



F1.6 Variables, constants and assignment

Programming basics · GCSE · OCR J277 2.2.1, AQA 8525 3.2.2, Edexcel
1CP2 6.3.2 · about 15 min

What this lesson is about

Storing, changing and naming values, and constants that never change.

- Giving a value a name
- Using a variable
- Naming
- Variables can hold the robot's answers
- Changing a variable
- Constants: values that should never change

Questions

8 marks in all

1. What does this program print?

[1 mark]

```
count = 0
count = count + 1
count = count + 1
print(count)
```

Answer:

2

Each `count = count + 1` works out the right-hand side with the current value, then stores the result back.

2. What does the `=` in `speed = 60` mean?

[1 mark]

- ☐ A Store the value 60 under the name speed
- ☐ B speed is equal to 60, as in maths
- ☐ C Check whether speed is 60
- ☐ D Print speed

Answer: A. `=` stores a value in a name. Read it right to left: take 60 and store it in speed.

3. Which of these are allowed as variable names?

[1 mark]

Tick every answer that is true.

- ☐ A `leg_length`
- ☐ B `speed2`
- ☐ C `2speed`
- ☐ D `leg length`

Answer: A, B. Names use letters, numbers and underscores, cannot start with a number, and cannot contain spaces.

4. `speed = 60` then `print(Speed)` . What happens? [1 mark]
- ☐ A NameError: Speed is a different name from speed
 - ☐ B It prints 60
 - ☐ C It prints Speed
 - ☐ D It prints 0
- Answer:** A. Names are case-sensitive, and Speed was never given a value.
5. What does this program print? [1 mark]
- ```
leg = 24
half = leg / 2
print(leg, half)
```
- Answer:**
- 24 12.0
- half stores the result of the calculation, and dividing gives a float.
6. Why store `before = distance()` before driving? [1 mark]
- ☐ A So the first reading can be compared with one taken later
  - ☐ B Because distance() only works once
  - ☐ C To make the robot drive further
  - ☐ D It makes no difference
- Answer:** A. Without a variable, the first reading would be gone by the time you took the second.
7. What is the difference between a variable and a constant? [1 mark]
- ☐ A A variable's value can change while the program runs; a constant's does not
  - ☐ B A constant can hold text and a variable cannot
  - ☐ C A variable is faster
  - ☐ D There is no difference in Python
- Answer:** A. A constant is set once and not changed. Python does not enforce it, so constants are written in capitals as a promise.
8. Four drives all use `MAX_SPEED` . Why is that better than typing 40 four times? [1 mark]
- ☐ A The speed only has to be changed in one place
  - ☐ B It makes the robot drive further
  - ☐ C Python runs constants faster
  - ☐ D Numbers are not allowed as arguments
- Answer:** A. One place to change the value, and no chance of missing one of the four copies.

## The task: speed report

---

Store 70 in a variable called `speed`. Print `speed: 70` and `half: 35`, both **using the variable** (the number 70 appears only on the first line, and 35 nowhere). Dividing with `/` prints `35.0`, which is not `35 : //` divides and keeps a whole number, and lesson F1.9 explains the difference. Then drive 20 cm at that speed.

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

speed = 70
print("speed:", speed)
```

**The hint students can ask for:** Put 70 in a variable called `speed`. Print `speed: 70` and `half: 35` using the variable, then drive 20 cm at that speed.

### A solution

```
from bugbot import *
connect()
speed = 70
print('speed:', speed)
print('half:', speed // 2)
forward(speed, distance=20)
```

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