



## 9.1 Sending

Talking to each other · Robot club · about 15 min

Name \_\_\_\_\_

Class \_\_\_\_\_

Date \_\_\_\_\_

### 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

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)`

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

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

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

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

6. What type of Python value is every radio message?

[1 mark]

---

---

## 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())
```

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

---

---

---

---

---

---

---



**Do it on the robot**

[www.bugbotlab.com/learn/9-1-sending/](http://www.bugbotlab.com/learn/9-1-sending/)

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

## Challenges

1. Send your heading as `facing <degrees>`.
2. Send `tick <seconds>` once a second for five seconds, using `clock()`.
3. Send your position as JSON ( `json.dumps({"x": x, "y": y})` ), the way lesson 8.7 saved a model.