

Microbit Lesson 4: Compound conditionals

Video

- [Microbit lesson 4 video](#)

Conditional operators

We can combine the two previous ideas of conditionally executing code and storing values in variables. We can ask Python to inspect the value of a variable and use that to determine whether to execute an "if statement" or "while loop".

While we're at it, let's also discover how to use the accelerometer. The following will continually read the x,y,z axis values from the accelerometer and store them into 3 variables: *x*, *y*, and *z*.

We can query those values to determine which way the microbit is angled.

```
from microbit import *

while not button_a.was_pressed():
    x,y,z = accelerometer.get_values()
    if x > 50:
        display.show(Image.ARROW_E)
    elif x < -50:
        display.show(Image.ARROW_W)
    else:
        display.show(".")
display.scroll("bye!")
```

Tilt tell

We can combine multiple *if* statements to use this trick to include the *x* and *y* axis.

```
from microbit import *
x,y,z = accelerometer.get_values()
while not button_a.was_pressed():
    if x < -200: # Tilted left
        if y < -200:
            display.show(Image.ARROW_NW)
        elif y > 200:
            display.show(Image.ARROW_SW)
        else:
            display.show(Image.ARROW_W)
    elif x > 200: # Tilted right
        if y < -200:
            display.show(Image.ARROW_NE)
        elif y > 200:
```

```

        display.show(Image.ARROW_SE)
    else:
        display.show(Image.ARROW_E)
else: # Neutral x-axis tilt
    if y < -200:
        display.show(Image.ARROW_N)
    elif y > 200:
        display.show(Image.ARROW_S)
    else:
        display.show(Image.HAPPY)
x,y,z = accelerometer.get_values()
display.scroll("Bye!")

```

Conditional execution operators

When posing a question for an `if` statement or `while` loop, the following examples illustrate the different comparison operations available.

- Greater than ... `>`
- Less than ... `<`
- Equals ... `==` ... note this is a double equal sign. A single equal sign SETS a value, a double equal sign COMPARES two values.

```

a = 10
b = 10
while True:
    if button_a.was_pressed():
        a = a + 1
    if button_b.was_pressed():
        a = a - 1
    if a > b:
        display.scroll("a is larger")
    if a == b:
        display.scroll("values are same")
    if a < b:
        display.scroll("a is smaller")

```

Rolling marble style game

Here is a more complicated example. Note that by the use of `or` in the `if` statement we can set multiple conditions in one go.

```

from microbit import *
row = 2
column = 2
keep_playing = True
score = 0
while keep_playing:

```

```
x,y,z = accelerometer.get_values()
if x < -100:
    column = column - 1
if x > 100:
    column = column + 1
if y < -100:
    row = row - 1
if y > 100:
    row = row + 1
if row > 4 or row < 0 or column > 4 or column < 0:
    keep_playing = False
else:
    display.clear()
    display.set_pixel(column, row, 9)
    score = score + 1
    sleep(200)
display.scroll("Game over")
display.scroll(score)
```

Activity

Make your own personalised modifications to the marble game.

For instance, can you create an easy mode / hard mode option?