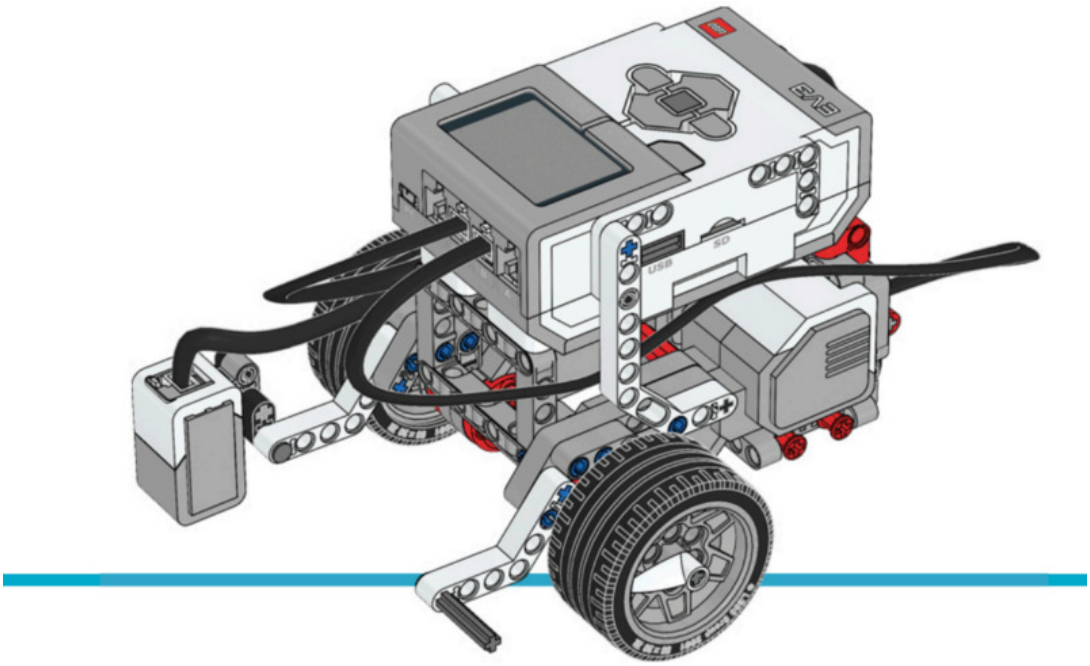
RoboRacers:

Algorithm Development

Will your algorithm be the fastest?

Welcome to RoboRacers! Today you will use a LEGO MINDSTORMS EV3 robot to speed along the racetrack! Let's get going...



Look out for these boxes:

Warning boxes have **very important** information.
Read these before moving on.

Hint boxes give tips to help you complete the tasks.
Read these if you get stuck.

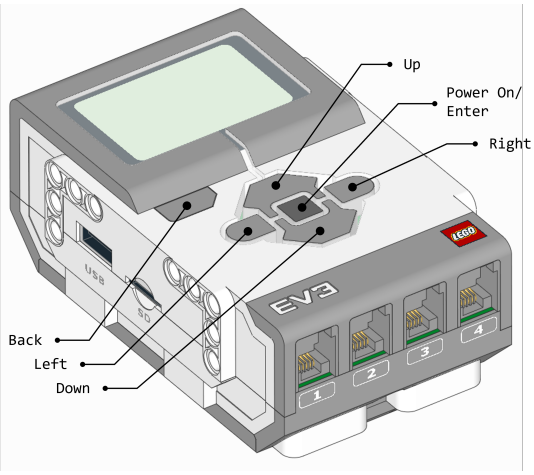
Think boxes give you something to think about and maybe an extra challenge.

This worksheet is for the RoboChallenge session *RoboRacers-Algorithms v2.0*.
Corresponding programs are available in the repository at `github.com/bennett-j/RoboChallenge`.

Getting Started

The Robot

If the robot has a green light on top then it is powered on and waiting for you! If you do not see a green light, ask your session leader for help. The `brick` in the picture below is the brains of the robot - it tells every other part of the robot what to do.

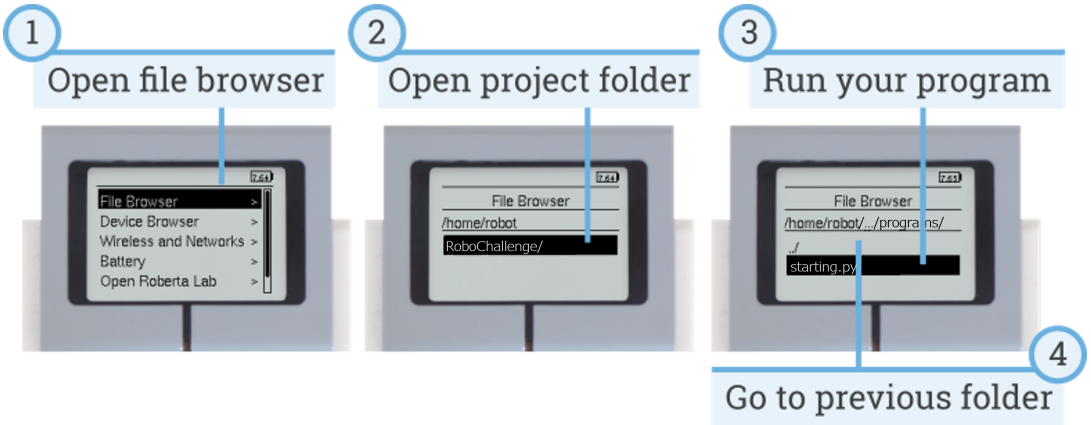


The `up`, `down`, `left`, and `right` buttons are not used to drive the robot. You will use them to select the robot's `programs` from the screen. Let's run a program to test the robot!

Running a program

Look at the small screen on the robot. It should look like the pictures below.

1. Use `up`, `down` and `enter` on the robot to open `File Browser`, then `RoboChallenge`.
2. Find the `starting.py` file and make sure it is highlighted in black.
3. Run the program by pressing the middle `enter` button. Listen closely!



What did the robot say? It should have said "Hello World!". This is a simple program to get you started. If the robot didn't speak or make any noise, ask your session leader for help.

To stop running a program, press the `back` button at the bottom left of the robot screen.

Editing a program

Programs don't write themselves. It's great that the robot said "Hello World", but that's not your name! So next, we'll change the program to greet you personally. You will write your code on the laptop using Visual Studio Code. This IDE (Integrated Development Environment) is used by many students and professionals to write code.

Look at the laptop in front of you. It should be logged in, with the file `starting.py` open. If you don't think this is the case, then ask your session leader for help. Follow the steps below to change `starting.py` to make the robot greet you.

Any line starting with `#` is a comment which the robot ignores. For example, `# This is a comment`. All other lines tell the robot to do something.

1. Find the line in the program which makes the robot say "Hello World!".
2. Change the program to say"Hello [your names]!" (for example, "Hello Jeremy!").

You will need to change the line which says
`ev3.speaker.say("Hello World!")`

Great, you've written your first program! Now, we need to send the program to the robot so it can run it.

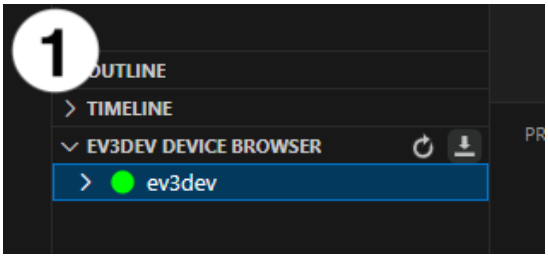
Downloading a program to the robot

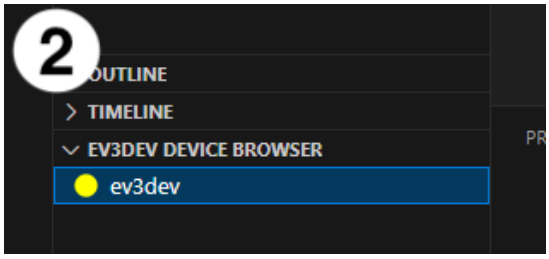
The first step is to make sure the robot is plugged in to the laptop via the USB cable. If you aren't sure where to plug the robot in, or if you don't have a cable, ask your session leader for help.

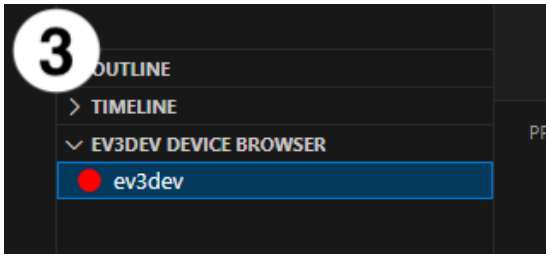
On your screen, you should see the `EV3DEV DEVICE BROWSER` at the bottom of the left panel. You may have to expand it by clicking on the small arrow next to it. If you don't see this, ask your session leader for help.

The next steps are very important! Make sure you follow them carefully. If you get stuck, ask your session leader for help.

Look for the small circle which is either green, yellow, or red. If it is green, like in picture 1 below, the robot is connected and ready to receive code. If it is yellow like in picture 2, the robot is currently in the process of connecting. If it is red, like in picture 3, the robot is not connected. If you are not connected, right click the robot's name, and then click `Reconnect`.

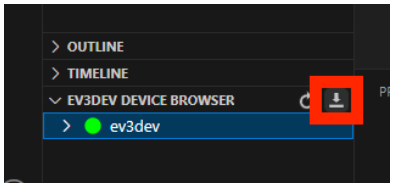




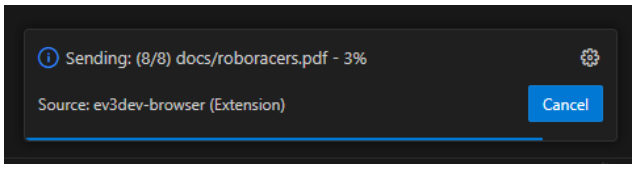


The circle must be green like in picture 1 before you can download your program to the robot. If you can't make it turn green, ask your session leader for help.

1. Hover the cursor to the right of where it says `EV3DEV DEVICE BROWSER` and click on the right hand button which looks like an arrow pointing downwards.



2. Watch in the bottom right hand side of the laptop screen whilst the files and programs are downloaded to the robot. You will see a blue loading bar fill from left to right, and a message saying `Download Complete` when it has finished.



3. Now, unplug the robot, and run the program again by following the steps in the `Running a Program` section above. Listen carefully to make sure the robot says your name!

Now that you have unplugged the robot, the little circle will be red when you plug it back in. This is normal, it just means the robot is not connected to the laptop. You will have to reconnect the robot to the laptop by right clicking on the robot's name and selecting `Reconnect`.

Getting the robot to move

Now we want to write programs to get the robot to move in different ways. On the laptop, open the file `moving.py` from the left panel. Your first tasks are to edit `moving.py` to make the robot complete the following movements. Read below for the commands you need to make the robot move.

1. Drive forwards 500 mm
2. Drive backwards 500 mm
3. Turn 180° clockwise
4. Turn 90° anti-clockwise
5. Drive in a square

To drive the robot in a straight line, we use:

```
robot.straight(distance)
```

We must replace the *word* `distance` with a *number*. Here, it is the distance in millimeters that we want to move.

To turn the robot on the spot, we use:

```
robot.turn(angle)
```

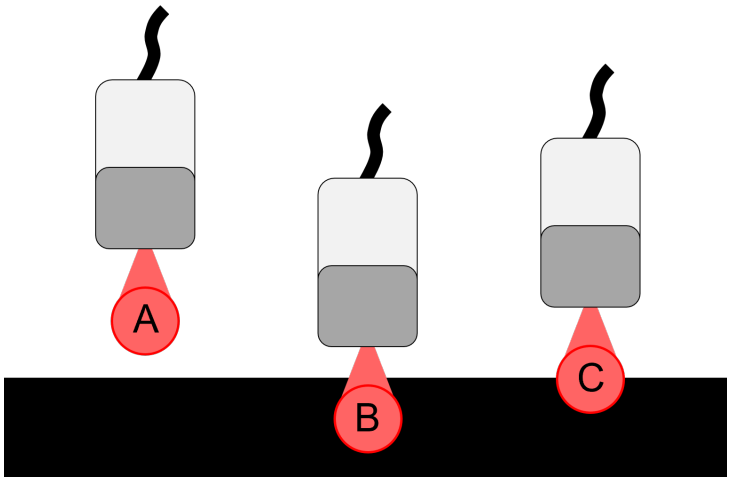
Here, you replace `angle` with the number of degrees you want the robot to turn.

Hint: Values can be negative.
Bonus hint: You can list commands one after another.

Challenge: Can you write a program to make the robot drive along the racetrack?

Sensing

Did the robot finish exactly where it started when you drove in a square? The robot has no way to "see" where it is. Just like you trying to walk in a big square whilst blindfolded. To let our robots "see", we use *sensors* on them. In this part, we'll use the reflectance sensor. It shines some light onto the floor beneath it and measures how much is reflected. Let's look at the diagram below with a black line and a white background...



In case **A**, the sensor is shining onto white and the reflectance value will be high (close to 100). However, in case **B**, the sensor is shining onto black and the reflectance value will be low (close to 0).

Think about what the sensor value might be in case **C** where the sensor is half on the black line and half on the white background.

1. Open the script `threshold-test.py`. You will see in the program we have added the sensor with the line

   ```
   line_sensor = ColorSensor(Port.S3)
   ```

   and that we measure the reflectance and show it on the robot's screen with

   ```
   ev3.screen.print(line_sensor.reflection())
   ```

2. Run the program and look at the reflectance values shown on the robot's screen.
3. Move the robot (with your hand) so the sensor is directly above the line and make a note of the reflectance value. Repeat to get a reflectance of the background surface.
4. Now open `sensing.py` and input the values you measured for `LINE_REF` and `BACKGROUND_REF`.

Now the robot can *sense* what is happening in the world, it needs to know how to react to what it *senses*. We put together a set of rules which makes a process we call an *algorithm*.

5. Design an algorithm so that your robot will drive forwards and stop when it reaches the line. Write your algorithm in the program `sensing.py` and test it.

Racing

Now it's time to get our robot to race the track by following the line!

1. Open `racing.py` and input the same values for `LINE_REF` and `BACKGROUND_REF`.
2. Design an algorithm and write the rest of the program so your robot will be the fastest around the racetrack!

Congratulations for getting your robot to race along the racetrack! Was your robot able to stay on the track? We hope you have learnt one way that robots can see the world and that you consider a future in robotics!