



1.5 Speed and precision

Driving · Robot club · about 15 min

Name _____

Class _____

Date _____

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
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
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
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]

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)
```

6. What does this program print?

[1 mark]

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

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

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()
```

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



Do it on the robot

www.bugbotlab.com/learn/1-5-speed-and-precision/

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

Challenges

1. Get the gentle stop within 1 cm.
2. Write `precise_forward(cm)` that does the fast-then-slow trick for any distance.
3. Do the same for turns: `precise_turn(deg)`.

