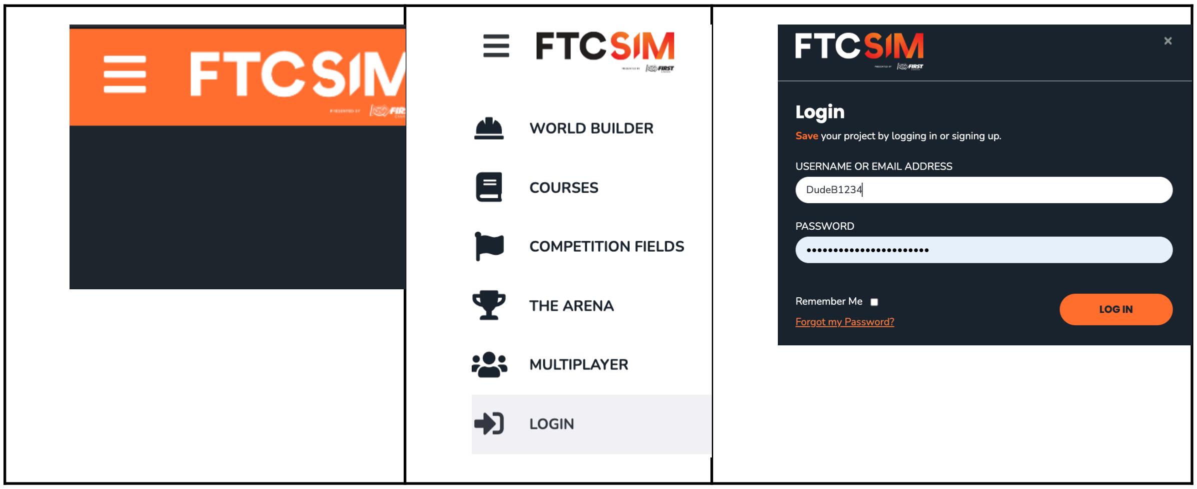
https://ftcsim.org/#frontpageCourses

Logging In

To log in, go to the “hamburger” menu (yes, that’s the actual name of that type of menu icon) at the top left of the web page. Then select “login”.



Login

Save your project by logging in or signing up.

USERNAME OR EMAIL ADDRESS: DudeB1234

PASSWORD: ••••••••••••••••••••••••

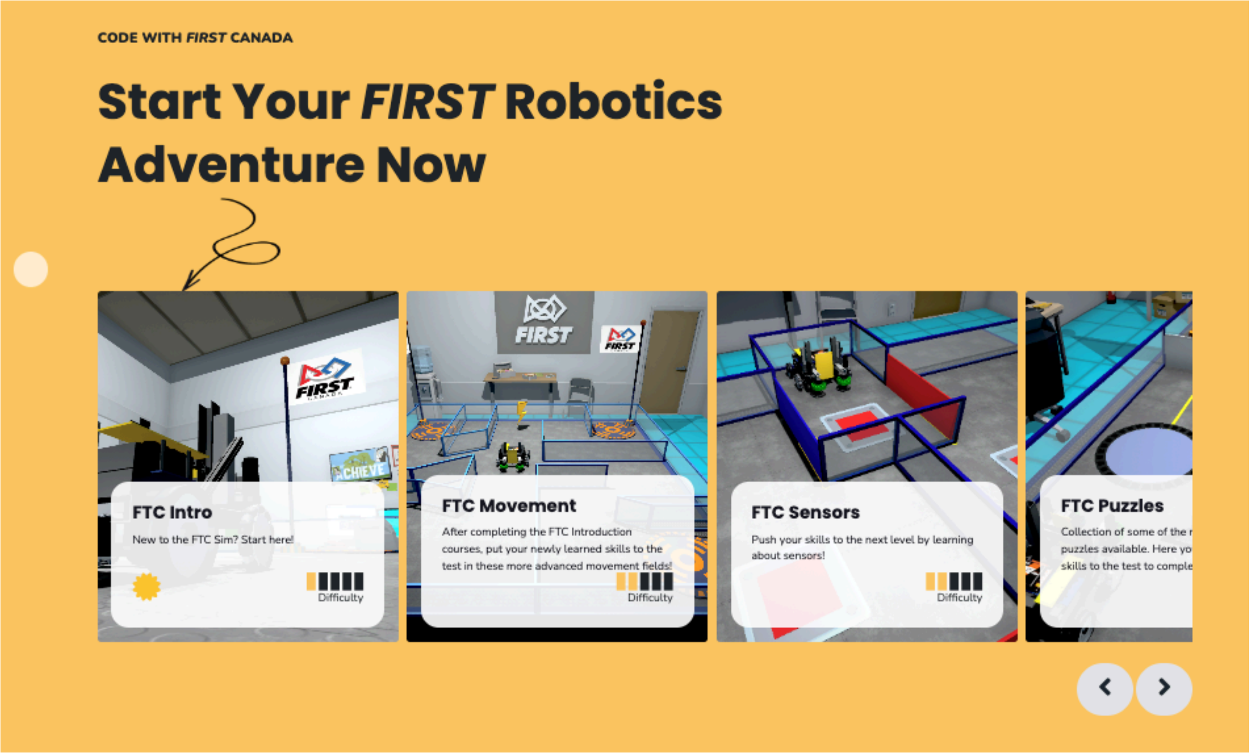
After you log in, you will see a “Profile” option in the menu which you can use to change your password if you so choose.

Start the lessons

Example shows how to navigate to the first lesson. Following lessons should follow the same pattern.

Go to https://ftcsim.org/#frontpageCourses. You can also navigate to it from the “hamburger” menu -> “Courses”.

Choose your category. For example “FTC Movement”



CODE WITH *FIRST* CANADA

Start Your *FIRST* Robotics Adventure Now

FTC Intro

New to the FTC Sim? Start here!

FTC Movement

After completing the FTC Introduction courses, put your newly learned skills to the test in these more advanced movement fields!

FTC Sensors

Push your skills to the next level by learning about sensors!

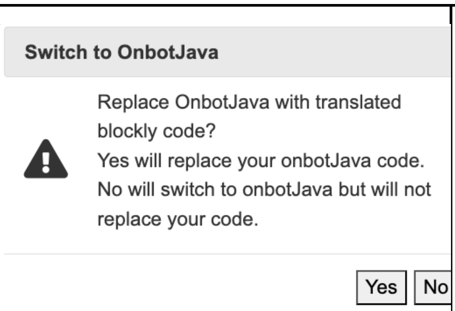
FTC Puzzles

Collection of some of the r puzzles available. Here yo skills to the test to comple

Go ahead and start with FTC Movement lesson 1. It will guide you through using Block programming to get the robot simulator working.

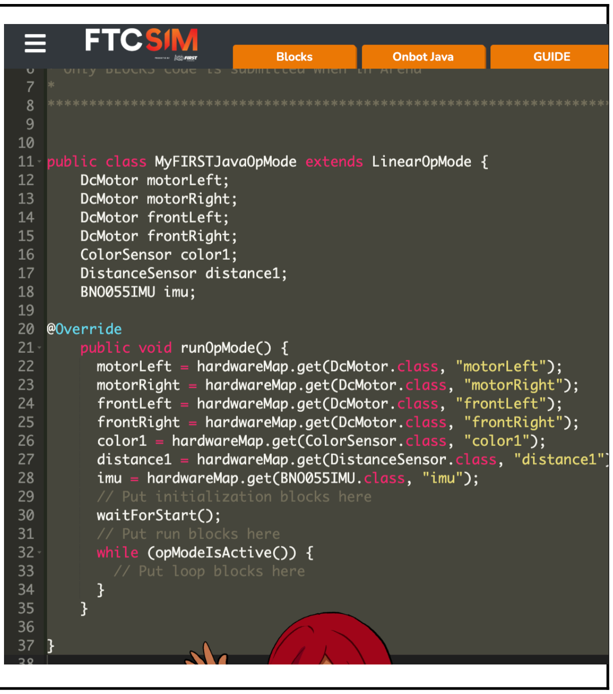
Hi! My name's Lisa!

After you update the Block code and everything seems to be working, click on the “OnBot Java” tab. This will pop up a prompt to ask you if you want to replace the OnBot Java code with the Blocky code. Click “yes”. Then take a look at what the Java code looks like.



Switch to OnbotJava

Replace OnbotJava with translated blockly code?
Yes will replace your onbotJava code.
No will switch to onbotJava but will not replace your code.



```
 6  * Only BLOCKS code is submitted when in Arena
 7  *
 8  ************************************************************************
 9
10
11 public class MyFIRSTJavaOpMode extends LinearOpMode {
12     DcMotor motorLeft;
13     DcMotor motorRight;
14     DcMotor frontLeft;
15     DcMotor frontRight;
16     ColorSensor color1;
17     DistanceSensor distance1;
18     BNO055IMU imu;
19
20 @Override
21     public void runOpMode() {
22       motorLeft = hardwareMap.get(DcMotor.class, "motorLeft");
23       motorRight = hardwareMap.get(DcMotor.class, "motorRight");
24       frontLeft = hardwareMap.get(DcMotor.class, "frontLeft");
25       frontRight = hardwareMap.get(DcMotor.class, "frontRight");
26       color1 = hardwareMap.get(ColorSensor.class, "color1");
27       distance1 = hardwareMap.get(DistanceSensor.class, "distance1"
28       imu = hardwareMap.get(BNO055IMU.class, "imu");
29       // Put initialization blocks here
30       waitForStart();
31       // Put run blocks here
32       while (opModeIsActive()) {
33         // Put loop blocks here
34       }
35     }
36
37 }
```

Feel free to experiment with the Java code. But keep in mind, the name of the class has to be ‘MyFIRSTJavaOpMode’. So as long as you don’t touch or change line 11 `***public class MyFIRSTJavaOpMode extends LinearOpMode {***`, you should not have any issues with this simulator.

When you are finished, continue on. Or keep working if you want to experiment or “perfect” your code solution.