



3.6 Project: precision parking

What this lesson is about

Round a box and into a bay facing east, on the worst robot in the course.

- The plan
- Leg one as a function
- Leg two: sideways, holding two things
- Leg three: turn and hold
- Where next

Questions

6 marks in all

1. In `hold_line(0, 67)`, what does `x = 0` mean? [1 mark]

- ☐ A The line the robot started on
- ☐ B The left edge of the mat
- ☐ C The middle of the box
- ☐ D The parking bay

Answer: A. `position()` counts from the start, so 0 is the starting line, which passes to the left of the box.

2. In `slide_right`, which part of the `drive` is the speed? [1 mark]

```
drive((target_y - y) * 10, 70, wrapped(0 - heading()) * 4)
```

- ☐ A The sideways part, 70
- ☐ B The forward part, holding y
- ☐ C The rotation part, holding the heading
- ☐ D There is no speed, it uses distance

Answer: A. The robot is sliding right, so sideways does the travelling. Forward now holds y and rotation holds the heading.

3. `hold_heading(90)` is running and the robot faces 60. What rotation does `max(-60, min(60, error * 3))` ask for? [1 mark]

Answer: 60. The error is 30 and 30 times 3 is 90, but the clamp caps it at 60.

4. What does this program print?

[1 mark]

```
def wrapped(h):
    return (h + 180) % 360 - 180

target = 90
for heading in [350, 80, 120]:
    error = wrapped(target - heading)
    print(heading, max(-60, min(60, error * 3)))
```

Answer:

```
350 60
80 30
120 -60
```

From 350 the shortest way to 90 is 100 degrees clockwise, which clamps to 60. From 80 it is 10, so 30. From 120 it is 30 the other way, which clamps to -60.

5. Put the parking program's main lines in order, after the functions have been defined.

[1 mark]

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

```
___ hold_line(0, 67)
___ print("facing", round(heading(), 1))
___ hold_heading(90)
___ slide_right(67, 55)
```

Answer:

```
hold_line(0, 67)
slide_right(67, 55)
hold_heading(90)
print("facing", round(heading(), 1))
```

Up past the box, across the top to the bay, then turn to face east and report.

6. Why does `hold_heading` loop `while abs(error) > 2` instead of until the error is exactly 0?

[1 mark]

- ☐ A The heading is never exactly on target, so it would never stop
- ☐ B `abs()` cannot give 0
- ☐ C An error of 2 is the same as 0
- ☐ D Python cannot compare with 0

Answer: A. Real sensors wobble by fractions of a degree. A small allowed error lets the loop finish.

The task: precision parking

Reach the bay without touching the box and finish facing 90, within 6 degrees.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()

def wrapped(h):
    return (h + 180) % 360 - 180

def hold_line(target_x, until_y):
    while position()[1] < until_y:
        # where am I? (cm from where I started)
        x, y = position()
        # forward, sideways, rotation: -100 to 100 each, until the next command
        drive(70, (target_x - x) * 10, wrapped(0 - heading()) * 4)
        # pause 0.1 s (the robot keeps doing what it was told)
        wait(0.1)
    # all motors off
    stop()

hold_line(0, 67)
```

The hint students can ask for: Get to the bay at (77, 77) without touching the box, and finish facing 90 (east). The robot curves and leaks: hold your lines with feedback, then turn and hold the final heading.

A solution

```
from bugbot import *
connect()
def wrapped(h):
    return (h + 180) % 360 - 180
def hold_line(x_target, until_y):
    while position()[1] < until_y:
        x, y = position()
        drive(70, (x_target - x) * 10, -wrapped(heading()) * 4)
        wait(0.1)
    stop()

def hold_heading(target):
    error = wrapped(heading()) - target
    while abs(error) > 2:
        drive(0, 0, -max(-60, min(60, error * 3)))
        wait(0.05)
        error = wrapped(heading()) - target
    stop()

hold_line(0, 67)          # up the left of the box (x is relative: 0 means x = 20 on the
                           mat)
while position()[0] < 55: # strafe right along the top of the box, holding y and heading
    x, y = position()
    drive((67 - y) * 10, 70, -wrapped(heading()) * 4)
    wait(0.1)
stop()
hold_heading(90)
print('parked at', position(), 'facing', round(heading()))
```

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