## Lessons 13 – Activity Sheet

## Getting Started

As well as creating and selecting random numbers the micro:bit can also select a random item from a list. Consider the program below.

from microbit import \*

import random

names = ["Mary", "Yolanda", "Damien", "Alia", "Kushal", "Mei Xiu", "Zoltan"]

display.scroll(random.choice(names))

The program begins by importing the random module – we covered imports on Lesson 4. We then create a list – a variable that holds a number of items. The last line of code uses the random module to select one of the items in the list using the coderandom.choice()and then scrolls it across the LED matrix.

Try this example program out on your micro:bit or create your own version.

## **Creating an Encrypted Message**

Random numbers are used for many applications, from the lottery to gaming to keeping data secure. One method of doing this is to encrypt the contents of the message. A famous method is known as the Caesar cipher where each letter of the alphabet is assigned a number. When you write out your message, you write the number instead of the letter. Unless you know the number / letter conversion you will not understand the code.

Consider the phrase below, use the table to convert it.

**9 6 13 13 16 24 16 19 13 5**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** | **I** | **J** | **K** | **L** | **M** | **N** | **O** | **P** | **Q** | **R** | **S** | **T** | **U** | **V** | **W** | **X** | **Y** | **Z** |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 1 |

Now we are going to create our own encryption code using the micro:bit. Adapt the program below so that it selects a random number between 1 and 26 each time that you press button A. Download the program to your micro:bit.

from microbit import \*

import random

while True:

if button\_a.is\_pressed():

display.show(str(random.randint(1, 6)))

sleep(2000)

else:

display.show(Image.TSHIRT)

On line six there is a short, two second delay – why is this used? What happens if it is removed?

Now use the micro:bit to assign the numbers to each letter. Press Button A, the number displayed is the encrypted number for letter A. Press button A again, the number displayed is the encrypted code for letter B and so on. Add the values to the table below.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** | **I** | **J** | **K** | **L** | **M** | **N** | **O** | **P** | **Q** | **R** | **S** | **T** | **U** | **V** | **W** | **X** | **Y** | **Z** |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Write out your message**

Now that you have the letters and the code, write out your message below.

Use your encryption table to convert the message. Share with another Learner – can they break your code and decode your message?

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## Success Criteria

* The program gives you a random number
* Use the program to write a short word
* The program gives you all 26 letters
* Use the program to write a longer sentence
* Use Button A to trigger the random number to be displayed
* Use Buttons A and B

## Pro-tip

The programs use a lot of brackets, ( and ), remember that each open bracket needs to be closed with a ). Some lines of code have several open brackets which require closing. Errors are produced if they are not closed.

## Test Time

Download your program code to your micro:bit and test it. Each time you press Button A it should display a random number between 1 and 26. You may want to increase the delay between the number being displayed and the program stopping to ensure that you have time to read the number.

## Stretch Tasks

Reverse the program so that each time you press Button A the program displays a random letter. Then add the letter to the table below. For example, if the first letter displayed is X then write X in position one, X is now one

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** | **21** | **22** | **23** | **24** | **25** | **26** |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

You will need to use and adapt the program below.

from microbit import \*

import random

names = ["Mary", "Yolanda", "Damien", "Alia", "Kushal", "Mei Xiu", "Zoltan"]

display.scroll(random.choice(names))

Instead of using random numbers you could use random letters. For example, letter A is replaced with letter H, letter B is replaced with letter F and so on. Use lists and the random.choice() code to select the letters.

Write a program that uses both the A and B buttons. When you press Button A it displays a random letter, which you write into the table below. When you press Button B it displays a random number, add this number to the table below. A truly random letter and number generator!

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## Final Thoughts

You have just learned how to:

* Use the random module
* Select a random number
* Select a random item from a list
* Combine this with a button press
* Write a program to create an encrypted code.

Thinking about your program, what are the limitations of the program? How could this program be used in a real-life application?

You can read more about encryption and its uses here: <https://www.globalsign.com/en/blog/glossary-of-cryptographic-algorithms/>

How about trying out some if these encryption games from the CIA <https://www.cia.gov/kids-page/games/break-the-code>