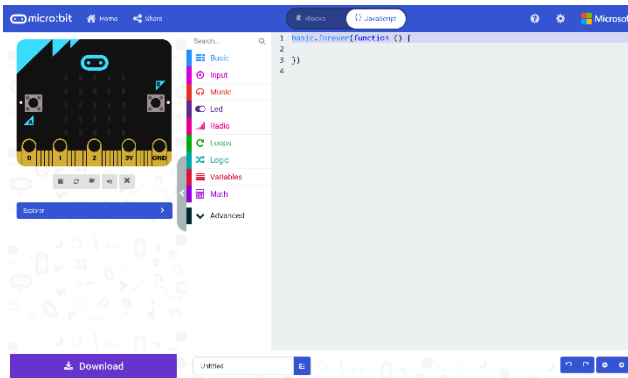


Show Message

Step 1



Go to microbit.org, then select the **Let's Code** option, next in the **MakeCode Editor** section select the **Let's Code** button, finally select the **New Project** button and select the **JavaScript** tab

Step 2

With the **JavaScript** tab selected in the **MakeCode Editor** you should remove the following code:

```
basic.forever(() => {  
})
```

Then in the **MakeCode Editor** you should enter the following code:

```
const items: string[] = [  
  "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",  
  "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "!", "?", " "  
];
```

const means a value that doesn't change and this is set to a **string[]** called **items** which is a special kind of value known as an array, which is a list of values, these can be identified with the use of a set of two square brackets, here it is a list of string. To get a particular value from an array you need an index, which is the position in the array, this starts from 0 for the first item and 1 for second and so on, so for example to get the fifth value from the array you'd use the index of 4.

Step 3

While still in the **JavaScript** tab of the **MakeCode Editor**, below the code entered in the previous step, you should enter the following code:

```
let counter: number = 0;
let editing: boolean = false;
let display: string = "";
let message: string = "";
```

let helps create another kind of value, these can change and are known as variables which can contain a single value, there are different types of value used here. First there's a **number** which can store a whole number such as 0, 1, 2, 3 and so on. Then there's a **boolean** this can be either **true** or **false** and here is set to **false**. Then there are two strings, and these can be any kind of text, they have been set to an empty string which is "".

Step 4

Again, while still in the **JavaScript** tab, below the code entered in the previous step, you should enter the following code:

```
function show(item: string) {
    basic.showString(item)
}
```

function is a block of code that you can use many times to do the same thing, they can also take in values known as parameters to use in the **function**. The parameter used here is called **item** and it is a **string**. The **function** will use this value to use or call the **basic.showString** built in **function** to display text on the **micro:bit** using the LEDs on the front.

Step 5

Once again while still in the **JavaScript** tab, below the code entered in the previous step, you should enter the following code:

```
input.onButtonPressed(Button.A, () => {
  if (counter == items.length) {
    counter = 0;
  }
  display = items[counter];
  counter++;
})

input.onButtonPressed(Button.B, () => {
  message += display;
  show("+");
})

input.onButtonPressed(Button.AB, () => {
  editing = !editing;
})
```

input are things that happen, also known as events, when you do something with the **micro:bit**, they are as follows:

1. The first **input** is when you press the **A** button on the **micro:bit** and when you do it will check the **counter** to see if it is the same value as the size or **length** of the **list string[]** array, then it will set **display** to be the item from the array with the same index as the **counter** – if this is the first item it will have the value "**A**", if second it will be "**B**" and so on, the line **counter++** will increase the value of **counter** by one for the next item.
2. The second **input** is when you press the **B** button on the **micro:bit** and when you do the **message += display** will add the whatever is in the **display** value onto the end of the **message** value, it will display a "+" on the **micro:bit** by using the **function show** briefly before continuing.
3. The third **input** is when you press both the **A** and **B** buttons on the **micro:bit** at the same time **editing = !editing** will set the value of **editing** to the opposite of its value, this is what the **!** does which is known as **not**.

Step 6

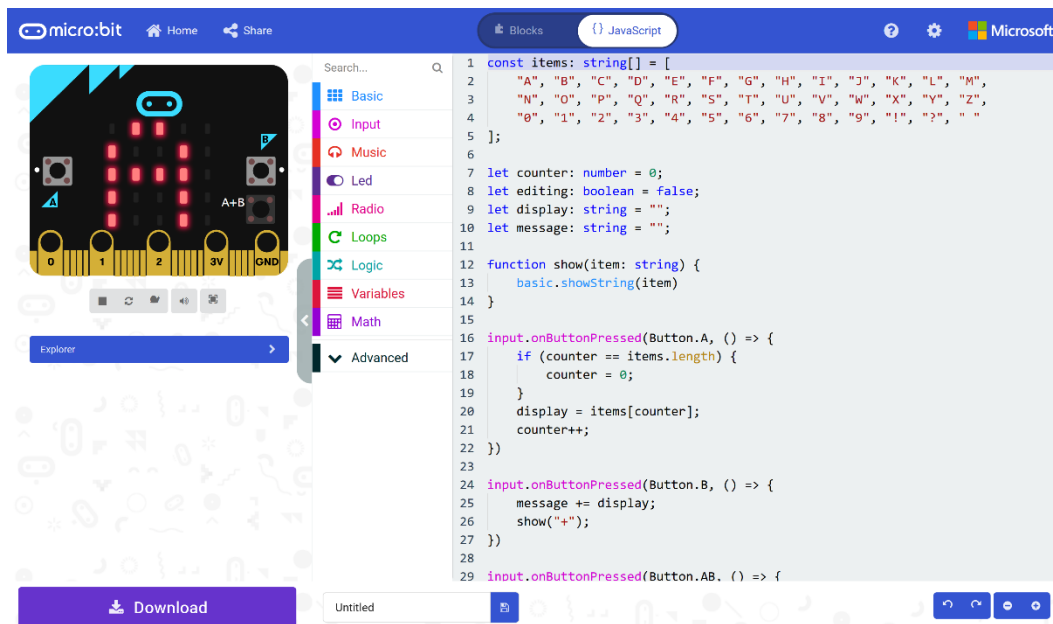
Finally, while still in the **JavaScript** tab, below the code entered in the previous step, you should enter the following code:

```
basic.forever(() => {
  if (editing) {
    show(display);
  }
  else {
    show(message);
  }
})
```

basic.forever is a **function** which will repeat, or loop, forever as long as the example is running on the **micro:bit**. Inside this **function** it will check the **editing** value in an **if** statement, when it is **true** the first part of the code will happen, this will show the **display** value on the **micro:bit** and in the second part, when **editing** is **false** it will show the message value on the **micro:bit** instead.

Step 7

Once done the **MakeCode Editor** should appear as follows:



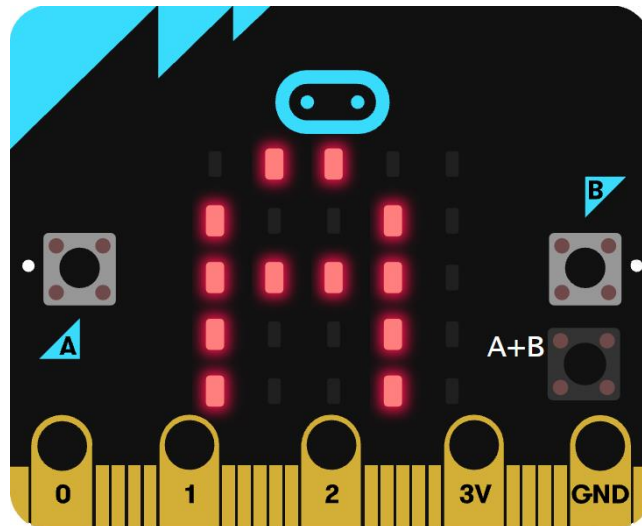
Step 8



That completes the **micro:bit** example, if not done already you can select the **Start the simulator** button to start the example

Step 9

When running on the virtual **micro:bit** you can create a message on the LEDs by pressing the **A+B** button on the **micro:bit** then press the **A** button to cycle through the list of letters and numbers and then press **B** button to add the letter or number to the list to be displayed, you can keep doing this to add more letters and numbers then once the message is complete press **A+B** again to display the full message.



You can also run the example on an actual **micro:bit** by connecting one to your computer and then choosing the **Download** option in the **MakeCode Editor** to download the example to your computer. Once downloaded you can then copy the **.hex** file from where you've downloaded it to the **micro:bit** the same way you'd copy to another drive or device connected to your computer, then once the example has been copied to the **micro:bit** it should start automatically.