



2.1 Where am I?

Sensing · Robot club · about 15 min

Name _____

Class _____

Date _____

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
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.

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
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)

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

6. What does this program print?

[1 mark]

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

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

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)
```

Plan your program here, then type it in and press Run.



Do it on the robot

www.bugbotlab.com/learn/2-1-where-am-i/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Print position() and velocity() every 0.5 s while driving.
2. Drive a triangle and print how far from home you end up: $x^2 + y^2$ under a square root.
3. Reset the heading after each turn of a square. Does the final error change?