



F4.3 Local and global variables

Functions and structured code · GCSE · OCR J277 2.2.3, AQA 8525 3.2.10,
Edexcel 1CP2 6.6.3 · about 15 min

Name _____

Class _____

Date _____

What this lesson is about

Scope: where a variable exists, and why passing values beats global.

- Local variables stay inside
- Same name, different variables
- Reading a global works; changing it does not
- Fix 1: pass it in and return it
- Fix 2: the global keyword

Questions

6 marks in all

1. A function sets `gap = 10` inside its body. After calling it, the program runs `print(gap)`. What happens? [1 mark]

- ☐ A NameError: gap only exists while the function runs
- ☐ B It prints 10
- ☐ C It prints gap
- ☐ D It prints nothing

2. What does this program print? [1 mark]

```
speed = 80

def gentle():
    speed = 30
    print(speed)

gentle()
print(speed)
```

3. `beeps = 0` outside, and inside a function `beeps = beeps + 1` with no `global` line. What happens when the function runs? [1 mark]

- ☐ A UnboundLocalError: assigning makes beeps local, and it has no value yet
- ☐ B beeps becomes 1
- ☐ C The global beeps becomes 1
- ☐ D NameError

4. Which are reasons local variables are good practice?

[1 mark]

Tick every answer that is true.

- ☐ A A subprogram can be understood and tested on its own
- ☐ B The same names can be used safely in different subprograms
- ☐ C The memory is freed when the subprogram ends
- ☐ D They make the program run without errors

5. What does this program print?

[1 mark]

```
count = 0

def add():
    global count
    count = count + 1

add()
add()
print(count)
```

6. What is the scope of a variable?

[1 mark]

- ☐ A The part of the program where it can be used
- ☐ B The largest value it can hold
- ☐ C Its data type
- ☐ D How long its name is

The task: count the beeps

This program should beep three times and print `beeps: 3`, but it crashes. Fix it, with parameters and `return` or with `global`.

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

beeps = 0

def beep():
    tone(880, 0.1)
    wait(0.1)
    beeps = beeps + 1

beep()
beep()
beep()
print("beeps:", beeps)
```

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



Do it on the robot

www.bugbotlab.com/learn/f4-3-local-and-global-variables/

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

Challenges

1. Write `longest_so_far(reading, best)` that returns whichever is larger, and use it in a loop of readings without any global variables.
2. Add a global constant `NOTE = 660` and use it inside `beep`. Why does reading it work without `global`?
3. Find a program from an earlier lesson that uses variables inside a function, and say which are local and which are global.