



1.3 Turning and heading

What this lesson is about

`turn_left()` and `turn_right()` by an angle, `heading()` as a compass, and turning on the spot.

- `turn_left` and `turn_right`
- Which way is which
- A square, with turns
- Turning without stopping
- Other shapes

Questions

7 marks in all

1. Which way does `turn_right(30, angle=90)` turn the robot, seen from above? [1 mark]
- ☐ A Clockwise, the way a clock's hands go
 - ☐ B Anticlockwise
 - ☐ C It slides right without turning
 - ☐ D It depends on which way the robot is facing

Answer: A. `turn_right` is always clockwise and `turn_left` always anticlockwise, whichever way the robot faces.

2. The robot starts at heading 0 and runs `turn_left(30, angle=90)`. What does `heading()` say? Give a whole number of degrees. [1 mark]

Answer: 270 (accept within 1). Headings go 0 to 360 clockwise. A quarter turn anticlockwise from 0 is the same direction as 270.

3. From heading 0 the robot turns right 90, right 90 again, then left 45. What does `heading()` say? Give a whole number of degrees. [1 mark]

Answer: 135 (accept within 1). 90 plus 90 is 180, and turning left takes 45 off, leaving 135.

4. What does `turn_right(40)` do, with no angle? [1 mark]
- ☐ A Starts turning clockwise and keeps turning until the next command
 - ☐ B Turns 40 degrees and stops
 - ☐ C Turns a quarter turn at speed 40
 - ☐ D Nothing, because it needs an angle

Answer: A. Without an angle it is like `forward(speed)`: it starts and returns at once. Use it when you want to turn until something happens.

5. A pentagon has five equal sides. What angle should each `turn_right` be? Give a whole number of degrees. [1 mark]

Answer: 72. Any regular shape turns 360 divided by the number of sides at each corner, and 360 divided by 5 is 72.

6. Put these lines in order to drive a square: each time, drive a side, then turn.

[1 mark]

Number the lines 1 to 6 to put them in the right order.

```
___ for side in range(4):
___ turn_right(30, angle=90)
___ from bugbot import *
___ connect()
___ forward(60, distance=25)
___ print("home?", position())
```

Answer:

```
from bugbot import *
connect()
for side in range(4):
    forward(60, distance=25)
    turn_right(30, angle=90)
print("home?", position())
```

The drive and the turn are both indented, so both repeat four times. The print is not indented, so it runs once at the end.

7. After driving a square, the robot is a little away from its start and a few degrees off. Why?

[1 mark]

- ☐ A The small errors in each move add up
- ☐ B range(4) only does three sides
- ☐ C turn_right cannot do exactly 90
- ☐ D position() is reset at each corner

Answer: A. Each drive and each turn is nearly right. Eight nearly right moves in a row can end up noticeably off.

The task: face east

Turn so the robot faces 90 degrees (to the right of where it started), then print `facing:` and its heading. Do not drive anywhere.

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

# turn 45 degrees clockwise, then stop
turn_right(30, angle=45)
print("facing:", heading())
```

The hint students can ask for: Turn so the robot faces 90 degrees (east, to the right), then print `facing:` and its heading. Do not drive anywhere.

A solution

```
from bugbot import *
connect()
turn_right(30, angle=90)
print('facing:', heading())
```

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