



9.5 Taking turns

Talking to each other · Robot club · about 20 min

What this lesson is about

One robot at a time through a corridor: announce, wait, go.

- Listen to the rhythm
- Wait, announce, go
- Straight down the middle

Questions

6 marks in all

1. In the corridor task, which message tells you it is time to go? [1 mark]

- ☐ A corridor clear
- ☐ B corridor busy
- ☐ C entering
- ☐ D out

Answer: A. The Porter says busy as it enters and clear as it leaves. Wait for clear.

2. The corridor stays clear for about 9 seconds, and you need about 6 seconds to get through. After hearing clear, how many seconds can you afford to delay before setting off? [1 mark]

Answer: 3 (accept within 0.5). $9 - 6 = 3$ seconds, which is why you go straight away.

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

```
batches = [("Porter", "corridor busy"), [], [("Porter", "corridor clear")]]
clear = False
checks = 0
for batch in batches:
    checks += 1
    for sender, text in batch:
        if "clear" in text:
            clear = True
    if clear:
        break
print("clear" if clear else "waiting", "after", checks, "checks")
```

Answer:

clear after 3 checks

Each batch is one call to `messages()`. The first has busy, the second nothing, and the third has clear, so it stops after 3 checks.

4. Put the steps for getting through the corridor in order.

[1 mark]

Number the lines 1 to 4 to put them in the right order.

- ___ Drive through, holding the middle
- ___ Wait for corridor clear
- ___ send("entering")
- ___ send("out")

Answer:

```
Wait for corridor clear
send("entering")
Drive through, holding the middle
send("out")
```

Wait, announce, go, and say when you are through.

5. The Porter does not listen to messages. Why send `entering` anyway?

[1 mark]

- ☐ A A protocol is the agreement: a robot that did listen could hold back for you
- ☐ B The Porter stops if it hears it
- ☐ C The checker ignores robots that do not send
- ☐ D `forward` only works after a send

Answer: A. The agreement is what matters, not the hardware at the other end.

6. Driving along the corridor, the robot faces heading 90. Why does it steer from `y` and not `x`?

[1 mark]

- ☐ A Facing along x, sideways drift shows up in y
- ☐ B x is always zero
- ☐ C `position()` only gives y
- ☐ D Heading 90 means facing up the mat

Answer: A. Heading is clockwise from up the mat, so 90 faces along x, and y is how far off the centre line it is.

The task: one at a time

Wait for `corridor clear`, send `entering`, drive through to the green zone without touching the walls or the Porter.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()
# radio this to every other robot
send('entering')
# drive forward at 60 for 72 cm, then stop
forward(60, distance=72)
# radio this to every other robot
send('out')
```

The hint students can ask for: The corridor fits one robot. The Porter says `corridor busy` as it goes in and `corridor clear` as it comes out. Wait for clear, send `entering`, drive through to the green zone (72 cm ahead), and do not touch anything.

A solution

```
from bugbot import *
connect()
def wrapped(h):
    return (h + 180) % 360 - 180

clear = False
while not clear:
    for sender, text in messages():
        print(sender, "says:", text)
        if "clear" in text:
            clear = True
    wait(0.1)
send("entering")
while position()[0] < 72:
    x, y = position()
    drive(60, y * 6, wrapped(90 - heading()) * 3)    # straight down the middle of the
corridor
    wait(0.1)
stop()
send("out")
```

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