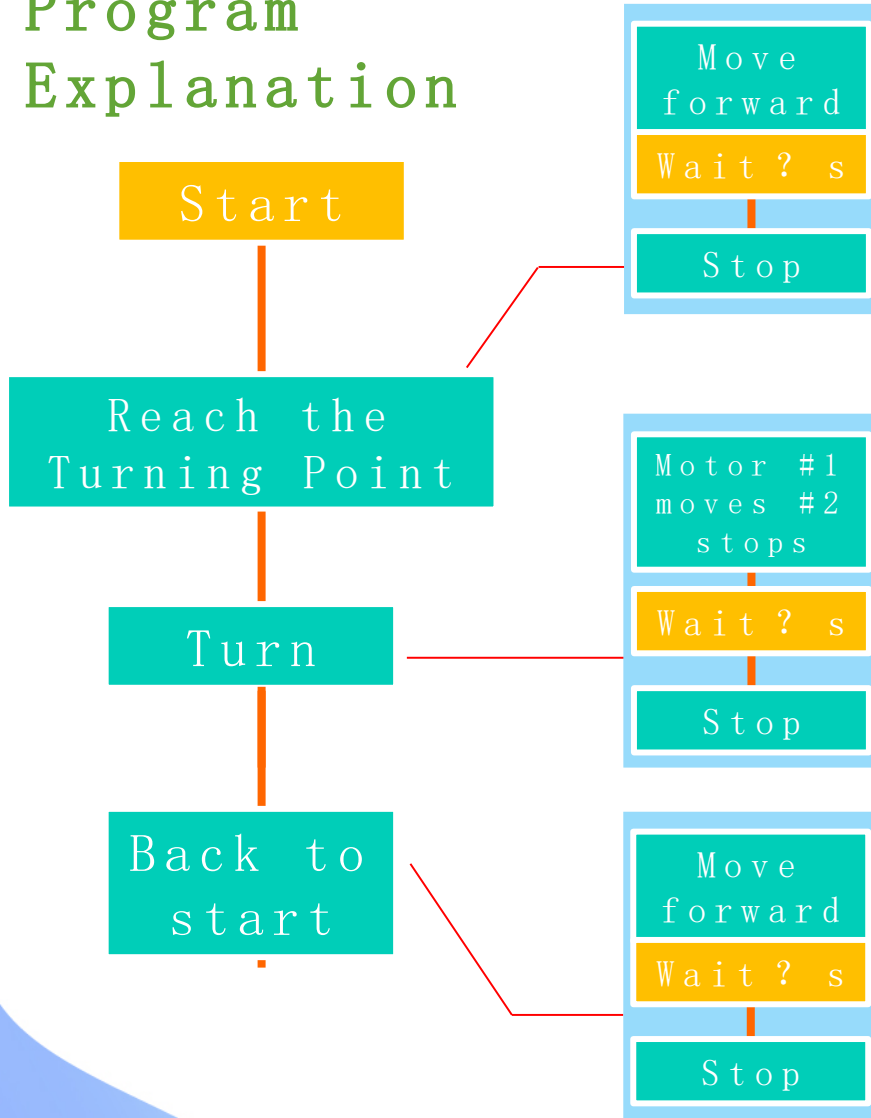


When the turning speed is constant, you can adjust the amount of the turn by controlling the time.



# Introductions

## Program Explanation





# Play and Try

Try It Out:

Click the "Start" button to see if the racing robot can complete the U-turn and return successfully.



```
when clicked
  set double built-in motors to keep running by speed at 1# (- 5 )% and 2# 5 %
  wait 1 seconds
  set double built-in motors to keep running by speed at 1# (- 0 )% and 2# 0 %
  set double built-in motors to keep running by speed at 1# (- 5 )% and 2# 0 %
  wait 1 seconds
  set double built-in motors to keep running by speed at 1# (- 5 )% and 2# 0 %
  set double built-in motors to keep running by speed at 1# (- 5 )% and 2# 5 %
  wait 1 seconds
  set double built-in motors to keep running by speed at 1# (- 0 )% and 2# 0 %
```

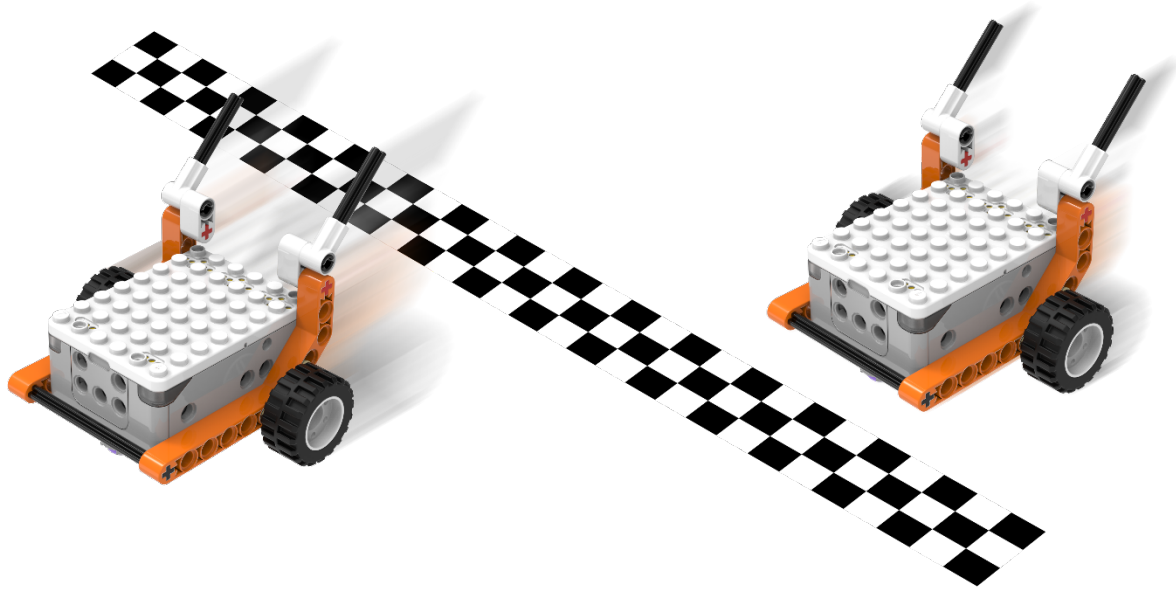
You need to set the waiting time according to the actual conditions.





# Play and Try

Contestants, get ready! The obstacle race is about to begin!



Kids, is there a way to make the robot even stronger?

# CREATION

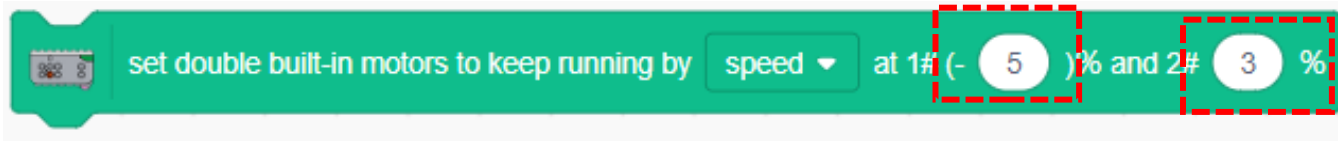
---



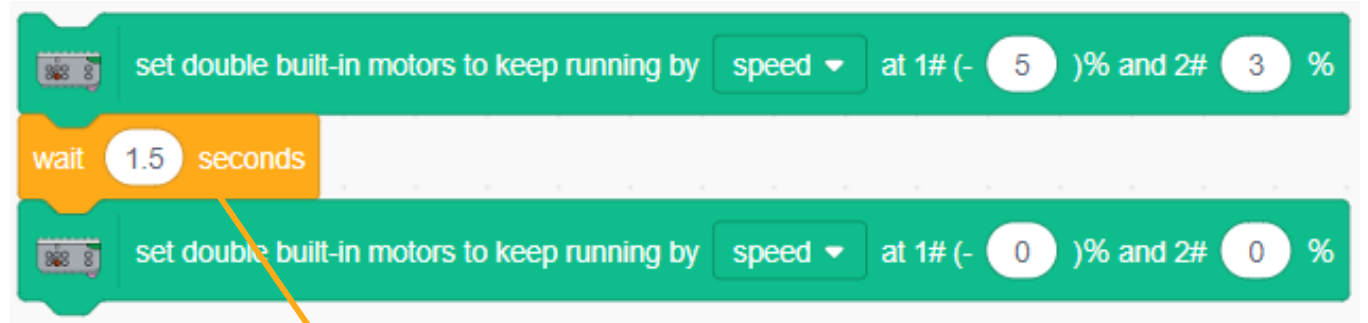
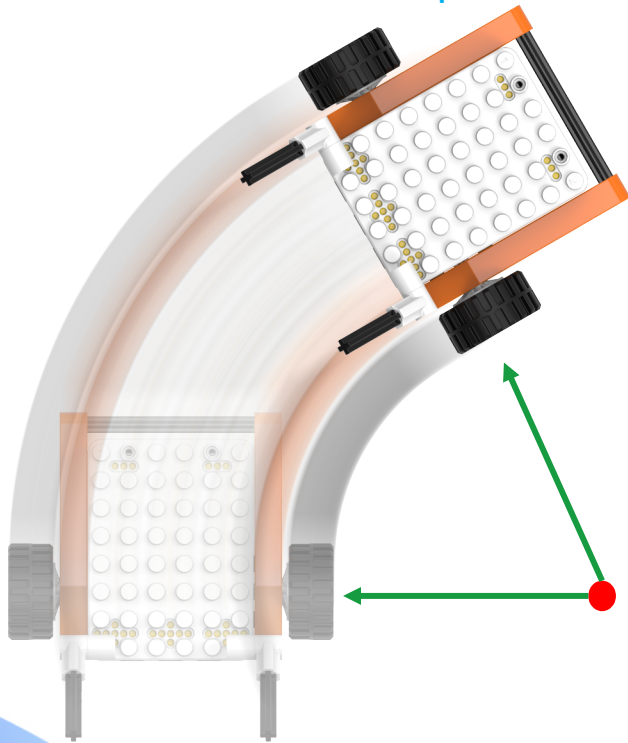


# Introductions

The robot turns (one fast and one slow).



Set one motor to run fast and the other motor to run slow to make the robot turn around a point slightly outside the wheel.

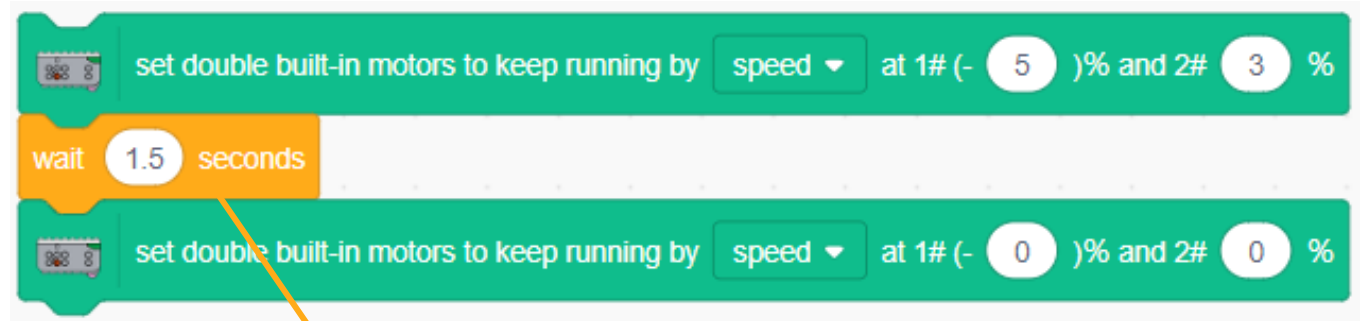


When the turning speed is constant, you can adjust the amount of the turn by controlling the duration.



# Create

Please use the new turning method to complete the obstacle race.

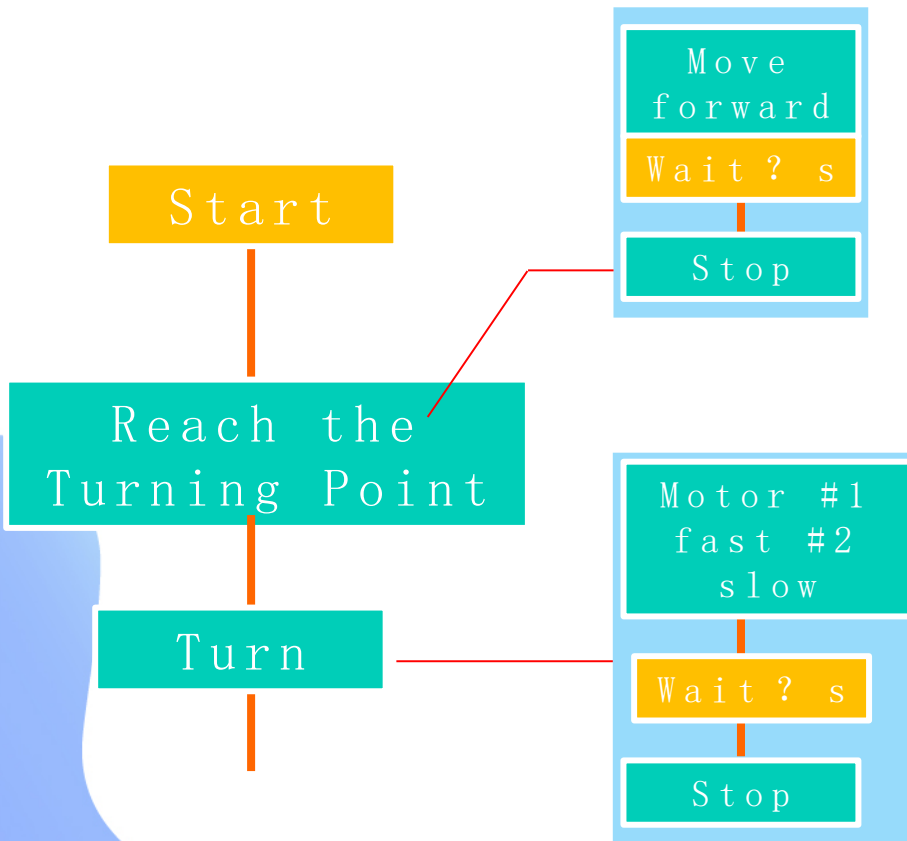


When the turning speed is constant, you can adjust the amount of the turn by controlling the duration.



# Introductions

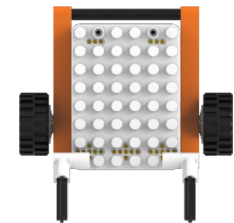
## Program Explanation



```
when clicked
  set double built-in motors to keep running by speed at 1# (- 5 )% and 2# 5 %
  wait 1 seconds
  set double built-in motors to keep running by speed at 1# (- 0 )% and 2# 0 %
```



```
set double built-in motors to keep running by speed at 1# (- 5 )% and 2# 3 %
wait 1 seconds
set double built-in motors to keep running by speed at 1# (- 0 )% and 2# 0 %
```





# Introductions

## Program Explanation

You need to adjust the waiting time according to the actual conditions.

```
when clicked
  set double built-in motors to keep running by speed at 1# (- 5 )% and 2# 5 %
  wait 1 seconds
  set double built-in motors to keep running by speed at 1# (- 0 )% and 2# 0 %
  wait 1 seconds
  set double built-in motors to keep running by speed at 1# (- 5 )% and 2# 3 %
  wait 1 seconds
  set double built-in motors to keep running by speed at 1# (- 0 )% and 2# 0 %
```

Allow the car to wait for a certain amount of time to reduce inertia and improve running accuracy



# Create

1. Complete the obstacle race 2.

Choose the appropriate turning mode to complete the race.





# Create

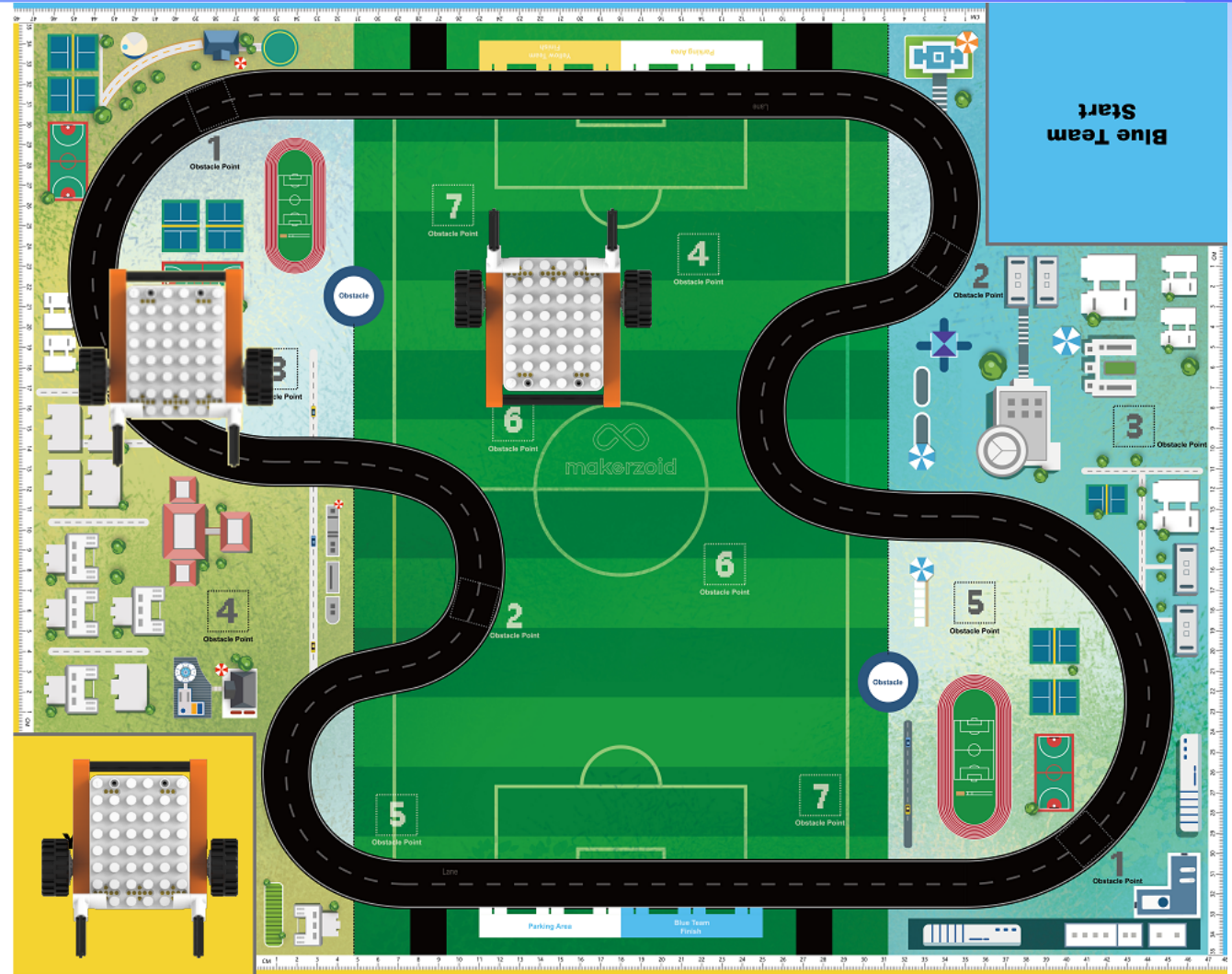
2. Perform a U-turn and return.

Adjust the program to enable the car to perform a U-turn and return.

The code block sequence consists of three blocks:

- A green 'set double built-in motors to keep running by speed' block with a dropdown menu set to 'speed', a minus sign, a circle containing '5', a percent sign, and a circle containing '3' followed by a percent sign.
- An orange 'wait' block with a circle containing '1' followed by the word 'seconds'.
- A green 'set double built-in motors to keep running by speed' block with a dropdown menu set to 'speed', a minus sign, a circle containing '0', a percent sign, and a circle containing '0' followed by a percent sign.

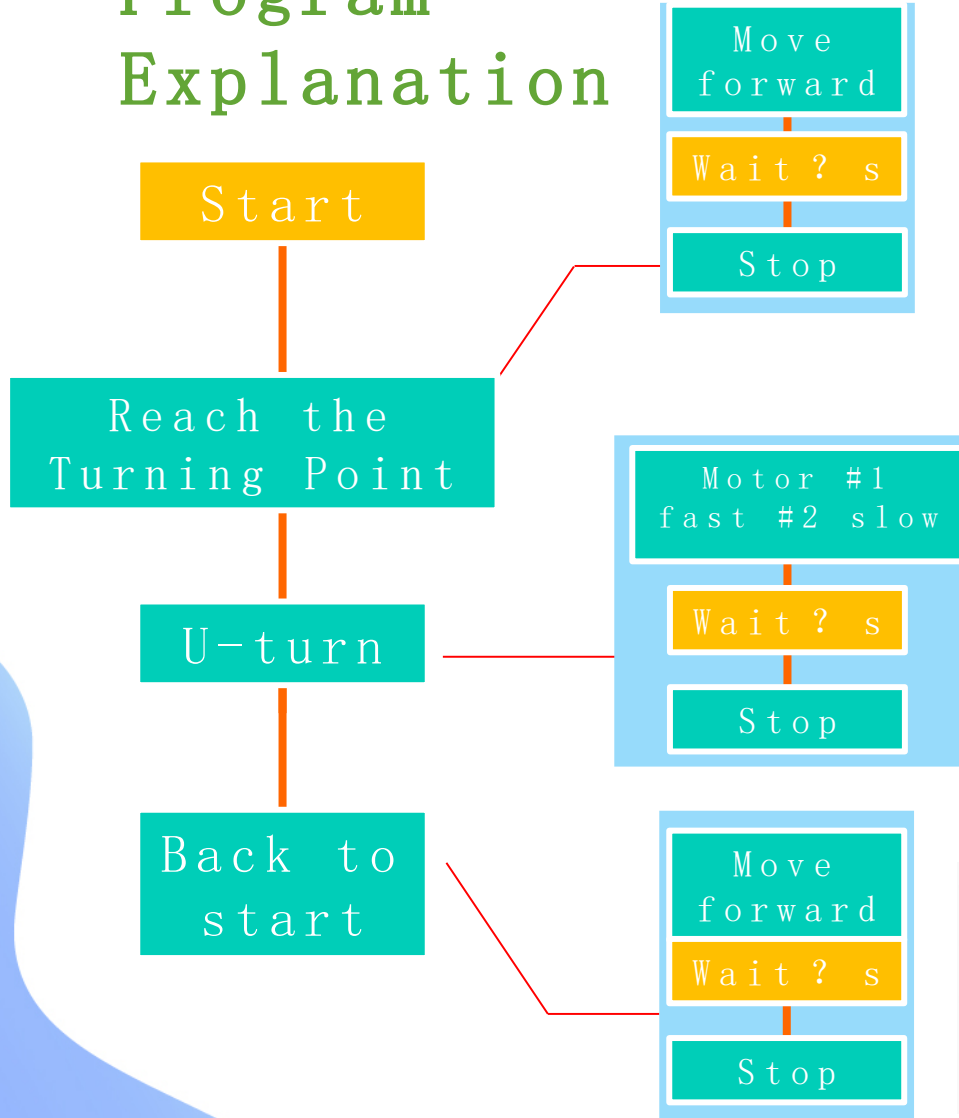
With a constant turning speed, you can adjust the amount of the turn by controlling the duration.





# Introductions

## Program Explanation



```
when clicked
  set double built-in motors to keep running by speed at 1# (- 5 )% and 2# 5 %
  wait 1 seconds
  set double built-in motors to keep running by speed at 1# (- 0 )% and 2# 0 %
  wait 1 seconds
```

```
set double built-in motors to keep running by speed at 1# (- 9 )% and 2# 3 %
wait 1.65 seconds
set double built-in motors to keep running by speed at 1# (- 0 )% and 2# 0 %
wait 1 seconds
```

```
set double built-in motors to keep running by speed at 1# (- 5 )% and 2# 5 %
wait 1 seconds
set double built-in motors to keep running by speed at 1# (- 0 )% and 2# 0 %
```

# SUMMARY

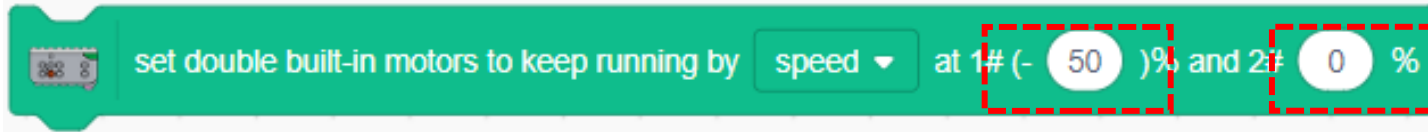
---



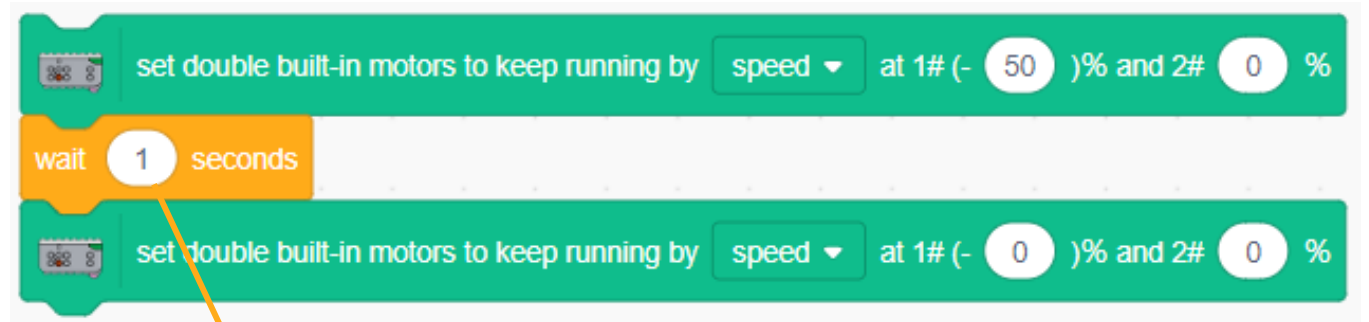
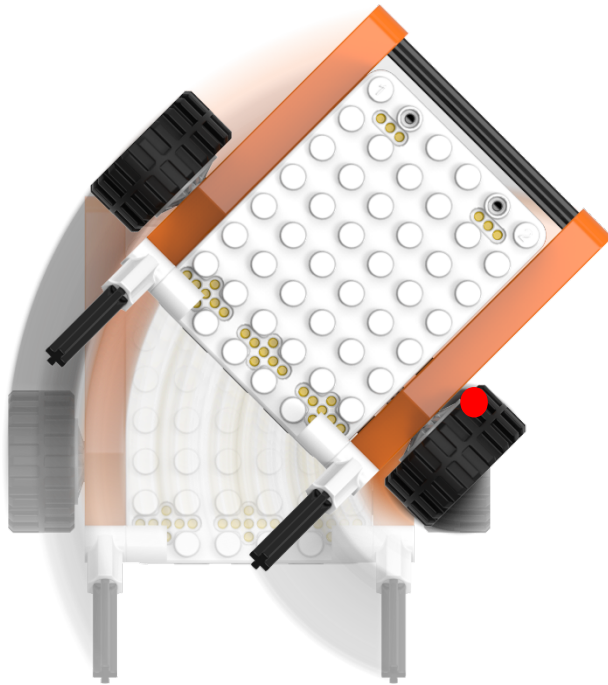


# Summary

1. The robot turns (one move and one stop).



Set one motor to move forward and the other motor to stop, allowing the robot to turn around the stationary wheel.

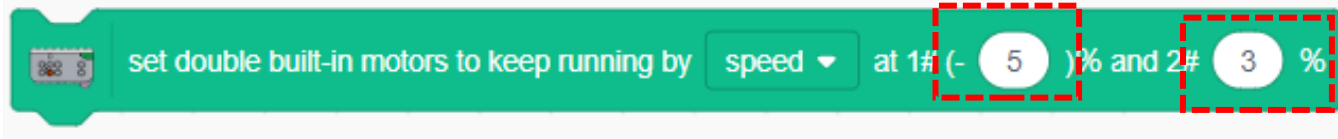


When the turning speed is constant, you can adjust the amount of the turn by controlling the time.

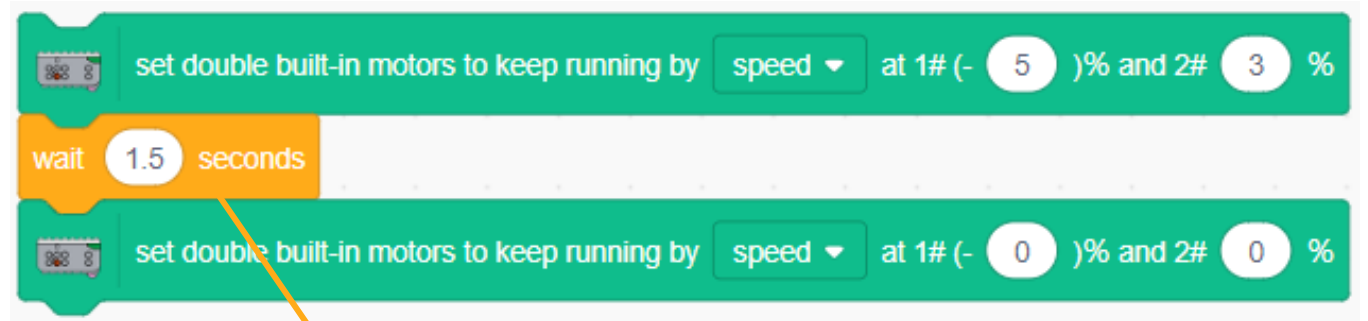
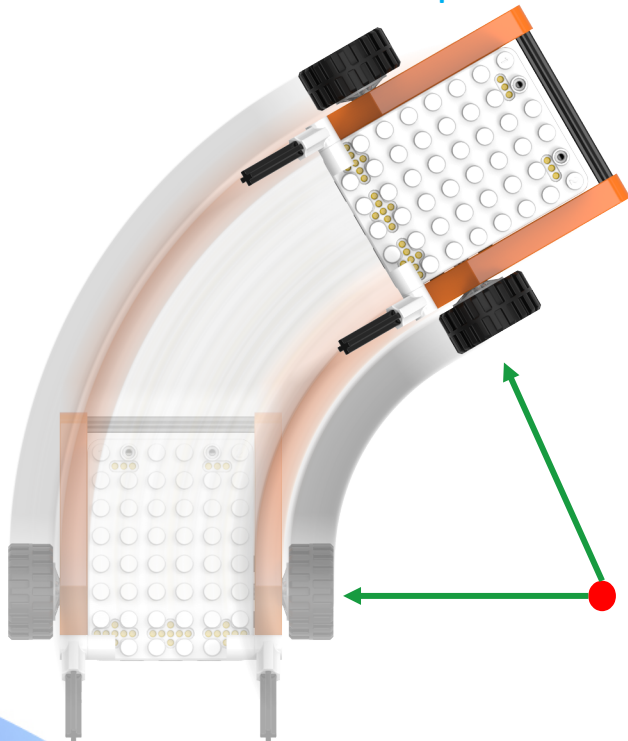


# Summary

2. The robot turns (one fast and one slow).



Set one motor to run fast and the other motor to run slow to make the robot turn around a point slightly outside the wheel.



When the turning speed is constant, you can adjust the amount of the turn by controlling the duration.



# Summary

## 3. Complete the competition

when clicked

set double built-in motors to keep running by speed at 1# (- 5 )% and 2# 5 %

wait 1 seconds

set double built-in motors to keep running by speed at 1# (- 0 )% and 2# 0 %

set double built-in motors to keep running by speed at 1# (- 5 )% and 2# 0 %

wait 1.65 seconds

set double built-in motors to keep running by speed at 1# (- 0 )% and 2# 0 %

set double built-in motors to keep running by speed at 1# (- 5 )% and 2# 5 %

wait 1 seconds

set double built-in motors to keep running by speed at 1# (- 0 )% and 2# 0 %



# SHARE WITH YOUR PARENTS

---

Share the knowledge about the obstacle racing with your mom and dad when you get home!

