



F2.2 Selection: if

Decisions and loops · GCSE · OCR J277 2.2.1, AQA 8525 3.2.4, Edexcel
1CP2 6.5.2 · about 12 min

Name _____

Class _____

Date _____

What this lesson is about

Relational operators, and a block that runs only when a condition is true.

- Questions with a yes-or-no answer
- if: run a block only when the answer is True
- Comparing text
- The answer can come from a variable

Questions

9 marks in all

1. What does this program print?

[1 mark]

```
speed = 40
if speed > 50:
    print("fast")
if speed > 30:
    print("medium")
if speed > 10:
    print("slow")
print("checked")
```

2. What is the difference between `=` and `==` ?

[1 mark]

- ☐ A `=` stores a value, `==` asks whether two values are equal
- ☐ B They mean the same
- ☐ C `==` stores a value, `=` compares
- ☐ D `=` is for text, `==` for numbers

3. Which comparison means "is not equal to"?

[1 mark]

- ☐ A `!=`
- ☐ B `<>`
- ☐ C `=!`
- ☐ D `not=`

4. What does this program print?

[1 mark]

```
d = 25
print(d < 30)
print(d >= 30)
```

5. What happens to the block under an `if` when the question is False? [1 mark]
- ☐ A It is skipped, and the program carries on below it
 - ☐ B It runs anyway
 - ☐ C The program stops
 - ☐ D Python shows an error

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

```
x = 7
if x % 2 == 0:
    print("even")
print("done")
```

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

```
answer = "Yes"
if answer == "yes":
    print("driving")
print("done")
```

8. `d` is exactly 40. Which condition is True? [1 mark]

- ☐ A `d <= 40`
- ☐ B `d < 40`
- ☐ C `d > 40`
- ☐ D `d != 40`

9. How does AQA pseudo-code write "is equal to"? [1 mark]

- ☐ A `=`
- ☐ B `==`
- ☐ C `===`
- ☐ D `EQUALS`

The task: near or far

Drive 20 cm. Then, with two separate `if` s: if the wall is less than 40 cm away, print `near` and make the LED red; if it is 40 or more, print `far` and make it green. The wall has been placed so the answer should be near, but write both.

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

forward(50, distance=20)
d = distance()
if d < 40:
    print("near")
```

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



Do it on the robot

www.bugbotlab.com/learn/f2-2-selection-if/

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

Challenges

1. Print "even" if the battery percentage is an even number (remember `%` from lesson F1.9).
2. Drive 10 cm only if the way ahead is clear for at least 30 cm, and print what you decided.
3. Ask for a password with `input` and turn the LED green only if it is right.