BBC Micro:bit 2

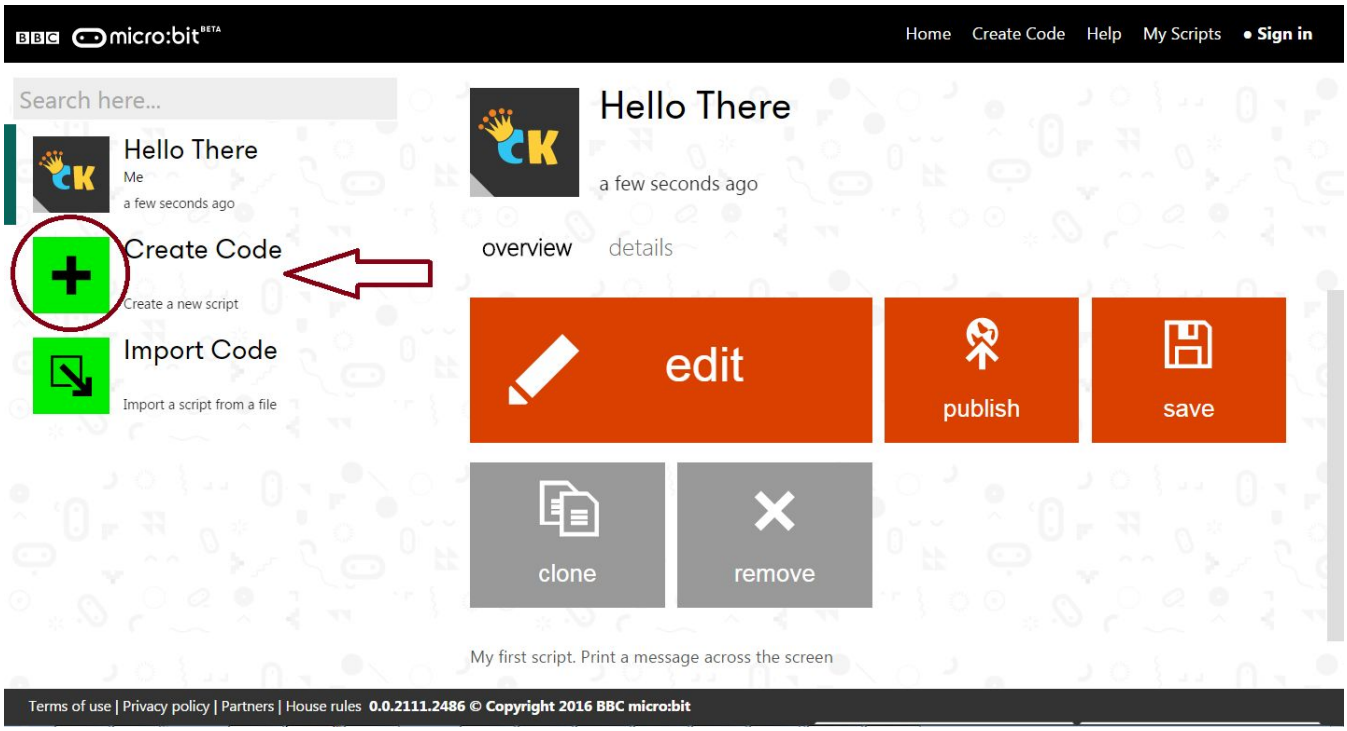
Project	Flashing Light
Description	Make the Micro:bit's LEDs flash.
Equipment Required	A computer that can access the web with a Javascript enabled browser.
Teaches	Use of the LEDs, use of the buttons, variables.
Prerequisites	Completed worksheet 1.

Step 1

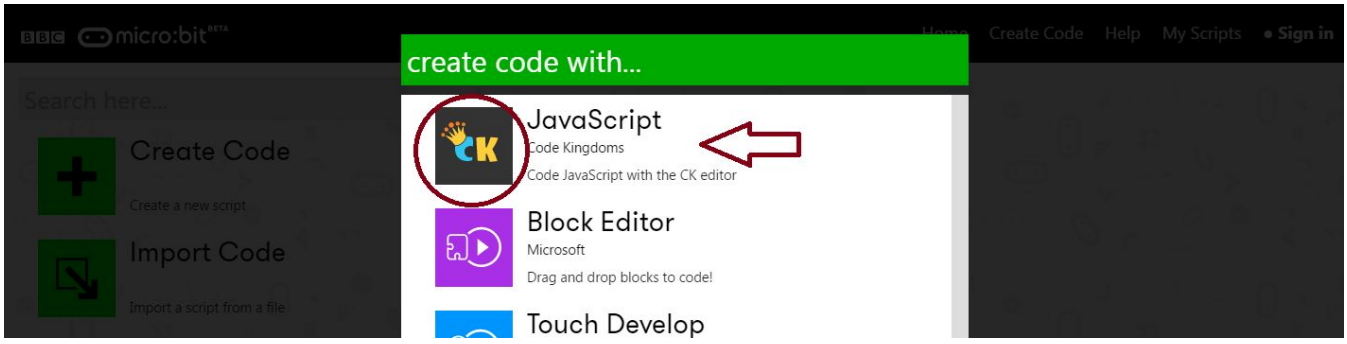
A new script

We're going to write some more code in Code kingdoms; this will flash all the LEDs on and off. If you're on the Micro:bit home page choose **Create Code**, then **New project** as we did in Worksheet 1.

If you're in the my scripts screen, click **Create Code**:



We're going to write another Code Kingdoms project so select this when asked:

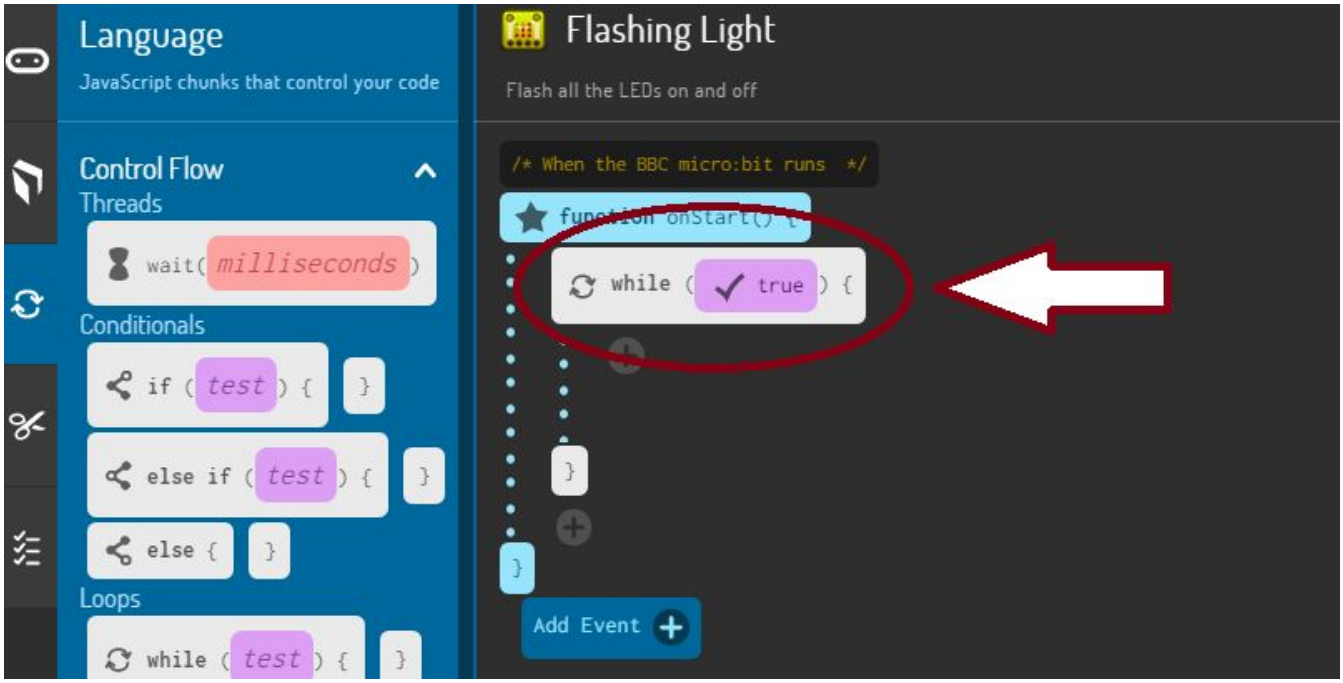


Step 2

Loop forever

Name your script '**Flashing Light**' and give it a description.

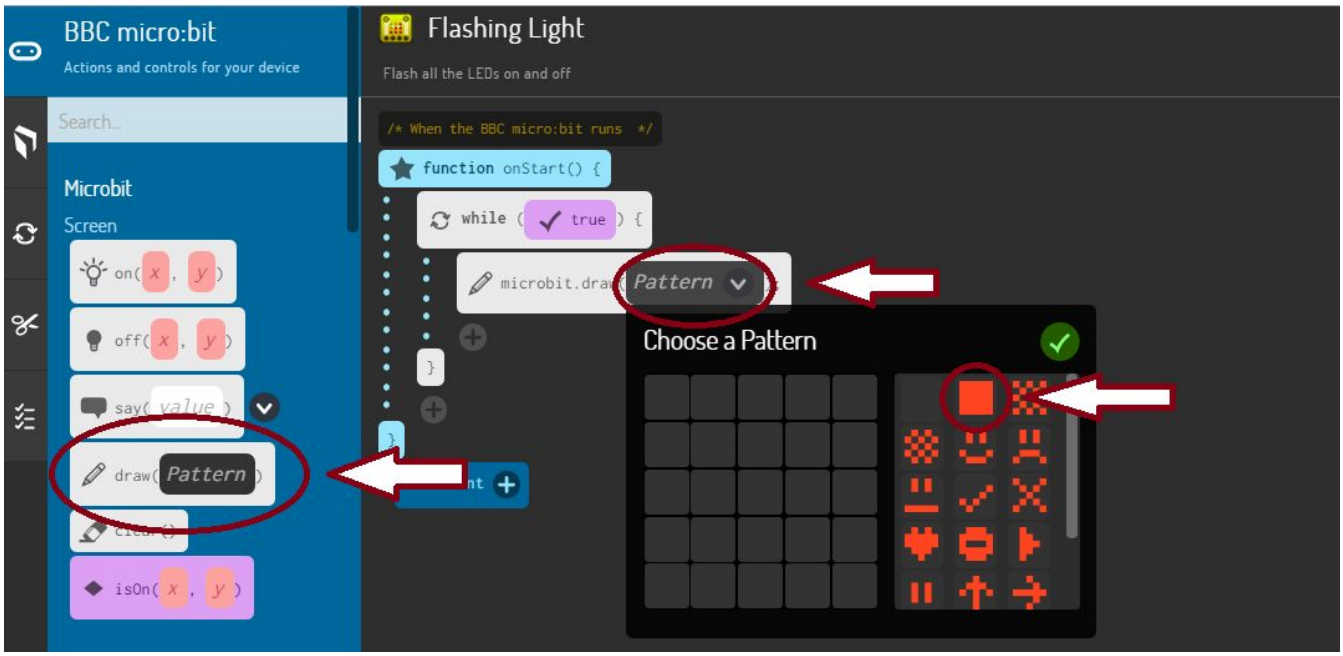
Just like the end of worksheet 1, we want our code to run forever so we're going to use a **while loop** from the **Language** menu. Drag it onto the coding area and use the **drop down menu** to select **true** for the **while** condition.



Step 3

Lights on

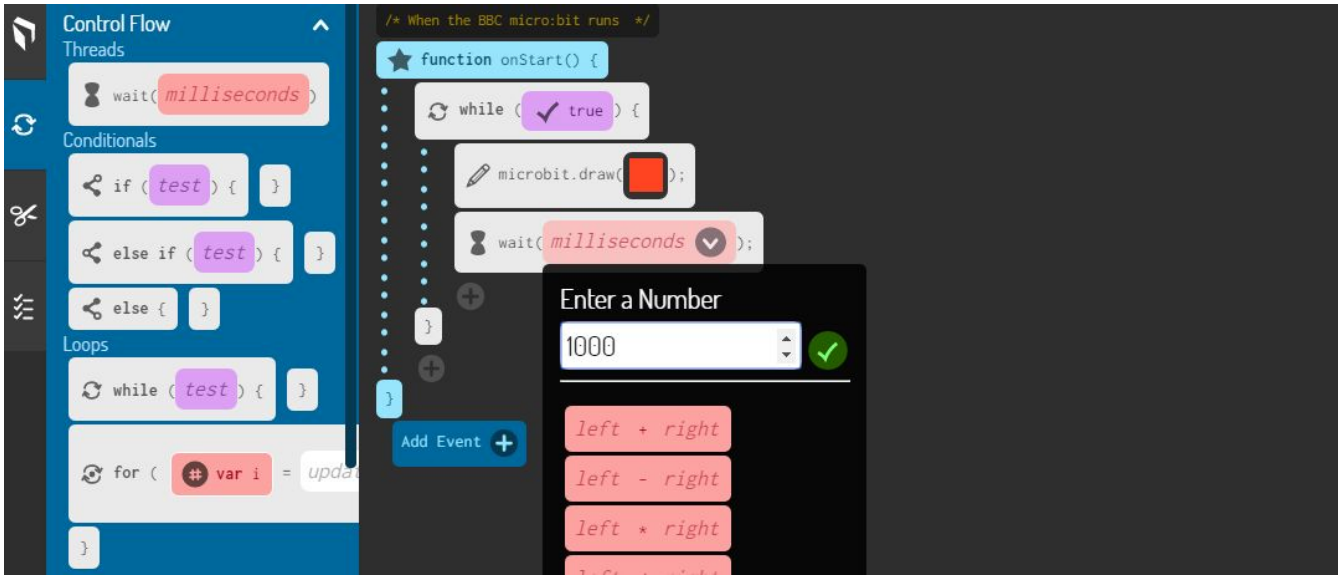
Next we want to turn all the LEDs on. We do this using the **Draw** chunk from the Micro:bit menu. Drag it so it's inside the **while loop** and use the **drop down menu** to select the '**all on**' pattern:



Step 4

Wait a moment

If we turn the LEDs on and off again without waiting for a moment, they won't appear to flash. So we need to add a **delay**. We do this using the '**wait**' chunk from the **Language** menu. Add it underneath the **draw** chunk. The **wait** chunk needs to know how long to wait, so select the **drop down menu** and enter **1000** as the number:



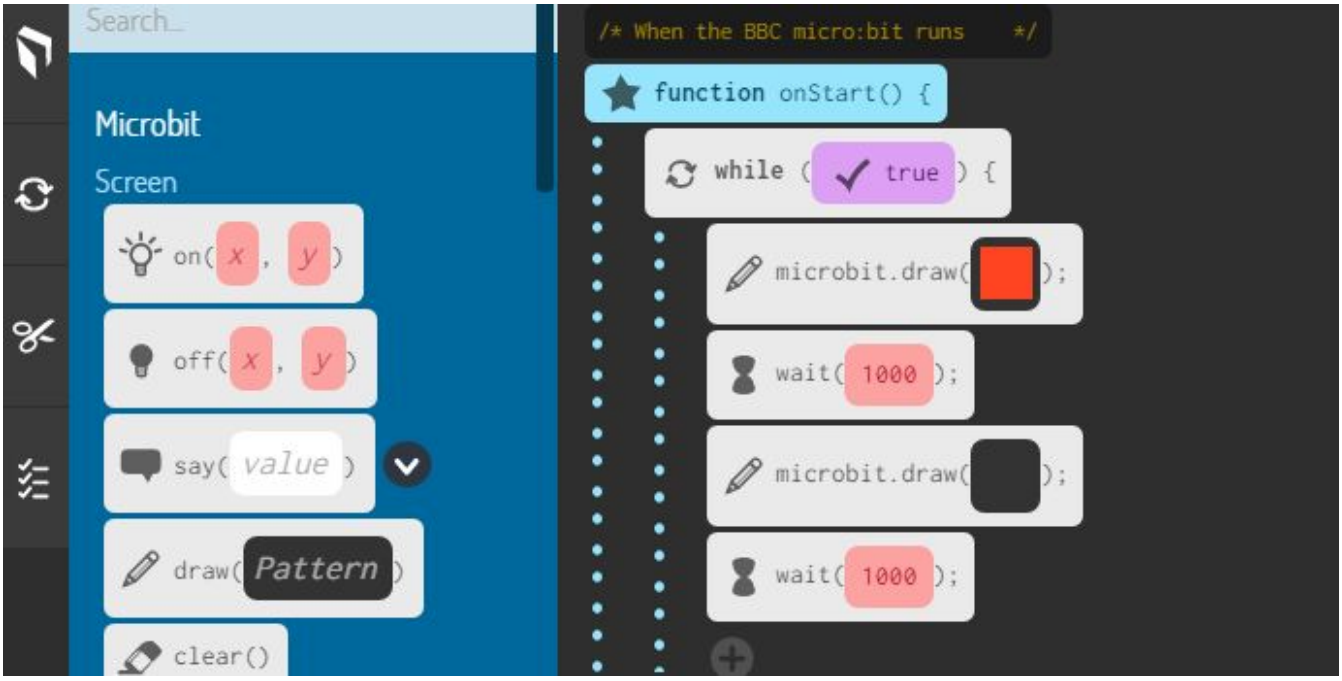
When a chunk needs more information like the wait chunk, the information is called a parameter. The wait chunk's parameter tells it how long to wait for until letting the code continue. Because we can have really short times to wait, its parameter is in milliseconds. There are 1000 milliseconds in a second so the code above will wait for 1 second.

Step 5

Off again

Now we want to repeat the above two steps but this time we want to turn the LEDs off and wait for a second. Can you work out how to do that and put the code chunks in?

Here's what it should look like:



Run the code in the simulator and on the Micro:bit if you can.

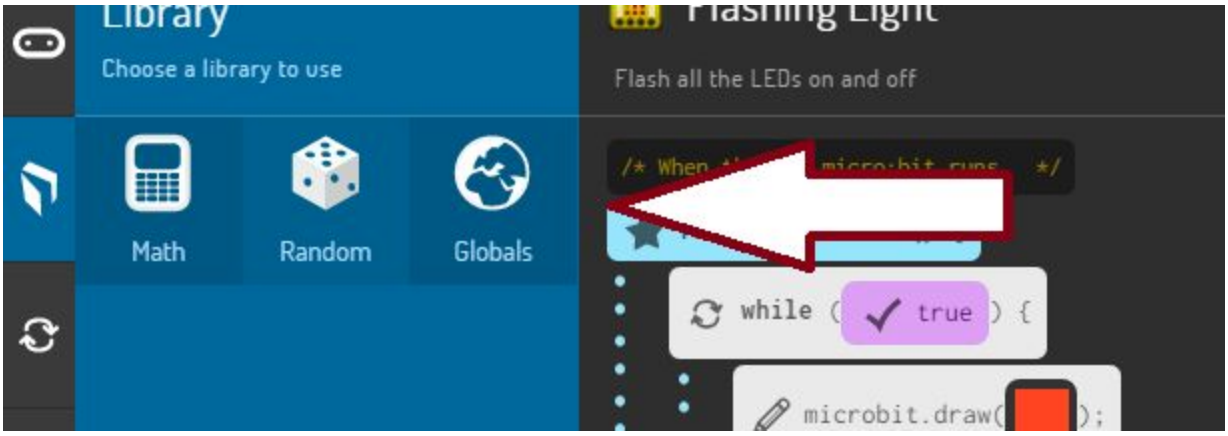
Try making the LEDs flash quicker and slower.
How about flashing different patterns?

Step 6

Variable

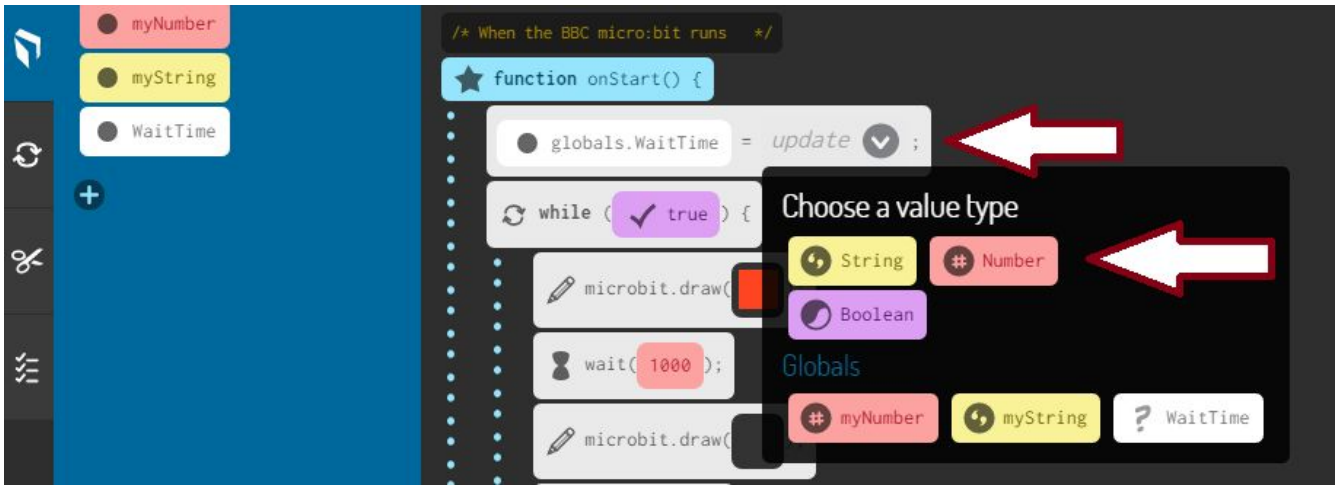
At the moment if we want to change the speed the LEDs flash at, we have to change two numbers; the wait after we turn the LEDs on and the wait after we turn them off. Suppose we want to be able to change one thing and have this alter both wait times. We can do this using a variable.

Select **Library** from the menu then click on **Globals**:

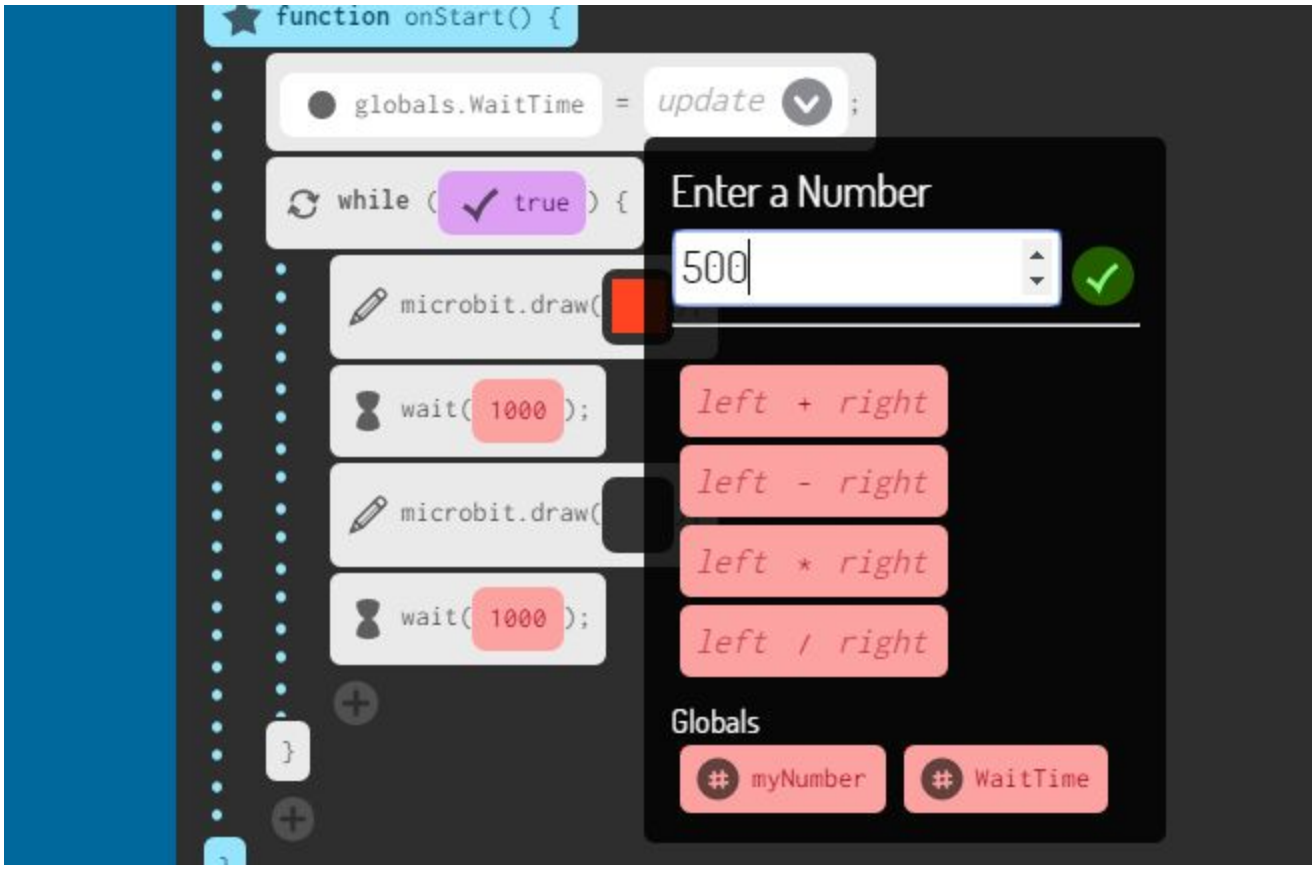


We're going to create a new variable by clicking on the **plus sign** in the chunks section.
As we're going to store how long we want to wait for, we'll call it **WaitTime** (variables don't normally have a space between words)

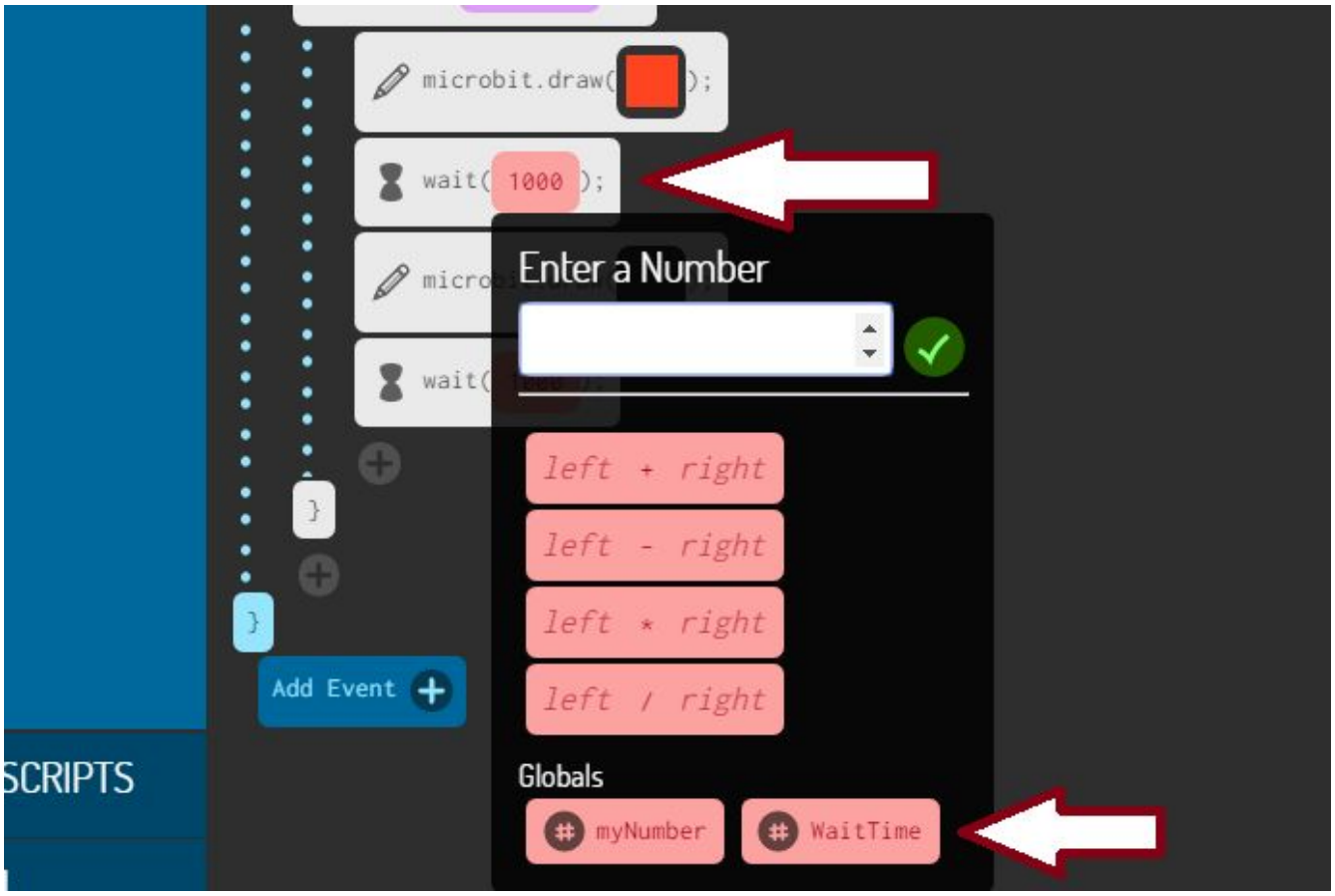
Now we need to set the variable to something in our code. Drag the new variable '**WaitTime**' into your code and drop it above the **while loop**. Click the **down arrow** and in the **update** part of the chunk select number. This tells our code that we'll be storing a number.



We're going to store **500** (half a second) in our **WaitTime** variable so enter 500:

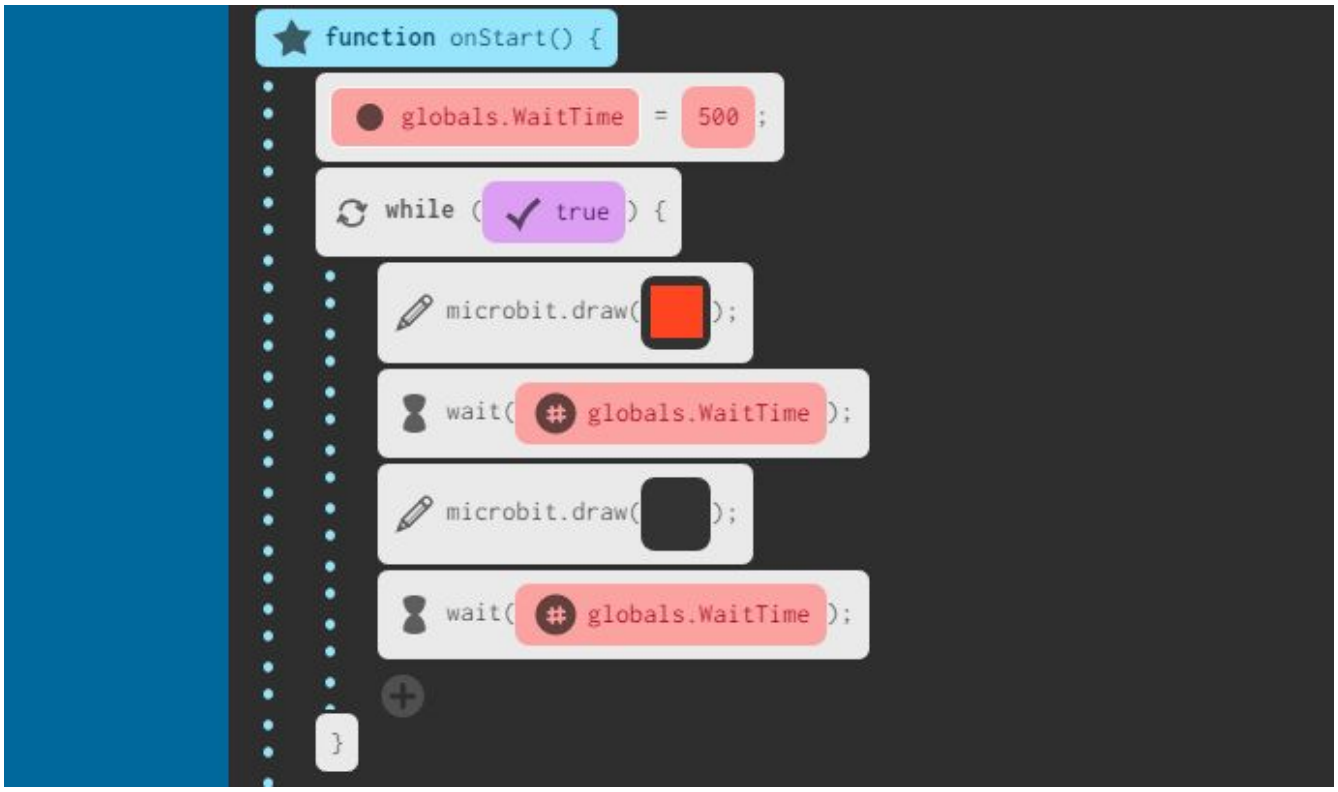


Now our variable **WaitTime** is set to **500** but we are not using it. So in the **wait** chunk after we've turned the LEDs on, click on the **number** (1000 in the picture above) and look at the bottom of the menu that appears, our variable **WaitTime** is available to select. Click on it to change the fixed number to our variable.



This means that instead of using 1000 as the time, the **wait** chunk will use whatever our variable **WaitTime** is set to (500).

Now change the second wait (after we've turned the LEDs off) to use your variable. Your code should look like this:



Run your code to check it does what you expect and that changing the value of **WaitTime** changes the speed of the LED flashing.

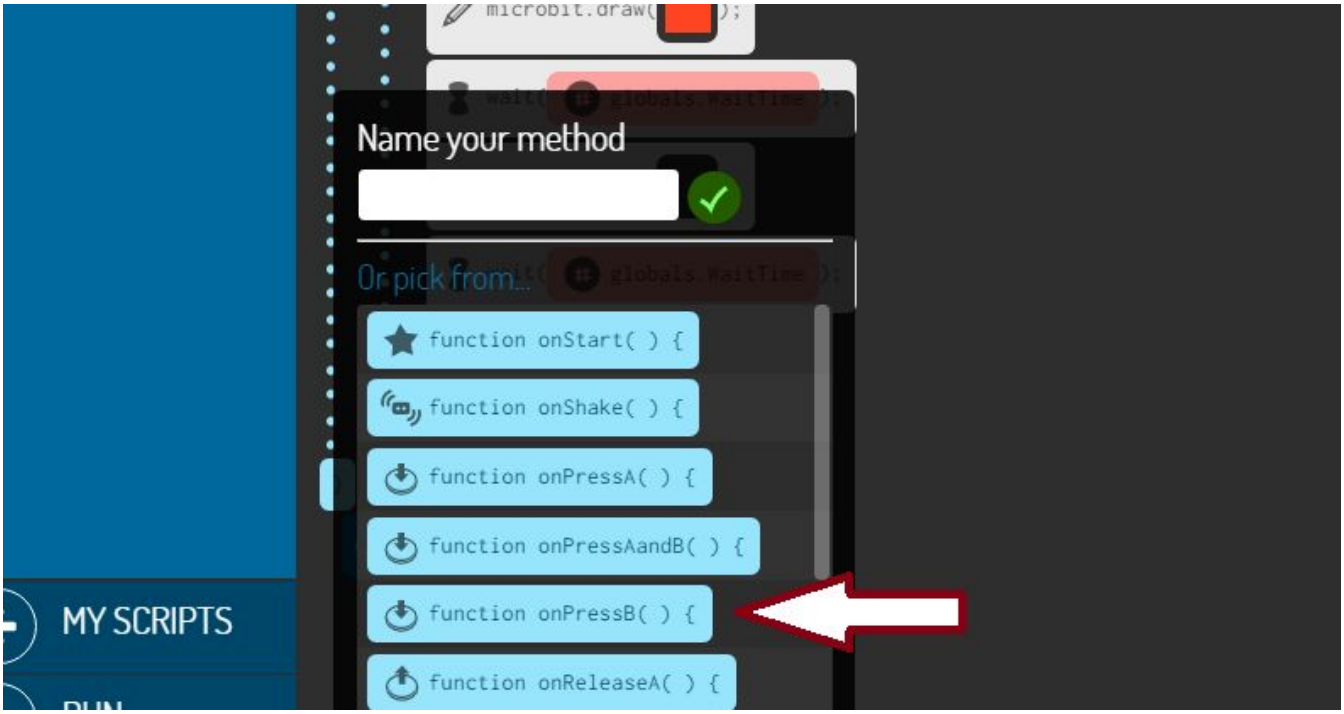
Step 7

Using events

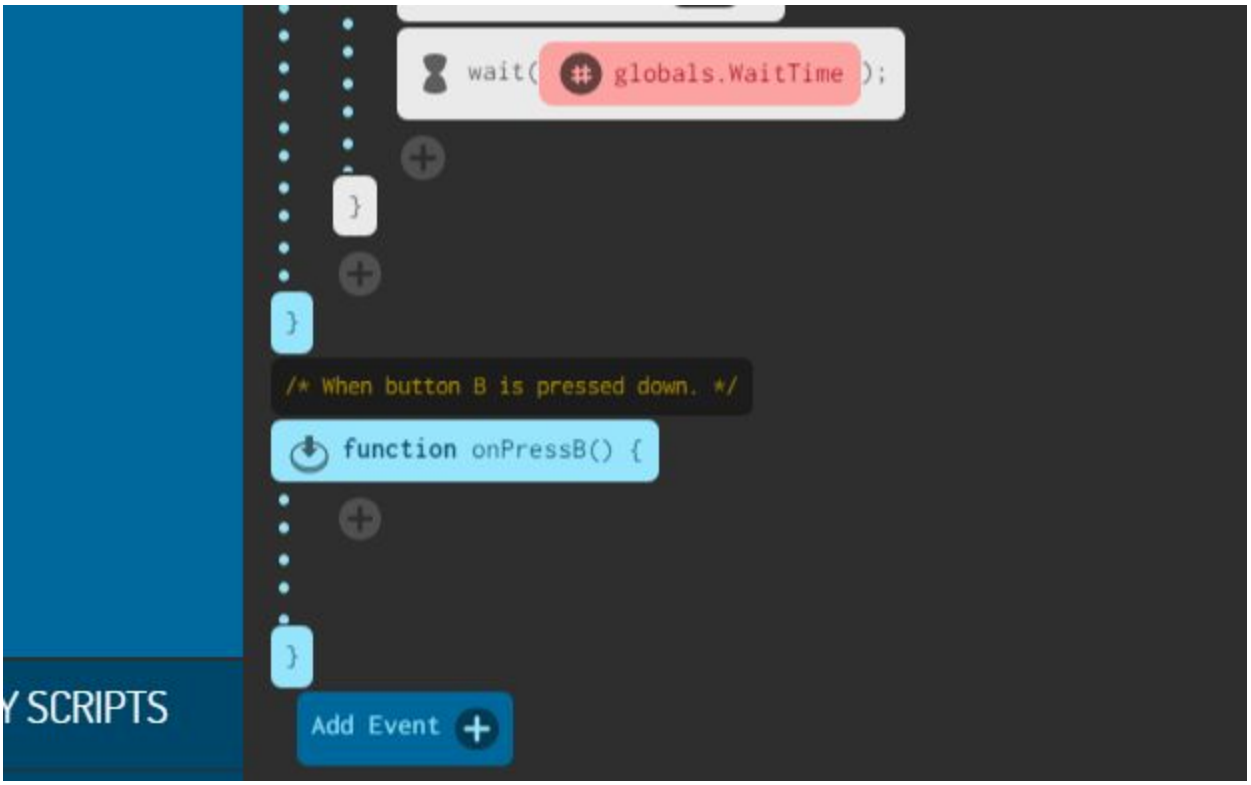
Now we're going to use the buttons on the Micro:bit to set the speed of the flashing. We can respond to things happening in the real world using **events**. We're going to slow down how fast the LEDs flash when someone presses the right hand side button. This button is called **button B**.

Click on the blue **Add Event** icon below your code.

We want to respond to the **button B** press so click on the **function onPressB( ) {**:



Your script should now look like this:

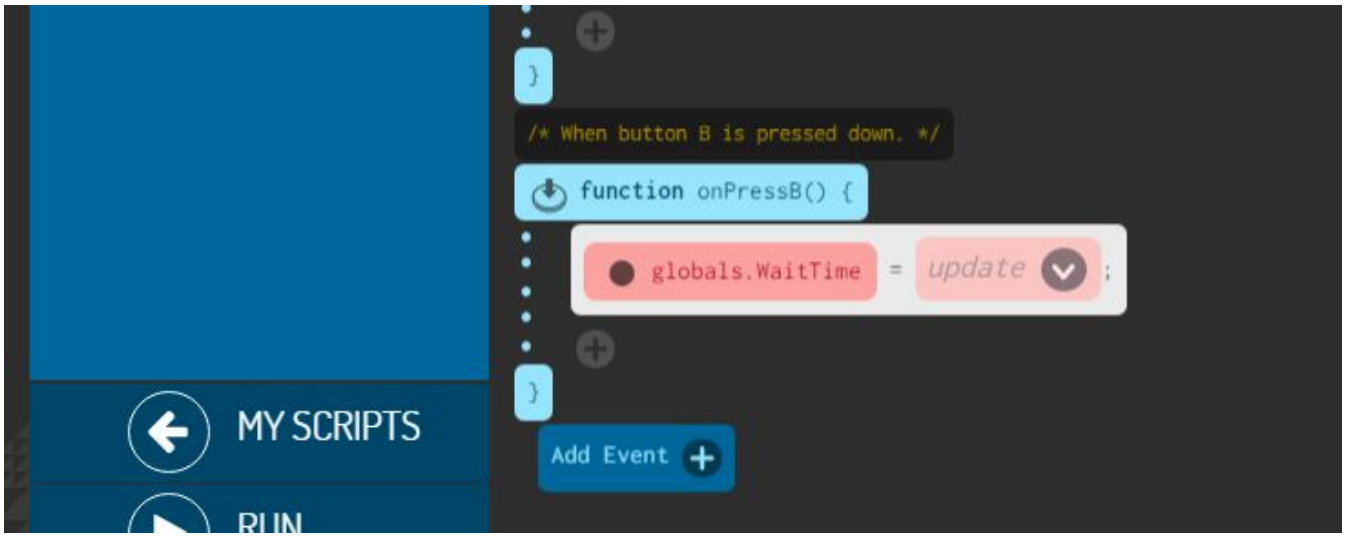


The Micro:bit has added a function onPressB(). Any code we put in this function will run whenever button B is pressed. Even better, it will run at the same time as our main function onStart() runs and flashes the LEDs.

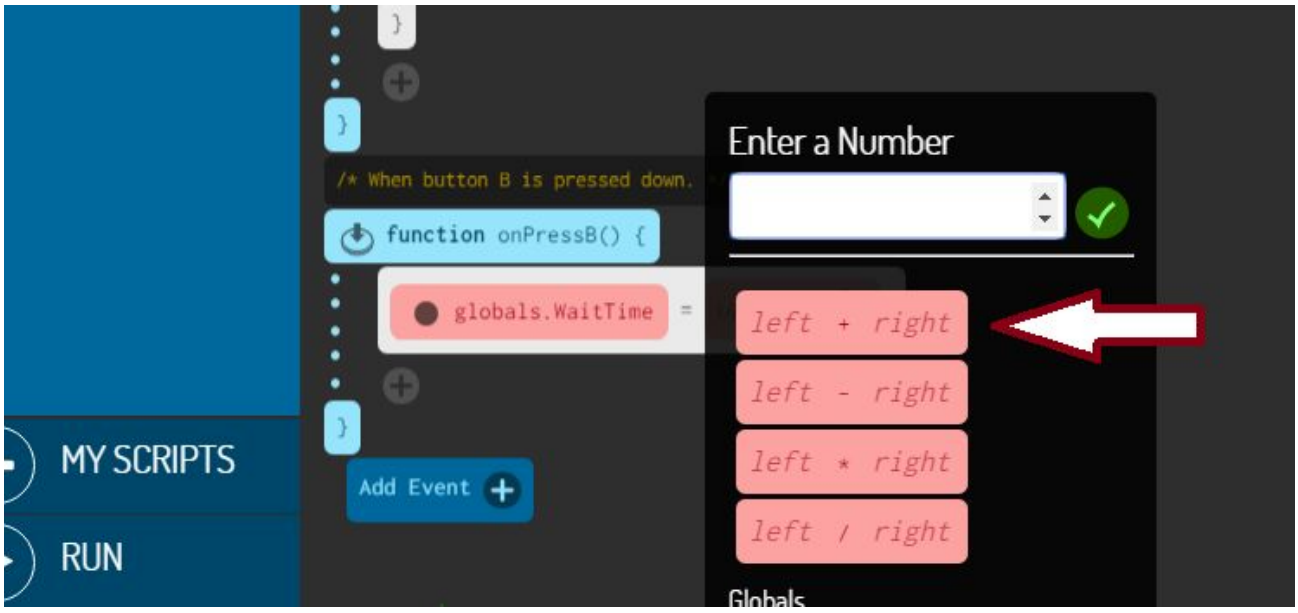
Step 8

We want to slow down the rate the LEDs flash when someone presses **button B**, so we need to give the variable **WaitTime** a larger value in the **onPressB()** function.

We need to go back to the **Globals** icon in the **Library** menu and drag **WaitTime** into the function **onPressB()**.



Now we need to update our variable using the **drop down menu** but this time we're not going to set it to a number, we're going to add to it. To do this, we select the **left + right** option from the **drop down menu**.



We now have two **drop down menus**. Lets add 250 milliseconds to **WaitTime** everytime the button is pressed. So we want to end up with **WaitTime=WaitTime + 250**.

Select the left hand **drop down menu** and then click on **WaitTime**. Then select the right hand **drop down menu** and enter **250**.

Your script should now look like this:

Now run your code. The LEDs will start flashing every half a second (500 milliseconds). Press **button B** (you can click on it on the simulator) and the speed they flash will slow down.

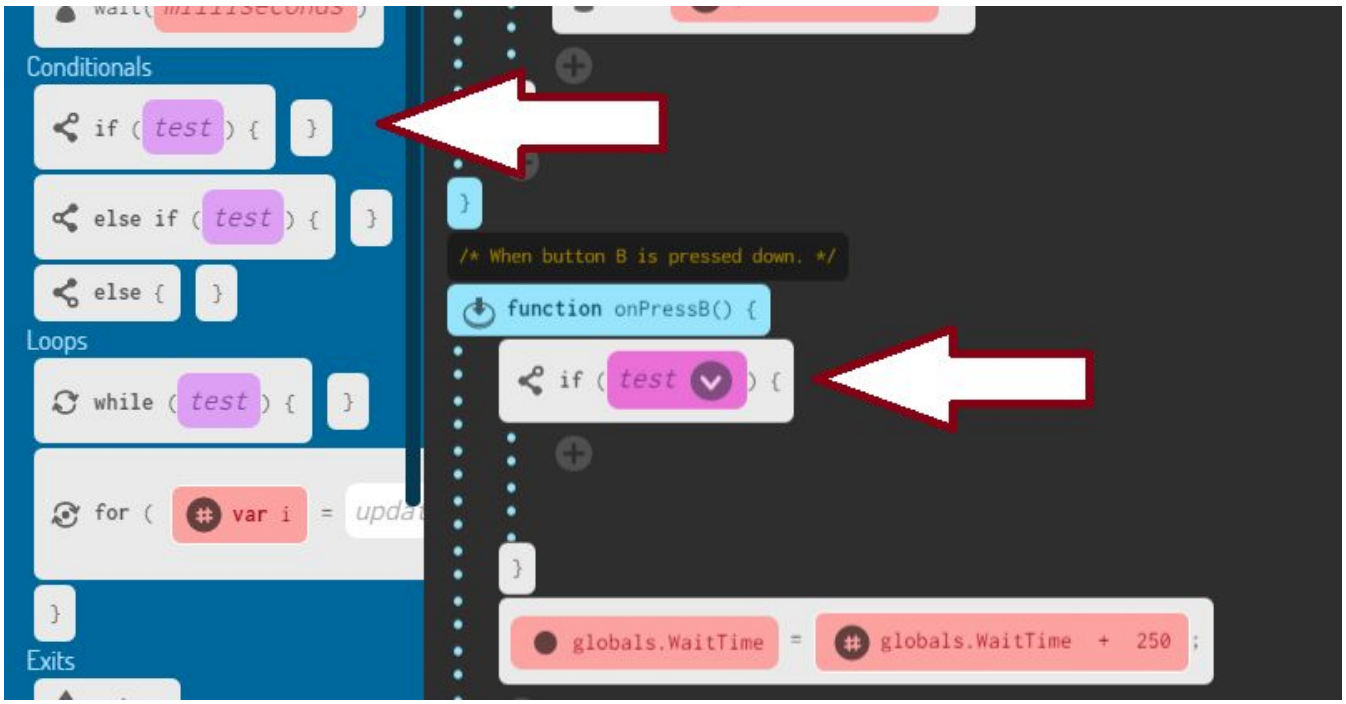
Step 9

Set a maximum time using conditionals

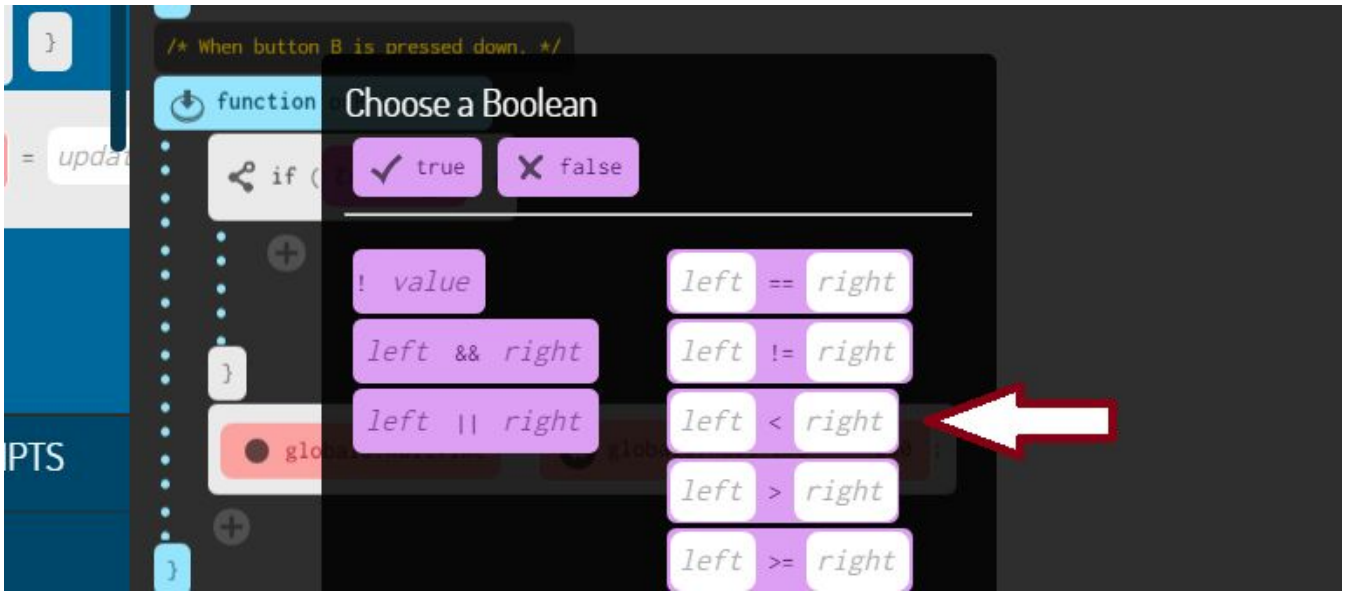
It would be nice if there was a maximum time between flashes, otherwise the delay between flashes could get so long that it looks like the code isn't running.

We can check that we are not going to make the delay too big before we change the value of **WaitTime**. Let's say if the value is already 5 seconds or more when the button is pressed, we won't make it any bigger and will ignore the press.

To do this, we'll use the **if** chunk from the **Language** menu. As we need to check the current value of **WaitTime** before we add to it, we need to put it before the chunk that adds 250 to **WaitTime**:

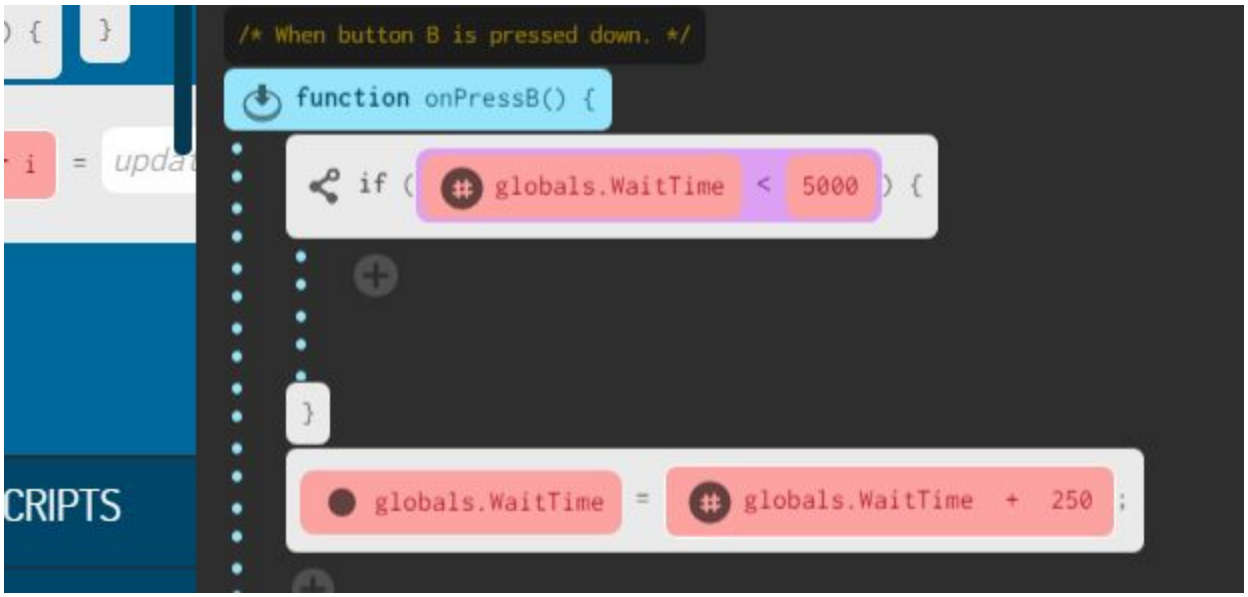


We only want to add to **WaitTime** if it's smaller than 5000 milliseconds, so on the **drop down menu** select the **left < right**:



In the left **drop down menu** we need to select **WaitTime** and in the right hand box we need a number (in this case enter 5000).

Your script should now look like this:



We need to drag our **WaitTime = WaitTime + 250** into the **if** section otherwise the add will happen anyway.

Now run your code. You will find that no matter how many times you press the B button, the time between flashes will not get longer than 5 seconds.

Additional projects

Try adding another event that makes the flashing faster when button A is pressed. You shouldn't let it get faster than 250 milliseconds.