

micro:bit buttons

Due Tuesday 4/26 (6th period), Wednesday 4/27/22 (7th period)

Today we're going to play with the micro:bit's buttons. There are two buttons, `button_a` and `button_b`, clearly labeled on the micro:bit. Buttons have three methods that can be used with either button:

```
is_pressed()      was_pressed()      get_presses()
```

The first two are booleans (that means they return either `True` or `False`) and the last one returns a number.

Here is an example of code using `is_pressed` with `button_b` where the micro:bit displays a heart if the user is pressing the B button and otherwise shows a question mark.

```
while True:
    if button_b.is_pressed():
        display.show(Image.HEART)
    else:
        display.show("?")
```

Here is an example of waiting for a bit and then checking if `button_b` was pressed during the pause:

```
while True:
    sleep(5000)
    if button_b.was_pressed():
        display.show("!")
    else:
        display.show(Image.HEART)
```

Finally, here is an example of code that waits and then displays how many times the user pressed `button_a`:

```
sleep(5000)
c = button_a.get_presses()
display.scroll(str(c))
```

Notice that you have to put the number, `button_a.get_presses()`, into an `str()` call to turn it into a string before you are able to display the number. The `display.scroll()` function only displays text.

A whole tutorial on how to use buttons is at the following link on the micro:bit web site if you want more information or examples. You don't have to go through it, but if you're struggling or want more information or ideas, please do.

<https://microbit-micropython.readthedocs.io/en/latest/tutorials/buttons.html>

(continued on next page)

Your task today:

Create a number guessing game on the micro:bit as follows:

- Create a program named "**microbit buttons**, your name" saved to your computer.
- Put your name in a comment at the top.
- Pick a random number from 0 to 9. See notes below.
- Make the micro:bit show a question mark symbol for six seconds, giving the user time to guess a number by pressing button_a several times.
- The micro:bit then displays the user's guess for one second.
- If the user guessed the secret number, display a happy face for one second, then say "BYE", then be done. Use the break command to leave the while True loop.
- If the user guessed too low or too high, show an image that indicates if the guess was too high (for example, the down arrow to show them that they need to guess a smaller number) or too low for a few seconds, then show a sad face for 1 second, then go back to the ? to let the user guess again.

Here's a link to all the Images that are built in: <https://microbit-micropython.readthedocs.io/en/latest/tutorials/images.html>

To get a random number you'll have to add an "import random" to the top of your program. Then, when you want a random number you can do this:

```
r = random.randint(0,9)
```

Make sure your random number is only chosen once per game. In other words, make sure you aren't choosing a different random number for each turn.

To display the user's guess, you need to do this:

```
display.show(guess)
```

You'll use an **if elif else** structure to evaluate the user's guess.

Call me over to see your program when you are done. Also, please turn in your code on the Google Classroom.

Extra credit challenge: make the game begin again with a new number if the user presses button_b after the game is over.