



2.1 Where am I?

Sensing · Robot club · about 15 min

What this lesson is about

position(), heading(), velocity(), and resetting them.

- position()
- heading()
- velocity()
- Resetting
- Estimates, not truth

Questions

7 marks in all

1. What do the two numbers in `position()` mean? [1 mark]

- ☐ A x is centimetres to the right and y is centimetres forward, from where the program started
- ☐ B x is the speed and y is the heading
- ☐ C x is forward and y is to the right
- ☐ D They count from the middle of the mat

Answer: A. Both are in centimetres from where the program started, measured along the mat: x across it to the right, y up it. For a robot that starts facing up the mat, that is right and forward.

2. The robot starts at heading 0, runs `turn_right(30, angle=45)`, then `turn_left(30, angle=135)`. [1 mark]
What does `heading()` say? Give a whole number of degrees.

Answer: 270 (accept within 1). 45 take away 135 is -90, and headings go 0 to 360, so -90 shows as 270.

3. What is `velocity()` especially useful for? [1 mark]

- ☐ A Knowing whether the robot has actually stopped
- ☐ B Setting how fast the robot drives
- ☐ C Finding how far away the wall is
- ☐ D Resetting the position

Answer: A. It tells you how fast the robot is moving right now. After `stop()` the robot may still be sliding for a moment.

4. Roughly what does the last line print?

[1 mark]

```
forward(60, distance=20)
turn_right(30, angle=90)
reset_position()
reset_heading()
forward(60, distance=10)
print(position())
```

- ☐ A (0, 10)
- ☐ B (10, 20)
- ☐ C (0, 30)
- ☐ D (10, 0)

Answer: D. The resets made that spot (0, 0) and the direction it faced heading 0, but position() still measures along the mat. After the right turn, forward is across the mat, so the 10 cm shows up in x: it prints (10.1, -0.0).

5. Which sensors are inside the robot's 9-axis inertial sensor?

[1 mark]

Tick every answer that is true.

- ☐ A Gyroscope
- ☐ B Accelerometer
- ☐ C Magnetometer
- ☐ D Optical-flow sensor
- ☐ E Time-of-flight sensor

Answer: A, B, C. The compass combines a gyroscope, accelerometer and magnetometer. The optical-flow sensor underneath measures position instead.

6. What does this program print?

[1 mark]

```
import math
x, y = 3, 4
print("from home:", math.sqrt(x*x + y*y), "cm")
```

Answer:

from home: 5.0 cm

Straight-line distance from home is the square root of x times x plus y times y. 9 plus 16 is 25, and its square root is 5.

7. After driving a square, the robot is not quite back where it started, and a few degrees off. What is the right way to think about that?

[1 mark]

- ☐ A Every move is a little off, and small errors add up
- ☐ B The optical-flow sensor is broken
- ☐ C position() only works in straight lines
- ☐ D The robot really is exactly home

Answer: A. Each drive and turn is nearly right, and on the real robot each reading of where it is is nearly right too. Real robots correct it by looking at landmarks, which the camera does in Module 4.

The task: report from the corner

Get to the point 30 cm right and 30 cm forward of where you start, and print a line in the shape `at (30.1, 29.8) facing 1.2` from `position()` and `heading()`.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()

# drive forward at 60 for 30 cm, then stop
forward(60, distance=30)
```

The hint students can ask for: Get to the point 30 cm right and 30 cm up from where you start (forward then right, or right then forward), and print your position and heading in the form `at (30.1, 29.8) facing 1.2`

A solution

```
from bugbot import *
connect()
forward(60, distance=30)
right(60, distance=30)
x, y = position()
print(f'at ({x}, {y}) facing {heading()}')
```

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