

## High level code and machine code | Part B

We could program directly using binary code. Every instruction that the CPU can understand will be represented by unique code called the opcode. So we'd need to know the opcode for every instruction that we want to use in our program. The difficulty is we need to learn the code, and we also need to enter the correct binary codes for the data values to. Programming in binary is called machine code.

As computers became more complex, this wasn't the most practical way of writing programs. Every CPU supports an instruction set that it recognises. It can be programmed using simple character sequences to represent the instructions called mnemonics. This is assembly language. A good way to learn about this is to use the Little Man Computer simulator. The Raspberry Pi has an ARM CPU, which has its own assembly language. You can find out more about that from the internet. If you wanted to add two numbers together, we need to use the instruction ADD, and then give labels to the data that we want to add. I have an example here on the board, and I have used some code that you could use in the Little Man Computer simulator to add two numbers together. What this does is allow 1 number to be input, it's then stored in a memory location, a second number is then input and the first number that we've stored is added to it. We then output the results, and then stop the program. Assembly language programming is fine for simple programs, but if I want to write a complex program, such as a game, it would take us a very long time and involve many lines of code.

High level programming languages have built in commands that allow us to write code that is easier for humans to understand and carry out useful operations in just a few lines. It is important to spell the commands correctly, and to input them in the correct order in the correct way. This is called the syntax of the language. Like learning any new language, some are easy to get to grips with, and hard to make mistakes with, but others are a little bit more tricky.