![CodeHS logo]

# micro:bit Exploration: Teacher Version
*Intro to Programming with micro:bit*
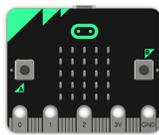
## Corresponding Material
1.1- Intro to micro:bit

## Discussion
Welcome to Physical Computing with micro:bit! In the coming lessons, you'll learn how to communicate with the computer in a language called JavaScript. We'll start by exploring and experimenting with programs and commands in the MakeCode editor.
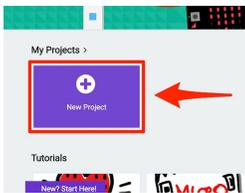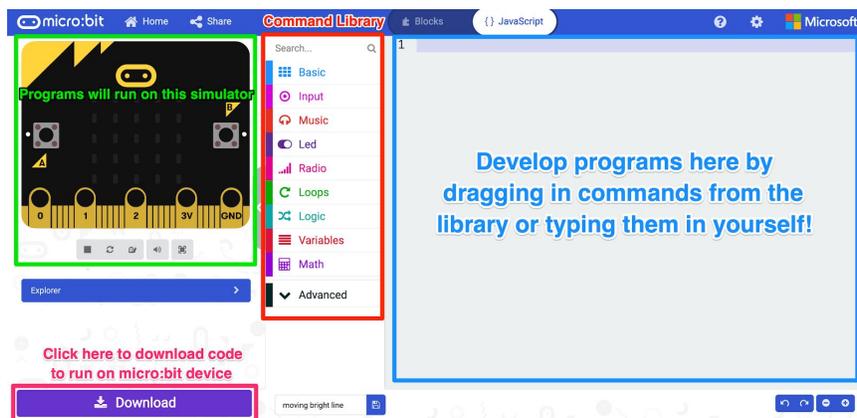
## Needed Materials

| (1) Micro:bit | (1) USB cord |
|---|---|
| | |

## Class Exercise

1. Head to MakeCode.microbit.org
2. Click on *New Project* (shown in the image to the left)
3. Click on *JavaScript* to switch from block to text coding.
4. Develop programs by dragging commands from the center library or by typing them into the editor yourself.
5. Test programs on the online simulator or run them to your micro:bit device by clicking the *Download* button.

Code #1: Explore the **showNumber** command
Remove any commands shown in the editor and type in the following code. (Make sure to type the text *exactly* as you see it, paying close attention to capitalization and punctuation.)

```
1 basic.showNumber(0)
2
```

Download the code to your micro:bit or run on the simulator and answer the following questions.

1) What do you see when you run the code on your micro:bit?
**The number 0 shows on the LED screen.**

2) What do you think will happen if you change the number inside the parentheses to **10**?
**The number 10 will show on the screen.** (*Student answers will vary here!*)

Go into the code editor and replace the **0** with a **10**. Download this code to your micro:bit or run on the simulator.

```
1 basic.showNumber(10)
2
```

3) Was your hypothesis correct? What happened when you ran the code?
**The number 10 scrolls across the screen once.**

4) Write a generic statement about how numbers show up on the micro:bit screen based on the number of digits in the number. Be as specific as possible.
**If the number is one digit, it is shown statically on the screen. If the number is more than 1 digit, the number scrolls across the screen one time and then the screen is blank.** (*The main thing we want students to start to do here is be specific and begin to note tiny functionality differences based on differences in commands.*)

Go into the code editor and replace the **10** with a **2020**. Download this code to your micro:bit or run on the simulator.

```
1 basic.showNumber(2020)
2
```

5) Was the generic statement you made correct? How can you tell?
**Yes, the number scrolled once across the screen because it contained more than 1 digit.**

Go into the code editor and replace the **2020** with your name. Try to run your program on the simulator.

```
1 basic.showNumber(Karel)
2
```

6) What happened when you tried to run the code?
**The program did not run. A message said to check for errors.** (*Here, students begin to see how errors can enter their programs if they aren't using a command correctly or don't fully know how it works.*)

Go into the code editor and replace your name with **2.5**. Try to run your program on the simulator.

```
1 basic.showNumber(2.5)
2
```

7) What happened when you tried to run the code?

**The number 2.5 scrolled across the screen as any multi-digit number has.** *(Students will not get into the difference between floats/doubles and integers in this course, but you can discuss that these numbers are viewed differently in computer programs if you'd like!)*

8) Write a generic statement that describes the type of element that can be entered into the `showNumber` command. Be as specific as possible.

**Any numerical value can be entered into the `showNumber` command. If a fraction is entered, the numbers are divided and the result is displayed.** *(You don't need to expect students to go right ahead and test a fraction, but it can be something you challenge them with! Instruct them to enter a fraction (2/4) and then ask them to explain why the micro:bit is displaying '.5' instead of '2/4'.)*

Code #2: Explore the **showString** command

Type the following code into your code editor.

```
1 basic.showString("Hello!")
2
```

Download the code to your micro:bit and answer the following questions.

1) What do you see when you run the code on your micro:bit?

**The text Hello! Scrolls across the screen the same as a multi-digit number. The quotation marks are not displayed on the screen, though.** *(Again, we want to get students into the habit of being as specific as possible as they go through these explorations because they need to fully understand a command and its limitations if they want to be able to use it correctly in their programs. If they don't come up with these specifics on their own, you can always prompt them by using probing questions.)*

2) What do you think will happen if you remove the quotation marks around the text `Hello!`?

**I don't know what will happen, but I think it will act differently because the quotation marks are probably there for a reason!** *(Note: It is fine for students to not know the answer but urge them to think critically about the command before and after the alteration.)*

Go into the code editor and remove the quotation marks around the word `Hello!`. Try to run your program on the simulator.

```
1 basic.showString(Hello!)
```

```
2
```

3) Was your hypothesis correct? What happened when you tried to run the code?

**The program did not run. The same message said to check for errors as when I tried to put letters inside the `showNumber` command.**

Go into the code editor and type "**8**" inside the parentheses. Download this code to your micro:bit or run on the simulator.

```
1 basic.showString("8")
2
```

4) What happened when you ran the code?

**The number 8 was shown statically on the screen, the same way it is if we used the `showNumber` command.** *(Encourage students to start making connections between all the commands they know as their toolbox grows.)*

Go into the code editor and type "**2.5**" inside the parentheses. Download this code to your micro:bit or run on the simulator.

```
1 basic.showString("2.5")
2
```

5) Write a generic statement that describes the type of element that can be entered into the **showString** command and how it shows on the screen. Be as specific as possible.

**Any value can be used in the `showString` command as long as the value is inside quotation marks. If there are not quotation marks present, the program will have an error and will not run. If the text is longer than one character, it will scroll across the screen. If it is only one character, it will be statically displayed.**

**Conclusion Questions**

1. How are the **showString** and **showNumber** commands different?

**The `showString` command displays anything but needs the item to be inside quotation marks. The `showNumber` command will only show numerical values and they don't need to be inside quotation marks.**

2. How are the **showString** and **showNumber** commands similar?

**The `showString` and `showNumber` commands both show the item entered in parentheses statically if it is one character in length, and scrolls it across the screen if it is longer than that.**