



0.1 Your first program

What this lesson is about

The two lines at the top, drive, wait, stop, print, and the rule that stops the robot on its own.

- The two lines at the top
- Make it move
- It stops on its own
- The other directions

Questions

7 marks in all

1. What does `connect()` do?

[1 mark]

- ☐ A Gets you the robot, ready to take commands
- ☐ B Brings in the robot's commands
- ☐ C Starts the robot driving
- ☐ D Prints a line under the program

Answer: A. from bugbot import * brings in the commands, and `connect()` gets you the robot itself. You need both lines at the top.

2. What does `forward(50)` do on its own?

[1 mark]

- ☐ A Starts the robot at half speed and goes straight on to the next line
- ☐ B Drives 50 cm and then stops
- ☐ C Drives for 50 seconds
- ☐ D Waits until the robot has finished moving

Answer: A. `forward(50)` only starts the robot going. It does not wait, which is why you need `wait()` to let it drive for a while.

3. This program has no `stop()`. What does the robot do?

[1 mark]

```
from bugbot import *
connect()

forward(60)
wait(1.5)
```

- ☐ A Drives for about 1.5 seconds, then stops by itself once the program stops sending it commands
- ☐ B Drives for ever until you close the page
- ☐ C Never moves, because there is no `stop()`
- ☐ D Stops straight away, because the program is too short

Answer: A. The real robot stops on its own if it hears nothing from your program for half a second, and the simulator stops it the moment the program ends. Either way a crashed program cannot drive it into a wall.

4. Put these lines in order to drive forward for two seconds, stop, and then say so.

[1 mark]

Number the lines 1 to 6 to put them in the right order.

```
___ connect()
___ wait(2)
___ from bugbot import *
___ forward(50)
___ print("done")
___ stop()
```

Answer:

```
from bugbot import *
connect()
forward(50)
wait(2)
stop()
print("done")
```

The two setup lines always come first. Then start driving, let time pass, switch the motors off, and only then print that you have finished.

5. What does this program print?

[1 mark]

```
print("drove for", 2, "seconds")
```

Answer:

```
drove for 2 seconds
```

Commas in `print()` put a single space between each thing, whether it is text or a number.

6. Which of these make BugBot move across the mat?

[1 mark]

Tick every answer that is true.

- ☐ A `forward(60)`
- ☐ B `right(60)`
- ☐ C `led("green")`
- ☐ D `backward(60)`
- ☐ E `print(position())`

Answer: A, B, D. `forward`, `backward` and `right` drive the robot. `led` only changes the light, and `print` only writes a line.

7. You drive forward for 1 second and back for 1 second. Why does `position()` not say exactly (0, 0)?

[1 mark]

- ☐ A Driving by time is never exact, and the robot leans a little as it goes
- ☐ B There is a mistake in the program
- ☐ C `position()` counts from the middle of the mat
- ☐ D `backward()` always goes twice as far

Answer: A. Timing a drive can never be exact, and the vibrating feet push a little unevenly. Module 2 shows how the sensors fix that.

The task: cross the line

Drive from the start into the green zone and stop inside it, using `forward`, `wait` and `stop`. If it stops short, wait longer or drive faster.

```
from bugbot import *
connect()

forward(50)
wait(1)
stop()
```

The hint students can ask for: The green zone starts 40 cm ahead. Drive long enough, or fast enough, to get there, then stop inside it.

A solution

```
from bugbot import *
connect()
forward(60)
wait(4)
stop()
```

Any program that meets the task's checks is marked correct in the simulator; this is one way, not the only way.