C - Secure the Planet!

One of your tasks on the mission is to secure the planet for mankind - learn how to speed up your programs to get our robot to patrol the planet surface.

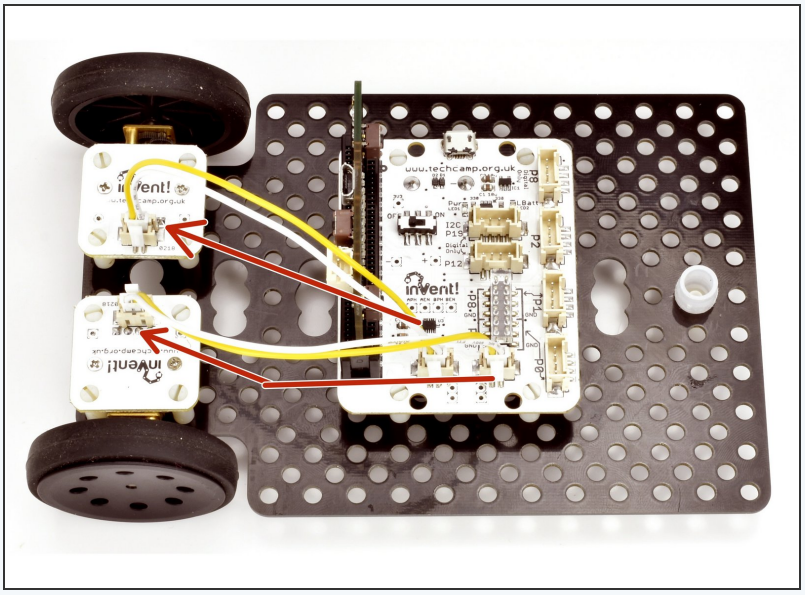
INTRODUCTION

One of your tasks on the mission is to secure the planet for mankind - learn how to speed up your programs to get our robot to patrol the planet surface.

Step 1

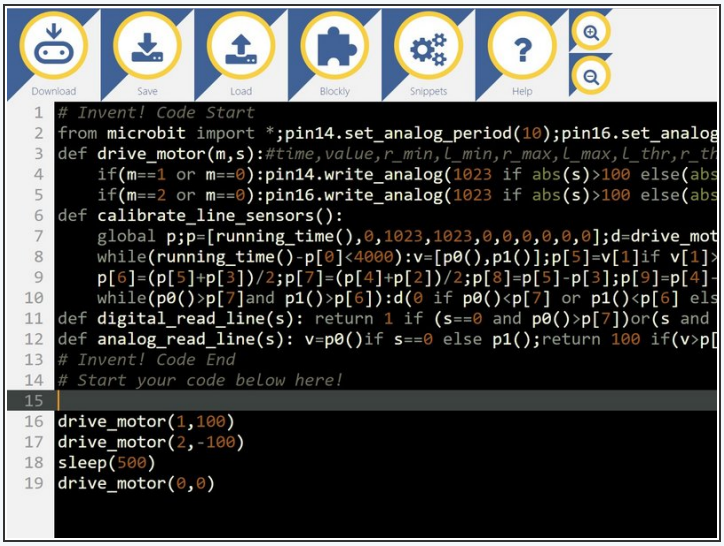
Robot Setup

- Make sure your robot is setup in the **same way** as the previous sections!



Step 2

Turning Accurately



- For the rest of this lesson, we need to be able to make the robot turn **accurately.**
- **Build** the simple test program in the picture - it should make your robot **spin** on the spot for **1 second**, and then stop.
- There are **1000 milliseconds in 1 second** - adjust the number until your robot turns by exactly **90 degrees.**

Step 3

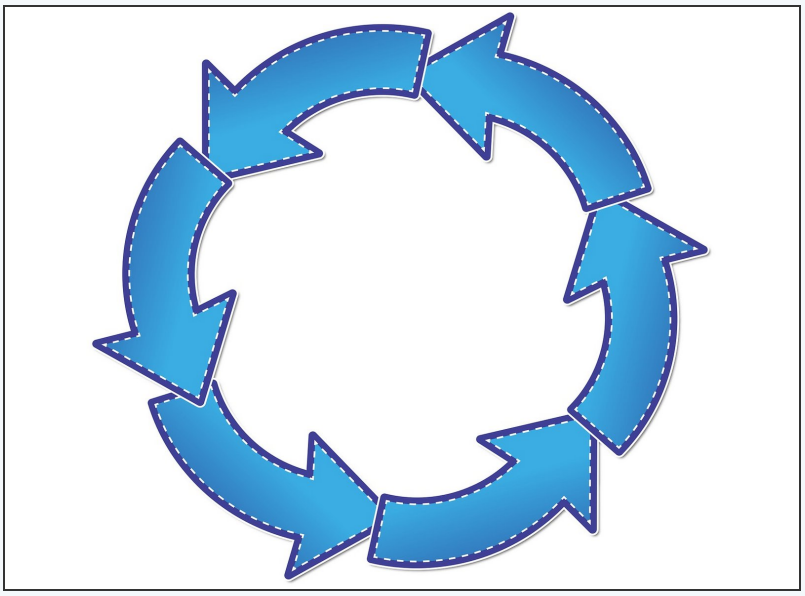
Driving in a Square


```
# Invent! Code End
# Start your code below here!

drive_motor(0,100)
sleep(1000)
drive_motor(1,100)
drive_motor(2,-100)
sleep(500)
drive_motor(0,100)
sleep(1000)
drive_motor(1,100)
drive_motor(2,-100)
sleep(500)
drive_motor(0,100)
sleep(1000)
drive_motor(1,100)
drive_motor(2,-100)
sleep(500)
drive_motor(0,100)
sleep(1000)
drive_motor(1,100)
drive_motor(2,-100)
sleep(500)
drive_motor(0,0)
```

- Now you can turn by 90 degrees, write a program that makes your robot **move in a square!**
- It should like something like the example in the picture but your **'delay'** times will be different.
- If you think about it, you only need to **reverse** 1 motor and then set it to **forwards** again to change direction - one motor can be going forward **all the time.**
- Also, you don't necessarily need to **stop** after turning - just **change the motor going in reverse to go forward again!**

Step 4

Using the Loop



- Making that last program took a while - and the robot was just doing the same things **over and over again.**
- Say we wanted to drive in a square 10 times - that would take **ages to program!**
- Driving in a square was doing the **same thing 4 times:**
  - Drive Forward
  - Delay
  - Turn
  - Delay
- We can use a **loop** to get the computer to repeat these steps for us!

Step 5

While Loops

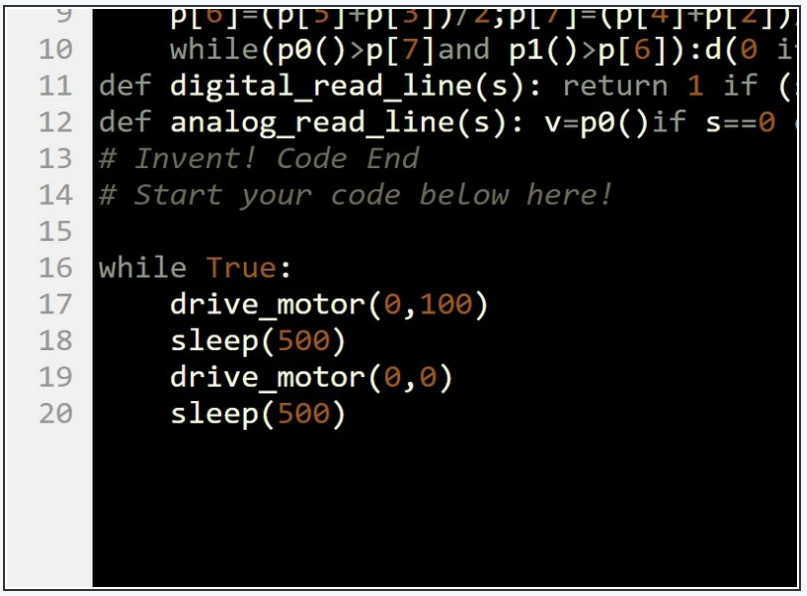
```
    while(running_time()-p[0]<4000):v=[p0(),p1()
    p[6]=(p[5]+p[3])/2;p[7]=(p[4]+p[2])/2;p[8]=p
    while(p0()>p[7]and p1()>p[6]):d(0 if p0()<p[
def digital_read_line(s): return 1 if (s==0 and
def analog_read_line(s): v=p0()if s==0 else p1()
# Invent! Code End
# Start your code below here!

while condition:
    # Run some code in here
    # (but only while the condition is true)
```

- We're going to be using a type of loop called a **while loop** to repeat the steps.
- A while loop works by **checking something** - this can be absolutely anything, for example if 1=1, if a switch is pressed, or if the program has been running for a certain length of time. This is called the **condition**.
- If the condition is **true**, any code **inside** the while loop is run **so long as the condition remains true.**
- Once the condition becomes **false**, the code inside the loop is no longer run, and the program moves to the next line.
- In Python, we always put a **colon (:)** after the condition, and any code inside the while loop is **indented** from the margin.

Step 6

Try it Out

- Let's write some code using a while loop to drive forwards, stop, drive forwards, stop, **forever**.
- To make sure the while loop runs forever, we just need to make the condition **always true!** The easiest way to do that is to simply use the line:
  - **while True:**
- Easy right? Don't forget the capital 'T' - true and false in Python are always **capitalised.**
- Inside the loop, we need some lines to **drive forwards, wait, stop, and wait** that will be repeated forever - add them in like the picture.



⚠ Don't forget to **indent** these lines of code (you can do it using the TAB key) - this means they are **inside the while loop**.

Step 7

Squares forever

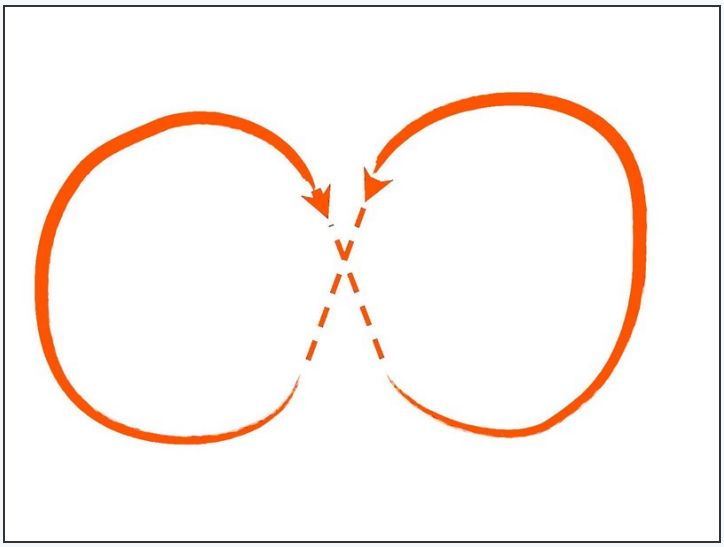

```
11 def digital_read_line(s): return 1 if (
12 def analog_read_line(s): v=p0()if s==0
13 # Invent! Code End
14 # Start your code below here!
15
16 drive_motor(0,100)
17 sleep(1000)
18 drive_motor(1,100)
19 drive_motor(2,-100)
20 sleep(500)
21
```

- Change your program so that you only have lines of code in the **loop,** that make the robot:
  - Drive **forwards**
  - Turn **90** degrees

Check the picture for a hint if you need it - the program in the picture will drive forwards, then turn 90 degrees, but it will only do it **once!**

Step 8

Figure of 8



- Now your robot can drive in a square using a loop, let's **change the code** so it will drive in a **figure of 8, forever.**
- Have a look at the picture if you don't know what a **figure of 8** is.
- Try to split the shape up into **2 sections,** and put it in a while loop to reduce the length of your program.

⚠ Make sure your robot drives in the figure of 8 **properly**, and ends up where it started !