WORKSHEET

F2.6 Condition-controlled loops: while

Decisions and loops · GCSE · OCR J277 2.2.1, AQA 8525 3.2.2, Edexcel 1CP2 6.2.2 · about 15 min

BugBotLab

Name _____ Class _____ Date _____

What this lesson is about

Looping until something changes, break, and loops that never end.

- while
- Counting with while
- Asking until the answer is right
- for or while?
- break: leave a loop early
- Loops that never end

Questions

8 marks in all

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

```
n = 1
while n < 20:
    print(n)
    n = n * 2
```

2. When should you use `while` rather than `for`? [1 mark]

- ☐ **A** When you will only know when to stop by looking, such as until the wall is close
- ☐ **B** When you know how many times
- ☐ **C** Always
- ☐ **D** Never: while is out of date

3. `count = 0` then `while count < 5: print(count)` with nothing else in the block. What happens? [1 mark]

- ☐ **A** It never ends: count never changes, so the question stays True
- ☐ **B** It prints 0 to 4
- ☐ **C** It prints nothing
- ☐ **D** SyntaxError

4. What does `break` do? [1 mark]

- ☐ **A** Leaves the loop immediately
- ☐ **B** Stops the whole program with an error
- ☐ **C** Skips one line
- ☐ **D** Pauses the robot

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

```
for i in range(10):
    if i == 3:
        break
    print(i)
print("out")
```

6. Why is there a `wait(0.1)` inside a loop that drives until the wall is close? [1 mark]

- ☐ **A** It gives the robot time to move before the next check
- ☐ **B** It makes the robot drive further
- ☐ **C** Python needs it to loop
- ☐ **D** It stops the robot

7. What is the difference between a while loop and REPEAT ... UNTIL? [1 mark]

- ☐ **A** REPEAT ... UNTIL checks at the end, so its block always runs at least once
- ☐ **B** There is no difference
- ☐ **C** while always runs at least once
- ☐ **D** REPEAT ... UNTIL can only count

8. What does AQA call a condition-controlled loop? [1 mark]

- ☐ **A** Indefinite iteration
- ☐ **B** Definite iteration
- ☐ **C** Nested selection
- ☐ **D** A subroutine

The task: creep to the wall

Use a `while` loop: while the wall is more than 25 cm away, drive 5 cm and print the distance. Stop inside the green band, without touching the wall.

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

while distance() > 25:
    forward(50, distance=5)
```

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

Do it on the robot

www.bugbotlab.com/learn/f2-6-condition-controlled-loops-while/

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

Challenges

1. In creep to the wall, count the steps and print the count at the end.
2. Write a loop that halves a number starting at 100 until it is below 1, printing each value.
3. Use `break` to stop a `while True:` loop when the robot has driven 50 cm, using `position()`.