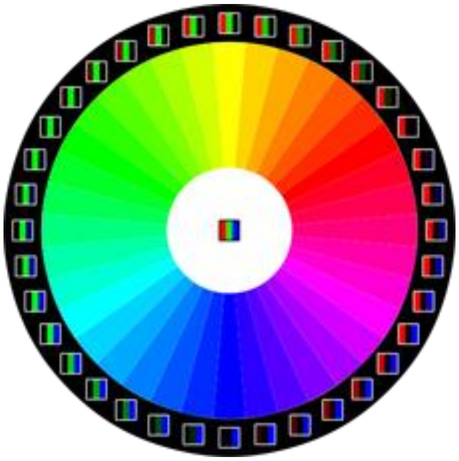
Changing Colors with Variables

Each LED has 3 different lights – Red, Green and Blue – all of which can be assigned a value between 0 and 255. The variety of the colors that can be attained using these primary colors is displayed in this color wheel:



In this tutorial, I want to display all the colors in this wheel on the robots LEDs. Have a think about what the pseudocode is going to be. Let’s start with the lights set to all blue and go clockwise round the wheel (so towards green). As we go around, we need to reduce the level of the blue and increase the level of the green:

```
Set LED to blue.
Repeat until fully green:
---- level of blue.
---- Increase level of green.
```

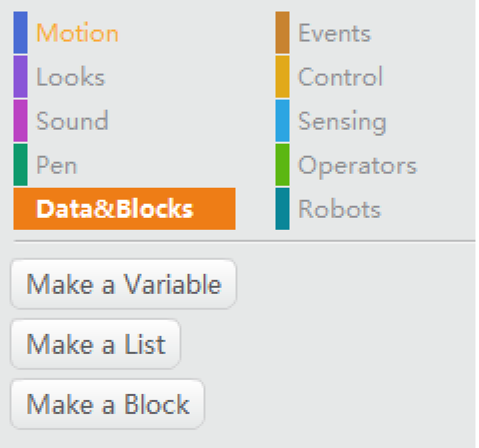
This is the start of the code. However, there is a problem. Once the LED has been set, we cannot “read” the levels of green and blue. So on the next loop, how do we know what to set the LEDs to? We need a variable.

Variables are places to store information. In mBlock, variables store numbers. Each variable is given a name. The name should be chosen to reflect what kind of information the variable holds. So, for my tutorial today, I am going to have 3 variables:

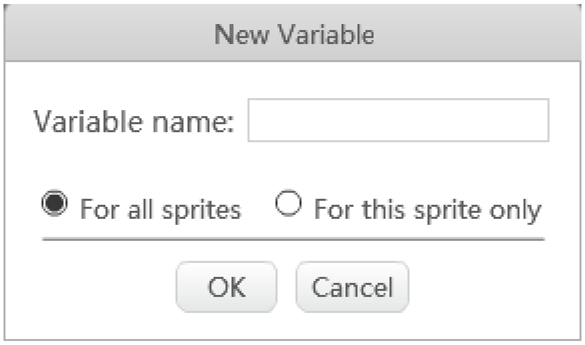
```
redSetting
greenSetting
blueSetting
```

A common naming convention is for variables to be all one word (no spaces) with the first letter being lower case and all subsequent words’ first letter being upper case.

To make a variable in mBlock, we need to go to the “Data&Blocks” script and choose “Make A Variable”:



When you press the button, it will ask you for a name:

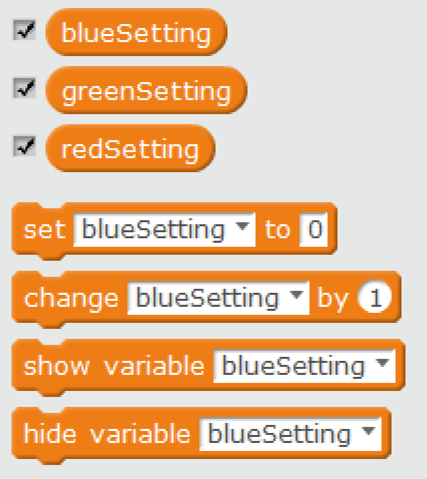


New Variable

Variable name:

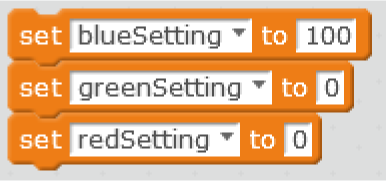
● For all sprites ○ For this sprite only

Make sure you choose a name that describes the information that the variable will hold. Once you have defined a variable, it will display some blocks related to the variable that you can use for programming:



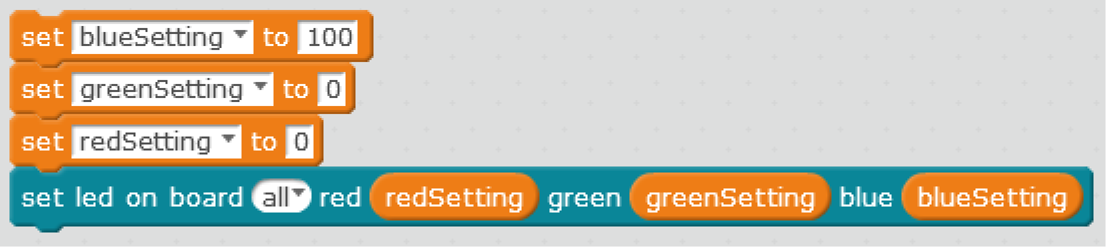
```
☑ blueSetting
☑ greenSetting
☑ redSetting

set blueSetting ▾ to 0
change blueSetting ▾ by 1
show variable blueSetting ▾
hide variable blueSetting ▾
```

Unless variables are going to be changed early on in a program, it is considered good practice to initialize variables at the start of a program. For my program, I want to start with blue turned on, and green and red turned off. I am going to turn the LEDs on up to a maximum value of 100:



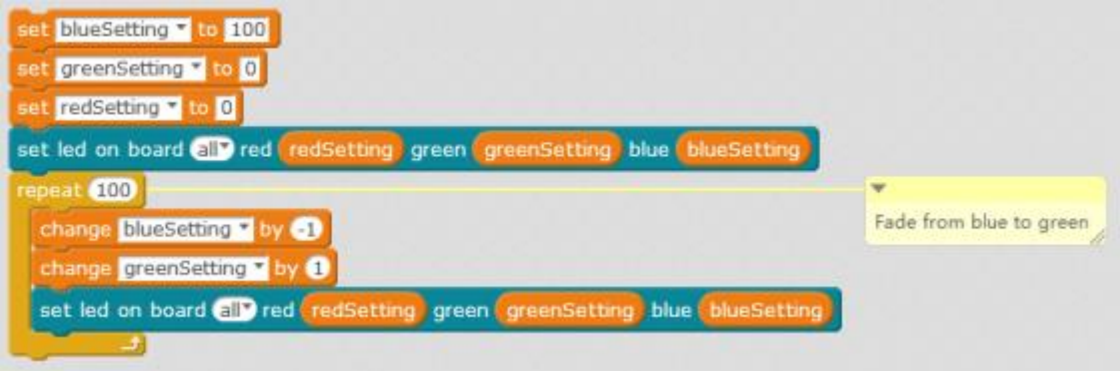
```
set blueSetting to 100
set greenSetting to 0
set redSetting to 0
```

And now I want to set the LEDs to the values of these variables:



```
set blueSetting to 100
set greenSetting to 0
set redSetting to 0
set led on board all red redSetting green greenSetting blue blueSetting
```

Now I want to decrease the blue setting one-by-one until I get to zero. At the same time, I want to increase the green setting until that gets to 100. So I will need to repeat this step 100 times:



```
set blueSetting to 100
set greenSetting to 0
set redSetting to 0
set led on board all red redSetting green greenSetting blue blueSetting
repeat 100
    change blueSetting by -1
    change greenSetting by 1
    set led on board all red redSetting green greenSetting blue blueSetting
```

Fade from blue to green

Challenge:

1. Can you finish the program so the LEDs continue to fade through the colors in the color wheel?
2. Can you write a program that measures the light (using the on-board light sensor) as the LEDs fade? What LED settings provide the darkest measurements? Write down any problems you face as you write this program. How did you solve the problems?