



1.3 Turning and heading

Driving · Robot club · about 15 min

Name _____

Class _____

Date _____

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

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

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

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())
```

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

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())
```

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



Do it on the robot

www.bugbotlab.com/learn/1-3-turning/

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

Challenges

1. Draw a triangle, then a pentagon, with the same loop and a different `sides`.
2. Turn 360 in four steps of 90 and print the heading after each. How far off is the last one?

3. Use `turn_left(30)` with no angle and a `while` loop to stop when `heading()` is close to 270.