ANSWERS

9.1 Sending

Talking to each other · Robot club · about 15 min

What this lesson is about

send(): a broadcast everyone hears; numbers into text.

- The radio
- The radio log
- Messages are text

Questions

6 marks in all

1. You call `send("hello")`. Who hears it? [1 mark]

- ☐ **A** Every other robot on the mat
- ☐ **B** Only the nearest robot
- ☐ **C** Only a robot you name
- ☐ **D** Nobody until they reply

> **Answer:** A. There is no address. A message is a shout, not a phone call.

2. Which line sends your position in a way another robot can read? [1 mark]

- ☐ **A** `send(f"at {x:.0f},{y:.0f}")`
- ☐ **B** `send("at x,y")`
- ☐ **C** `send(position())`
- ☐ **D** `send(x, y)`

> **Answer:** A. A message is text, so the numbers must be put into the string. `"at x,y"` sends the letters x and y, not the numbers.

3. What does this program print? [1 mark]

```
x, y = 12.6, 30.4
print(f"at {x:.0f},{y:.0f}")
```

> **Answer:**
>
> ```
> at 13,30
> ```
>
> `:.0f` shows no decimal places, rounding to the nearest whole number: 12.6 becomes 13 and 30.4 becomes 30.

4. Two robots are not understanding each other. Where do you look first? [1 mark]

- ☐ **A** The radio log under the robot view, which shows every message with its time and sender
- ☐ **B** The depth grid
- ☐ **C** The robot's position
- ☐ **D** The gripper

> **Answer:** A. The log shows yours as well as theirs, so you can see exactly what was said and when.

5. You send a question. Is a reply guaranteed? [1 mark]

- ☐ **A** No: the radio is a shout, and nobody has to answer
- ☐ **B** Yes, every robot must reply
- ☐ **C** Yes, but only from the nearest robot
- ☐ **D** Only if you send it twice

> **Answer:** A. A program must cope with no answer. Lesson 9.3 adds a timeout for exactly this.

6. What type of Python value is every radio message? [1 mark]

> **Answer:** string. Messages are text. Numbers have to be turned into text before sending and back into numbers when received.

The task: call out

Send `hello`, drive forward 20 cm, then send your position as `at <x>,<y>` in whole centimetres from the start.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()
# radio this to every other robot
send('hello')
# pause 1 s (the robot keeps doing what it was told)
wait(1)
print(messages())
```

The hint students can ask for: Send `hello`, drive forward 20 cm, then send your position as `at <x>,<y>` (whole centimetres, from the start).

A solution

```
from bugbot import *
connect()
send("hello")
forward(50, distance=20)
x, y = position()
send(f"at {x:.0f},{y:.0f}")
wait(0.5)
print(messages())
```

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