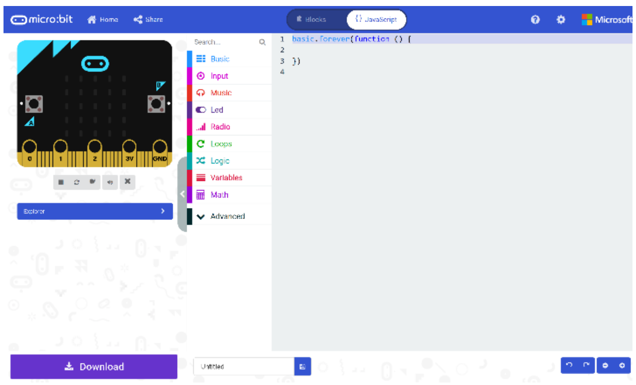
Compass Game

Step 1



```
basic.forever(function () {

})
```

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:

```
let compass: boolean = true;
let current: string = '';
let pattern: string[] = [];
```

let helps create a value which can change and are known as variables which can contain a single value. Here there's a **boolean** which can store either **true** or **false** it is called **compass** and is set to **true**, then there is string called current and is set to an empty value or **''**, finally there's an array, or list of **string** values which is denoted by the use of two square brackets or **[]** and is set to an empty array which is also a pair of square brackets or **[]**.

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:

```
function heading(bearing: number): string {
    let result: string = "N";
    if (bearing < 45) {
        result = "N";
    } else if (bearing < 135) {
        result = "E";
    } else if (bearing < 225) {
        result = "S"
    } else if (bearing < 315) {
        result = "W";
    }
    return result;
}

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.

1. The first **function** is called **heading** and this a **number** parameter called **bearing**. Inside there is a **let** which is what will be returned and is called **result** and is set to **"N"**. Following this is an **if** statement which uses the value of **bearing** to make a choice – when this value is less than **45** then the **result** will be set to **"N"**, otherwise when the bearing is less than **135** then the **result** will be set to **"E"** – there are other checks for other values and these all relate to the number of degrees around a circle that represent the points on a compass.
2. The second **function** is called **show** and 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 4

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, () => {
    pattern.push(current);
    compass = false;
    show("+");
    basic.pause(500);
    compass = true;
})

input.onButtonPressed(Button.B, () => {
    let display: string = '';
    compass = false;
    show(display);
    for (let i: number = 0; i < pattern.length; i++) {
        display += pattern[i];
    }
    show(display);
    basic.pause(1000);
    compass = true;
})

input.onButtonPressed(Button.AB, () => {
    pattern = [];
})
```

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

3. The first **input** is when you press the **A** button on the **micro:bit** and when you do it will use **push** on the **pattern** list to add an item to the list or array. It then sets the **compass** value to **false** then it uses **show** to display a **+** on the **micro:bit** LEDs then is followed by a delay of half a second – this is what **basic.pause(500)** does, finally it sets the **compass** value back to **true**.

4. The second **input** is when you press the **B** button on the **micro:bit** – when you do it has a new value or variable called **display** which is set to an empty string or **''** then it sets the **compass** value to **false** then it uses **show** with this empty string to clear the LEDs. There is a **for** loop, which allows something to be repeated – in this case it repeats from **0** to the number of items in the **pattern** array which is what **pattern.length** does, then inside this **for** loop it appends the item at the position in **pattern** array of the value of **i** which is set by the loop to the **display** value. After this it then uses **show** to output the value of **display** to the **micro:bit** using the LEDs. It then has a delay of one second – which is what **basic.pause(1000)** does, then finally it sets the **compass** value back to **true**.

5. The third **input** is when you press both the **A** and **B** buttons on the **micro:bit** at the same time **pattern** will be reset to an empty array – this is what **[]** is.

Step 5

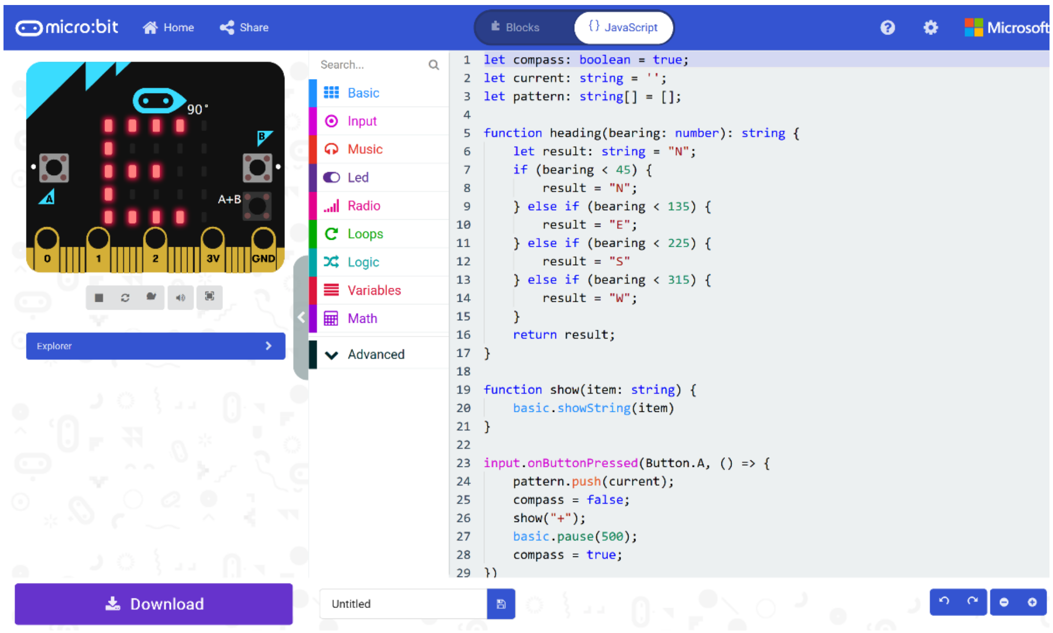
Finally, while still in the **JavaScript** tab, below the code entered in the previous step, you should enter the following code:

```
basic.forever(() => {
    if (compass) {
        current = heading(input.compassHeading());
        show(current);
    }
})
```

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 value of **compass** and when this is **true** it will then set the **current** value using the function called **heading** where the parameter **bearing** has been provided with the current direction the **micro:bit** is pointing – that is what **input.compassHeading()** does. Then it will use or call the function named **show** – this will display value of **current** on the **micro:bit** using the LEDs.

Step 6

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



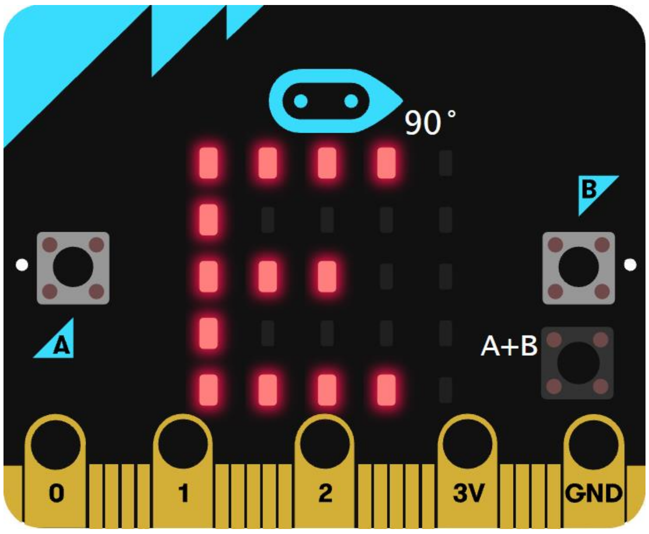
```
let compass: boolean = true;
let current: string = '';
let pattern: string[] = [];

function heading(bearing: number): string {
    let result: string = "N";
    if (bearing < 45) {
        result = "N";
    } else if (bearing < 135) {
        result = "E";
    } else if (bearing < 225) {
        result = "S"
    } else if (bearing < 315) {
        result = "W";
    }
    return result;
}

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

input.onButtonPressed(Button.A, () => {
    pattern.push(current);
    compass = false;
    show("+");
    basic.pause(500);
    compass = true;
})
```

Step 7

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

Step 8

When running on the virtual **micro:bit** you can set the compass by moving around the **micro:bit** logo this will display the direction it is pointing in, for example **S** for South – you can turn a real **micro:bit** in your hand to point in a particular direction do the same thing, you'll need to calibrate it by waving it around in a figure-of-eight in the air until the **micro:bit** indicates this has completed. You can then press the **A** button to add a direction to a list which can be displayed at any time by pressing the **B** button which will display all the directions added – to clear the list at any time you just press the **A** and **B** buttons together.



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.