



9.2 Listening

Talking to each other · Robot club · about 15 min

What this lesson is about

`messages()`: what arrived since you last asked, and who sent it.

- They queue up
- Ask often
- Whose message?

Questions

6 marks in all

1. What does `messages()` give you? [1 mark]

- ☐ A Everything that arrived since you last asked, oldest first, as (sender, text) pairs
- ☐ B Only the newest message
- ☐ C Every message ever sent in the run, including yours
- ☐ D The text of the next message, waiting until one comes

Answer: A. Each call empties the queue, and each item is a pair you can unpack as `sender`, `text`.

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

```
inbox = [("Beacon", "tick 1"), ("Scout", "at 25,55"), ("Beacon", "tick 2")]
for sender, text in inbox:
    if sender == "Beacon":
        print("heard", sender + ":", text)
```

Answer:

```
heard Beacon: tick 1
heard Beacon: tick 2
```

Unpacking each pair gives the sender and the text. The Scout's message is skipped because it is not from the Beacon.

3. The Beacon sends a message once a second, the first at 0.1 s. Your program waits 3.2 s and then calls `messages()`. How many messages does it get? [1 mark]

Answer: 4. Ticks at 0.1, 1.1, 2.1 and 3.1 seconds have arrived. The radio kept them until you asked.

4. A polling loop checks `messages()` but has no `wait`. What is the problem? [1 mark]

- ☐ A It asks thousands of times a second and the robot has no time for anything else
- ☐ B It misses every message
- ☐ C It sends a message each time round
- ☐ D Nothing, it is the fastest way to listen

Answer: A. Check, act, wait. A short `wait(0.1)` still hears each message within a tenth of a second.

5. Your program is busy driving for 5 seconds and does not call `messages()` . What happens to the Beacon's messages? [1 mark]
- ☐ A They queue up and are all there next time you ask
 - ☐ B They are lost
 - ☐ C Only the last one is kept
 - ☐ D The Beacon stops sending

Answer: A. The radio does not lose what you do not read straight away.

6. You call `send("hello")` and then `messages()` . Is your own `hello` in the list? [1 mark]
- ☐ A No: `send` goes to every other robot, so you only get what others sent
 - ☐ B Yes, everyone hears everything, you included
 - ☐ C Yes, as the first item
 - ☐ D Only if nobody else replied

Answer: A. Reading your own message as if someone else said it is a classic bug. The radio log shows yours; `messages()` does not.

The task: listen

Print `heard <sender>: <text>` for each of the Beacon's messages, at least five of them. Do not drive.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()
# pause 2 s (the robot keeps doing what it was told)
wait(2)
print(messages())
```

The hint students can ask for: The Beacon sends `tick <n>` every second. Keep asking `messages()` and print `heard <sender>: <text>` for each one. Do not drive.

A solution

```
from bugbot import *
connect()
for tick in range(70):
    for sender, text in messages():
        print(f"heard {sender}: {text}")
    wait(0.1)
```

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