



1.5 Speed and precision

What this lesson is about

Coasting, overshoot, and why fast and accurate pull in different directions.

- Coasting
- Why: momentum
- The robot allows for it
- The trick: fast, then slow
- Turning has the same problem
- Measure it yourself

Questions

7 marks in all

1. The robot drives by time at speeds 30, 60 and 100, then `stop()`. Which slides furthest after the motors switch off? [1 mark]
- ☐ A Speed 100
 - ☐ B Speed 30
 - ☐ C Speed 60
 - ☐ D They all slide the same

Answer: A. The faster it is going, the more momentum it has, so the mat's friction takes longer to stop it.

2. How does `distance=` allow for coasting? [1 mark]
- ☐ A It switches the motors off early, by the amount it expects the robot to slide
 - ☐ B It drives backwards a little after stopping
 - ☐ C It always drives at a slow speed
 - ☐ D It waits for the robot to hit the target and then brakes

Answer: A. The robot knows how fast it is going and predicts the slide. The prediction is never perfect, so fast drives are still a little further off.

3. Why drive `forward(100, distance=15)` and then `forward(25, distance=5)`, instead of 20 cm at 100? [1 mark]
- ☐ A Most of the way is at full speed, and the finish has the slow speed's accuracy
 - ☐ B Speed 100 cannot go further than 15 cm
 - ☐ C It makes the robot go perfectly straight
 - ☐ D It is the only way to drive 20 cm

Answer: A. Most of the way is fast, and only the final part, where the stopping point is decided, is slow. Real robots do the same.

4. You want to stop 50 cm ahead, with the last 8 cm at a slow speed. What distance should the fast `forward` be? Give a whole number of centimetres. [1 mark]

Answer: 42. The two legs add up to the whole journey, so the fast part is 50 minus 8, which is 42 cm.

5. What does this program print?

[1 mark]

```
errors = []
for y in [20.1, 20.3, 19.6]:
    errors.append(round(y - 20, 1))
print("errors:", errors)
```

Answer:

errors: [0.1, 0.3, -0.4]

Each error is where the robot got to minus 20. A negative error, like -0.4, means it stopped short.

6. What does this program print?

[1 mark]

```
speed = 60
y = 19.8
print(f"speed {speed}: asked 20, got {y}")
```

Answer:

speed 60: asked 20, got 19.8

In an f-string the names inside curly brackets are replaced by their values.

7. A fast turn overshoots by about a degree. Why does that matter?

[1 mark]

- ☐ **A** Over a long route those degrees add up
- ☐ **B** It does not matter, a degree is too small to notice
- ☐ **C** The robot will refuse to turn again
- ☐ **D** It makes the next forward go backwards

Answer: A. One degree is tiny, but every turn adds its own. A slower turn overshoots by only a fraction of a degree.

The task: gentle stop

Stop with the robot's centre within 2 cm of the small square 50 cm ahead, having driven at least 40 cm. The starter times the drive; timing is never that accurate.

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

# drive forward at 100 (keeps going until the next command)
forward(100)
# pause 2.5 s (the robot keeps doing what it was told)
wait(2.5)
# all motors off
stop()
```

The hint students can ask for: Stop with the robot's centre within 2 cm of the small square 50 cm ahead. Timing a drive is never that accurate: measure it.

A solution

```
from bugbot import *
connect()
forward(100, distance=44)
forward(25, distance=6)
```

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