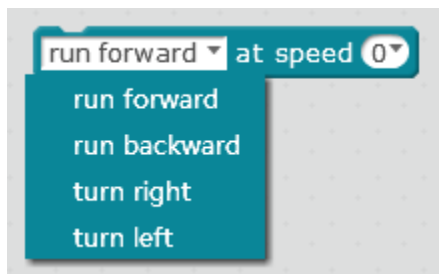


Move in a Square

NOTE: While following this tutorial and developing your own program, make sure that your robot is either propped up so the wheels aren't in contact with a surface, or in a space where it can move freely – Please make sure it doesn't fall off any desks or tables.

ANOTHER NOTE: This tutorial runs programs from the mBlock IDE. It does not update any programs onto the robot board.

Robots are made to move and in this tutorial, we are going to learn how. The block that controls robot movement has 2 drop-down menus. The first menu controls the direction the robot moves in:



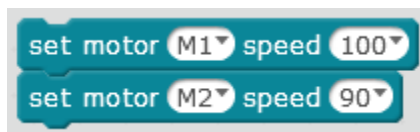
And the second menu controls power. Top power is 255, 0 stops the motors, and negative numbers reverse the direction. (NOTE: Slower power levels may not be strong enough to get the robot moving.)

And the second menu controls power. Top power is 255, 0 stops the motors, and negative numbers reverse the direction. (NOTE: Slower power levels may not be strong enough to get the robot moving.)

So to start, I am going to get my robot moving forward at a speed of 100:



It is possible that your robot does not run perfectly straight. This could be for a few reasons such as one of the wheels not being perfectly aligned or one of the wheels being more tightly attached to the robot causing more friction. If this is the case, you could fix this by replacing the run forward block with 2 blocks setting the motor power levels separately:



My robot is going straight though, so I am going to continue using the “run forward” block. I want to write a program that moves in a square. So let's go forward for one second, then turn right. I want to turn right until I have turned 90 degrees. I'm going to first try turning for one second and then see if that's too far or not far enough:

```
run forward at speed 100
wait 1 secs
turn right at speed 100
wait 1 secs
```

Can you see why this did not run as I wanted? If you can't see, why not try running this code yourself before reading on?

The robot runs the program and then finishes. The penultimate command is to turn right. Then the robot waits for one second. Then the program ends. *At no point in the program are the motors turned off.* So the motors keep running. I need to add a line to stop the motors:

```
run forward at speed 100
wait 1 secs
turn right at speed 100
wait 1 secs
run forward at speed 0
```

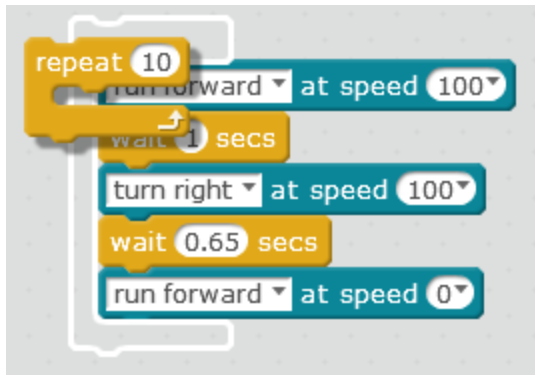
Ok, my robot moved a little too far. As I want to test how far to turn, and that is unrelated to the part of the code that moves forward, I am going to isolate the turning part of the code. This will make testing easier and faster:

```
run forward at speed 100
wait 1 secs
turn right at speed 100
wait 1 secs
run forward at speed 0
```

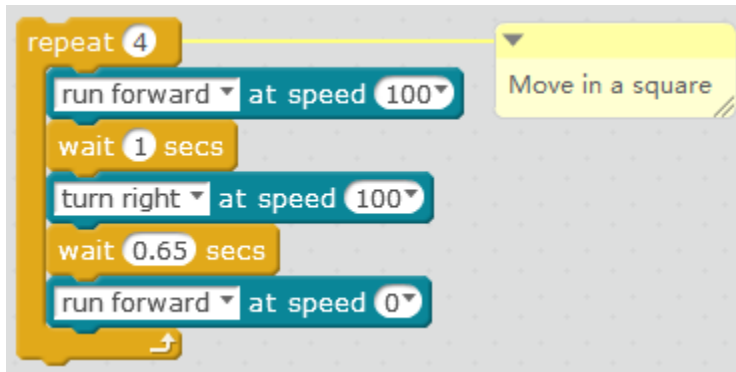
This code is isolated so is easier to test

After a few tests, I found that my robot needs 0.65 seconds to make a 90 degree turn. The time it needs to turn 90 degrees depends on many things. For example, what kind of battery you are using, how charged the battery is, how much friction there is between the motor and the wheel, and the speed setting of the turn will all affect the time needed to turn 90 degrees. So, when you do your tests, it is more than likely you will get a different number.

I need to go forward and turn 4 times to make 4 sides. There is a repeat block I can use to make a loop. Put all the things you want to be repeated inside the "repeat" block:



And change the number of repeats to 4:



Challenge:

1. Do I need the last movement command in my repeat loop? Can you make the code more streamlined – so it still does what we want, but with less commands issued?
2. Can you write a program that moves the robot in a rectangle?
3. Can you extend your program from question 2 to write a program that gives a warning sound before starting, and has lights on while moving?