I hope you are finding the Lessons interesting. David Whale designed them all. He’s also written the factoids in the micro:bit Wizard column. The micro:bit wizard is a nickname he was given when he worked at micro:bit.

Open Source developers often use a “nic” or nickname. They go by it when they are contributing code. If you join a community you may well hear people referred to by their nic’s not their real name.

I used to work with an elmo and an agy. In agy’s case it was his initials that made up his nic. Mine is brocka, you will find me on GitHub under that nic or username.

I always find that when I am doing something I enjoy, like when I am on the first half of a 10-day holiday then the first half is longer than the second. The first 5 days seem longer than the second 5.

That’s not logical is it…

Most things about coding are very logical. Are you finding that? Logic and experience tend to help coders to work out solutions when things go wrong.

But it’s the way it feels!

Amanda Brock is CEO at OpenUK
What other applications could the MiniMU glove be used for?

Guy, Perthshire

Any application where gestures and sound work well together would be a good application for the MiniMU glove – for example, detecting hand movements and turning these into speech or any form of musical performance that is gesture-based. You could add flex sensors and also detect finger movements, it might also be really handy in sporting applications such as an automatic tennis coach that analyses your serve performance. Your imagination is the only limit!

Why when pins touch together does my micro:bit play the same note over and over again?

Adonai, Essex

If the 3V and 0V pins touch together, this will cause the micro:bit to reset and it will re-run your program from the start again. If you only play one note, check that your crocodile clips on 3V and 0V are not shorting together or shorting against the little pins in between, as this will cause it to repeatedly reset and then play the first note of your tune, reset again, etc. Always make sure to attach the crocodile clips at 90 degrees to the board to get a good connection.

The Irish politician Edna Kennedy stated that “Learning to code is learning to create and innovate”. Let’s consider some new code learning to improve our creations and inventions.

**Functions** are self-contained blocks of code that accomplish a certain task. They normally take in data, process it, and return a result. **Functions** are first defined meaning they are created and then called which means the code is run.

Once written, functions can be reused in lots of places in a program. They can even be called from the inside of other functions. When finished a function returns to the point in the program that it was called from.

Some functions and code make use of constants which are variables which store a number that doesn’t change. We indicate constants by giving them uppercase variable names as a reminder not to change their value.

We have learned a lot about selection but how do If statements work? They evaluate an expression such as running time > alarm time to decide if it is True or False. If the expression is True, then the code is run and if it is false then the code is not run.

By now, you should have some confidence about innovating some ideas and making some creations of your own.

When you write your own code remember to include a licence. This legal document lets others use the copyright in the code which belongs to you the creator of the code and gives other developers details of the permission that they have to reuse your code.

Keep Coding!

David Whale is a Software Engineer

Pamela Boal is an Educationalist
Open Source projects can be very large with millions of lines of code. Most of the lines are simple with pre-known coding patterns. However, the size of the projects can make the collaboration among developers difficult. Misunderstandings between developers cause program errors – these are bugs!

As new developers, we want to directly contribute to a project by writing new features. The problem is that this can be difficult to do. We need to understand the whole structure of the program.

Fortunately, there is an easy way to start: bug fixes!

In Lesson 5, you defined a function. It took a specific input (two variables), performed a defined task (compare variable), and returned a specific output (sound depending on the check).

We do not need to understand the whole program to check if this function is correct, we can study each function separately!

Open Source projects are divided with many different functions. So, for every bug identified, we can easily start checking if each function is behaving as expected.

This approach has two benefits:

Firstly, you help the project by fixing a problem and people will recognise you for doing that!

Secondly, you learn more about the project which is necessary to become a senior contributor!

I’m the founder of the Tika Open Source project. Tika is a content detection and analysis toolkit that extracts text, metadata and language information from any type of file including 1000s of files that are present on the Internet.

If you’ve ever done a search on the Internet, chances are Tika helped the search engine to index information about the results.

Tika also helped to expose financial fraud amongst the world’s elite, and was the key technology that helped the world’s journalists to analyse the Panama Papers which won the 2017 Pulitzer Prize in Exploratory Reporting.

At NASA’s Jet Propulsion Laboratory, I manage the Artificial Intelligence, analytics and custom development activities for the laboratory.

We are using Tika, Python, TensorFlow and other Open Source Software to help to put Rovers on Mars; to make crucial measurements of the Earth’s signals like temperature, CO2, or cloud cover; and to explore interstellar space with Voyager launched over 40 years ago.

Not only are we using Open Source, we are contributing to it daily.

Open Source projects can be very large with millions of lines of code. Most of the lines are simple with pre-known coding patterns. However, the size of the projects can make the collaboration among developers difficult. Misunderstandings between developers cause program errors – these are bugs!

As new developers, we want to directly contribute to a project by writing new features. The problem is that this can be difficult to do. We need to understand the whole structure of the program.

Fortunately, there is an easy way to start: bug fixes!

In Lesson 5, you defined a function. It took a specific input (two variables), performed a defined task (compare variable), and returned a specific output (sound depending on the check).

We do not need to understand the whole program to check if this function is correct, we can study each function separately!

Open Source projects are divided with many different functions. So, for every bug identified, we can easily start checking if each function is behaving as expected.

This approach has two benefits:

Firstly, you help the project by fixing a problem and people will recognise you for doing that!

Secondly, you learn more about the project which is necessary to become a senior contributor!

I’m the founder of the Tika Open Source project. Tika is a content detection and analysis toolkit that extracts text, metadata and language information from any type of file including 1000s of files that are present on the Internet.

If you’ve ever done a search on the Internet, chances are Tika helped the search engine to index information about the results.

Tika also helped to expose financial fraud amongst the world’s elite, and was the key technology that helped the world’s journalists to analyse the Panama Papers which won the 2017 Pulitzer Prize in Exploratory Reporting.

At NASA’s Jet Propulsion Laboratory, I manage the Artificial Intelligence, analytics and custom development activities for the laboratory.

We are using Tika, Python, TensorFlow and other Open Source Software to help to put Rovers on Mars; to make crucial measurements of the Earth’s signals like temperature, CO2, or cloud cover; and to explore interstellar space with Voyager launched over 40 years ago.

Not only are we using Open Source, we are contributing to it daily.

Chris Mattmann is Division Manager of the Artificial Intelligence, Analytics and Innovative Development Division at NASA’s Jet Propulsion Laboratory.

Chris Mattmann is Division Manager of the Artificial Intelligence, Analytics and Innovative Development Division at NASA’s Jet Propulsion Laboratory.
I used to love sugary drinks when I was younger. Their sweet taste and variety of flavours would always make me want to drink them over water or tea.

But one day, in my early twenties, I read my favourite soft drink’s label. I was so shocked with the level of sugar and unpronounceable additives in that bottle that I decided to stop drinking soda altogether!

I don’t expect you to stop too, but you should definitely try to develop a love for water and definitely drink more of it.

Why? It’s crucial for our bodies’ performance.

Making sure you stay hydrated is the number one rule in health and nutrition. Did you know that humans are made up of 70% water? Our blood, sweat and digestive fluids are mostly made of water. This is why the body can go weeks without food, and only a few days without water.

Water also helps your mood and memory.

Research shows that even mild dehydration can negatively affect your memory and mood. It causes you to be upset, tired, and unable to concentrate.

On the flip side, the sugar in soft drinks is terrible. It causes cavities, unhealthy fats, and a number of diseases and conditions including diabetes, even for young people.

I hope that I have inspired you to drink more water. If so, recommended amounts are between 1.3 and 1.5 litres each day for girls aged 9-13, and 1.5-1.7 litres a day for boys aged 9-13.

For those of you aged between 14 and 18 (both boys and girls), the recommended water intake is 1.9 to 2.6 litres a day.

Matthew Springer is a Founder
Across
2. What is it called when you have formal permission to use, perform or release something?
5. What is a person who writes computer programs called?
7. What measures proper acceleration?
9. What is a collection of instructions that can be executed by a computer to perform a specific task called?
11. What is it called when your micro:bit remembers how long it has been since it was last reset? (Two words, 5 letters, 1 punctuation mark, 3 letters)
12. What does EULA stand for? (Four words, 3, 4, 7 and 9 letters)
14. What is it called when you keep an eye on the progress of something?

Down
1. What does OSI stand for? (Three words, 4, 6 and 10 letters)
3. What is it called when you replace water in the body?
4. What is a licence called that has been authorised by the OSI? (Two words, 8 and 7 letters)
6. What is an input device that records data about the physical environment around it called?
8. What is it called when you check something is working properly?
10. What is it called when a program, code or software is working smoothly?
13. In computing what is it called when you store a value that doesn't change?

To enter the prize draw you must submit the completed Crossword and Word Puzzle from Ezine1, by email to ezine6@openuk.uk. All entries are subject to our terms and conditions which you can read https://openuk.uk/ezine-6-comp-terms-and-conditions. By entering you agree to them and confirm that you have parental or guardian permission if you are under 16 years of age. One winner will be drawn from completed entries received by 10 September. No cash alternative. UK residents only. Judges decision is final. No correspondence will be entered into. Surname and county of prize winners will be made available on request. Promoter, OpenUK.
Adam’s Mum keeps nagging him to drink more water so that he doesn’t get dehydrated in the hot weather, and this gave him the idea of using his MiniMU glove as a reminder system. Instead of his Mum reminding him to drink more water, his glove could sense when he lifts his bottle to drink, and then start a timer. When the timer runs out, his glove could beep to remind him that it is time to drink more water – it would certainly be much more fun than his Mum nagging him all the time! Adam is a bit stuck with the coding though, can you help him out?

In today’s lesson you will help Adam to design a program for the ‘Water Glove’ project. By doing this you will learn about:

- event handlers;
- variables, including boolean variables;
- if statements;
- functions;
- and constants.

You will also learn how to measure elapsed time on the micro:bit, as well as making a more complex gesture out of two simple gestures.

You will need:

- your assembled MiniMU glove;
- the MakeCode web coding editor;
- a water bottle with some water in it.
Figure 1: The MakeCode web coding editor.

Figure 2: The OnStart event handler, and 'start' function.

Figure 3: Forever loop – this sounds the alarm.

Figure 4: Design for the 'drink' gesture.
Figure 5: OnTiltRight and OnScreenUp gestures (drink gesture).

Figure 6: Copyright, Licence (EULA) and Attribution.
Additional Information – adding comments to MakeCode

To add a comment to a MakeCode program, right-click with the mouse pointer on any block. From the pop-up menu, choose 'Add Comment', and it adds a yellow note attached to the block.

You can then type in any text you want in that comment.

If you click on the small triangle in the top left corner of the comment, it will hide it.

You can show the comment again by clicking on the speech bubble icon in the top left corner of the block.

To delete a comment, click on the dustbin icon in the top right corner of the comment.