Electronics

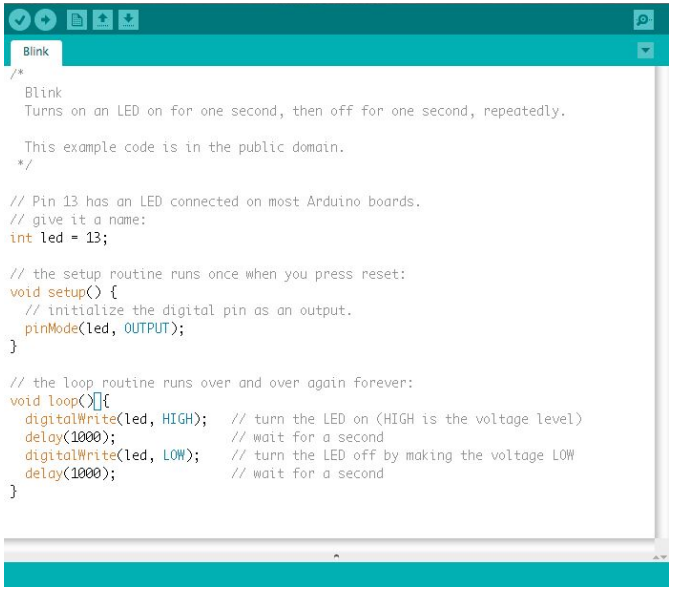
Programming 101

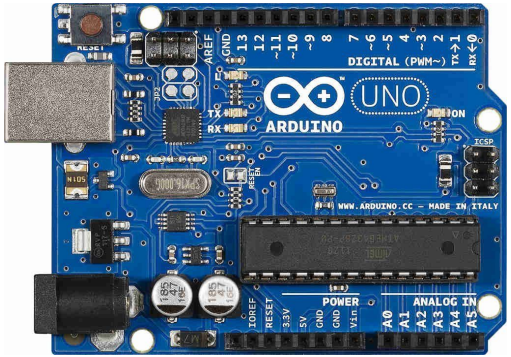
Programming 101>> Software vs hardware

software | hardware

The cables, the loads and the breadboard are our hardware.

The arduino board is also hardware, but it is the computer controlling the hardware. We tell the arduino how to power and control the circuits using a computer programme, which is a software.





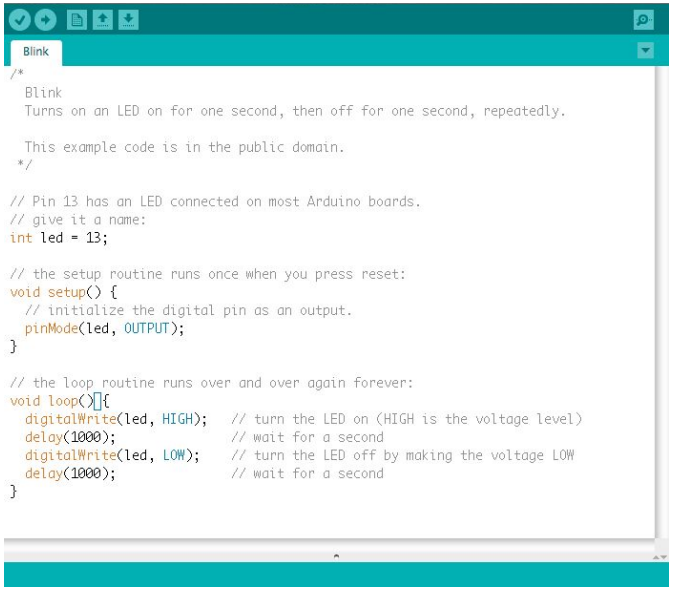
Programming 101>> Software vs hardware

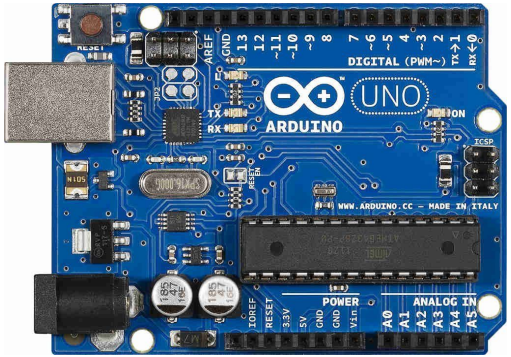
software | hardware

Programmes are the way we tell computers what we want them to do and how they should behave.

There are different programming languages, and like spoken languages each has its own rules, and there are different ways to tell the computer to do things.

Gcode, Python etc



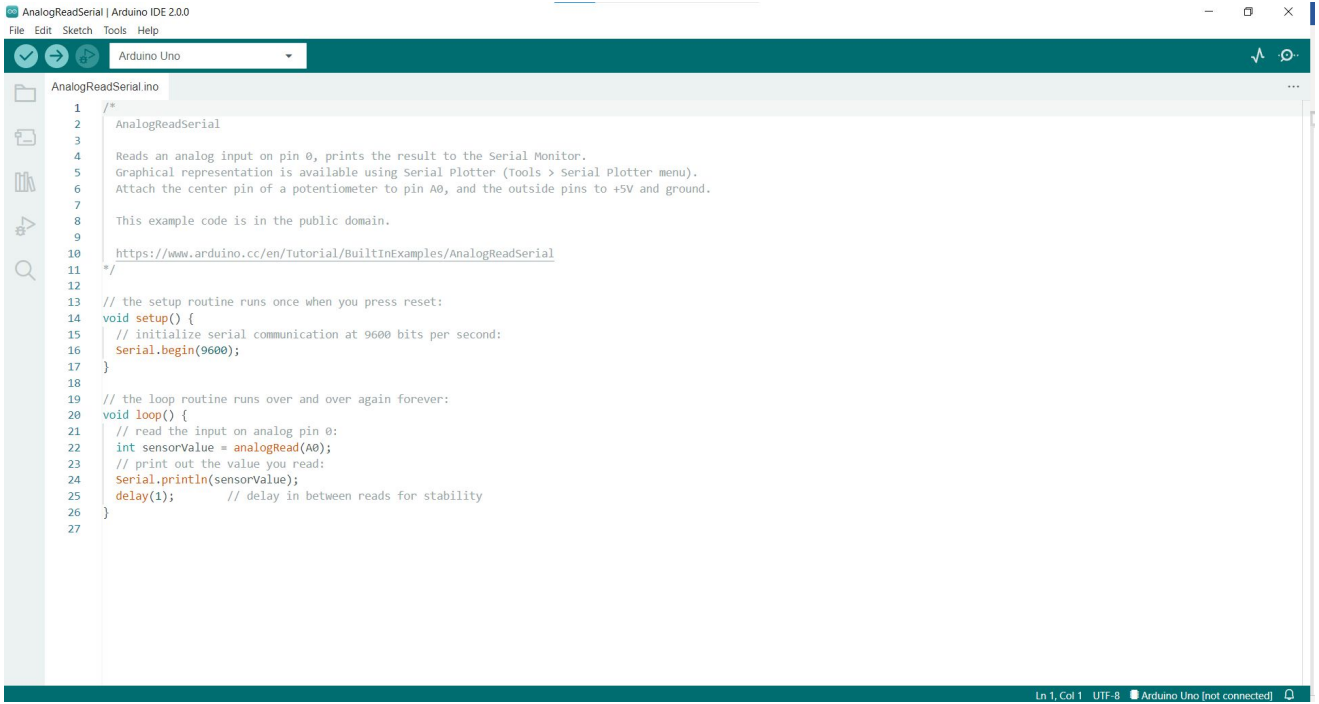


Programming 101>> Arduino IDE

The Arduino IDE (integrated development environment) is an open-source platform where we write, test and upload programmes to the board.

It can be downloaded here: https://www.arduino.cc/en/software

It uses a programming language called C++.



Programming 101>> Code structure

A programme is made of three parts:

Definitions

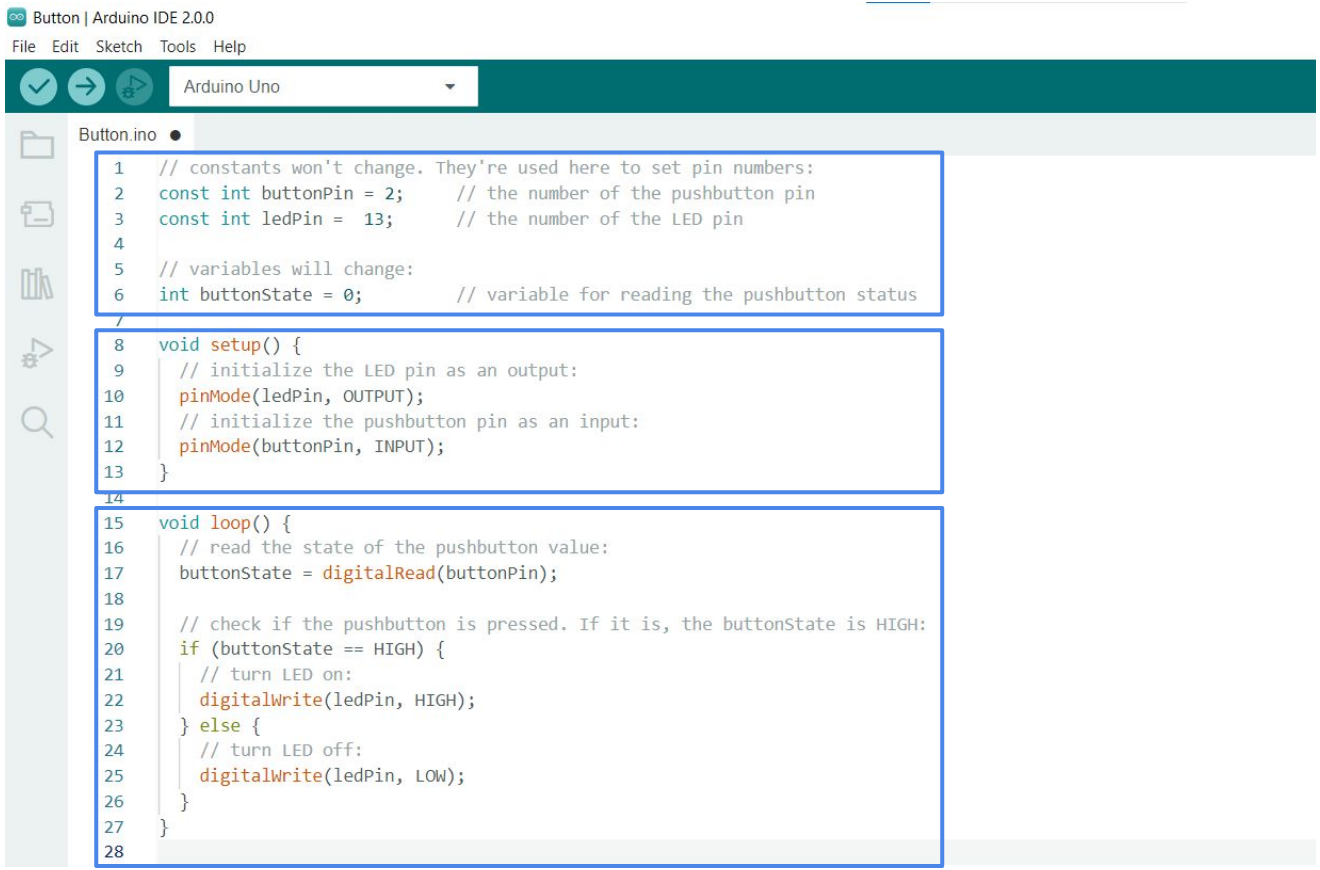
This is where we set variables, create objects and import libraries

Setup()

Is code that we run once at the beginning of each programme. Setting the pin modes, starting serial monitor etc

Loop()

The main part of the code, that runs forever



Programming 101>> Functions

Functions are used to tell the board to do something.

For example:
delay() is used to tell the programme to wait before it does something

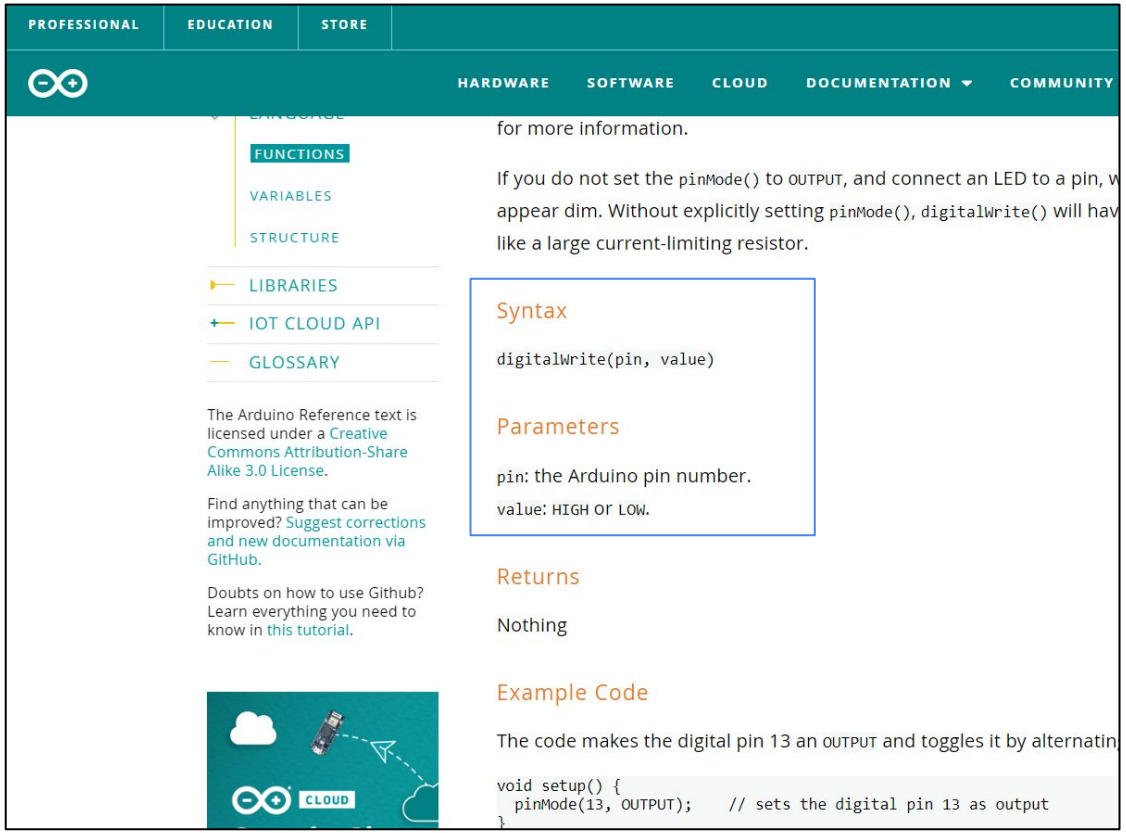
digitalWrite() is used to tell a digital pin to send a current, or not to.

Programming 101>> Functions

Each function has specific syntax, which determines what information we need to give it in order to execute the operation. The digitalWrite() function will need to know what pin we want to control, and whether we want to send current or not.

The correct syntax is:
digitalWrite(pin, value)

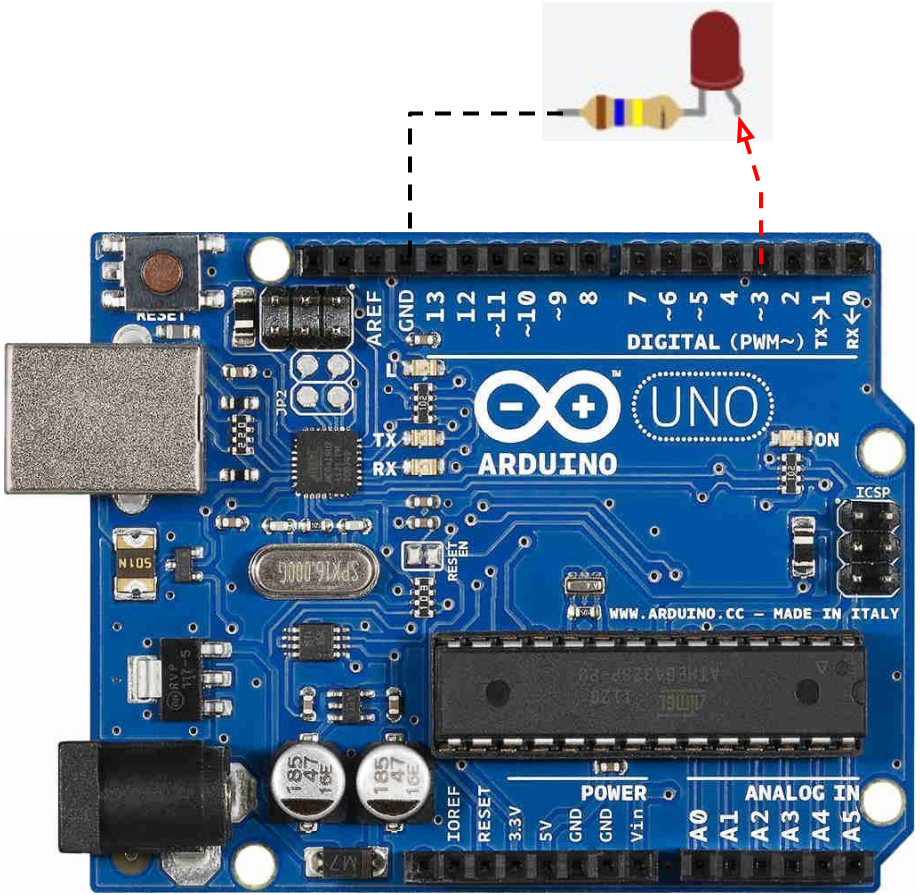
The syntax for each function can be found on the arduino website:
https://www.arduino.cc/reference/en/



Programming 101>> Functions

So if we write:
digitalWrite(3, HIGH)
The programme will set digital pin number 3 to HIGH, which means it will send 5V.

We can use the same function to then turn it off, by writing:
digitalWrite(3, LOW)



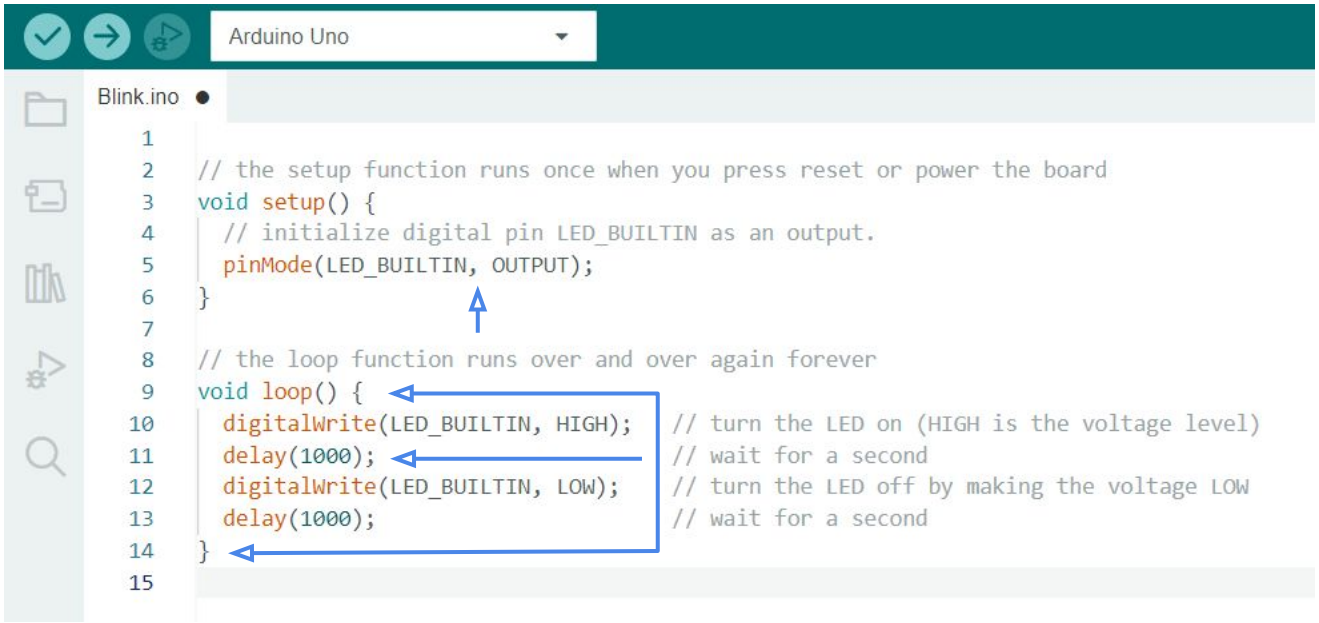
Programming 101>> Syntax

The programme itself has syntax as well, which if we don't write properly will prevent it from running.

At the end of every line of code there needs to be a semicolon ';'

The parameters of a function are put inside normal brackets '( )', separated by ','

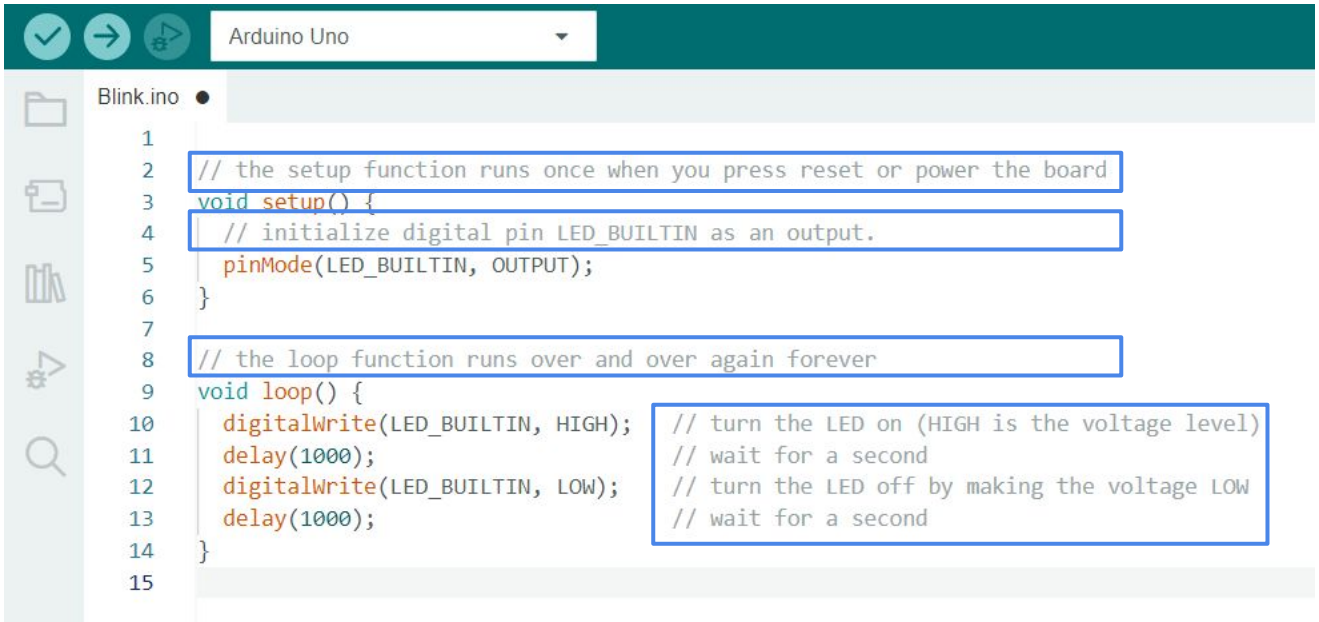
Blocks of code are organised inside curly brackets '{ }'



Programming 101>> Syntax

Anything written after a '//' is ignored by the programme, and is for you or anyone else who reads the code in the future.

Codes can be complicated, so it's always a good idea to make sure they are organised and have comments along the way to explain what each part of the code is doing.



Programming 101>> Variables and Constants

Variables and Constants are used to store data inside of them.

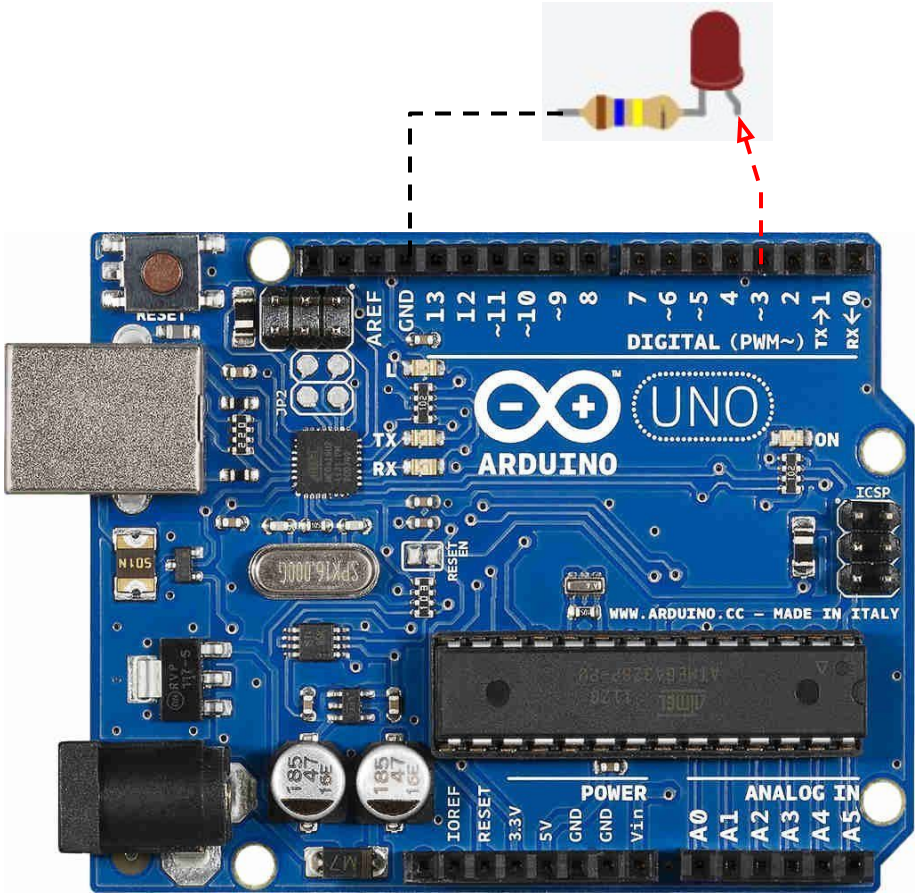
Variables will be used to store data that changes, and we can name our own to make the code more readable.

Constants will have the same value for the entire programme. Some of these constants are already known to the programme.
For example: HIGH and LOW.

Programming 101>> Variables and Constants

For example,I can declare a constant Red_LED_Pin, which I'll use throughout the programme instead of writing the number 3. This makes the code more readable, especially when there are several pins in use for different loads.

This also allows flexibility, because if I need to connect the LED to a different pin,I only have to change the pin number once where I declared the constant's value.



Programming 101>> Variables and Constants

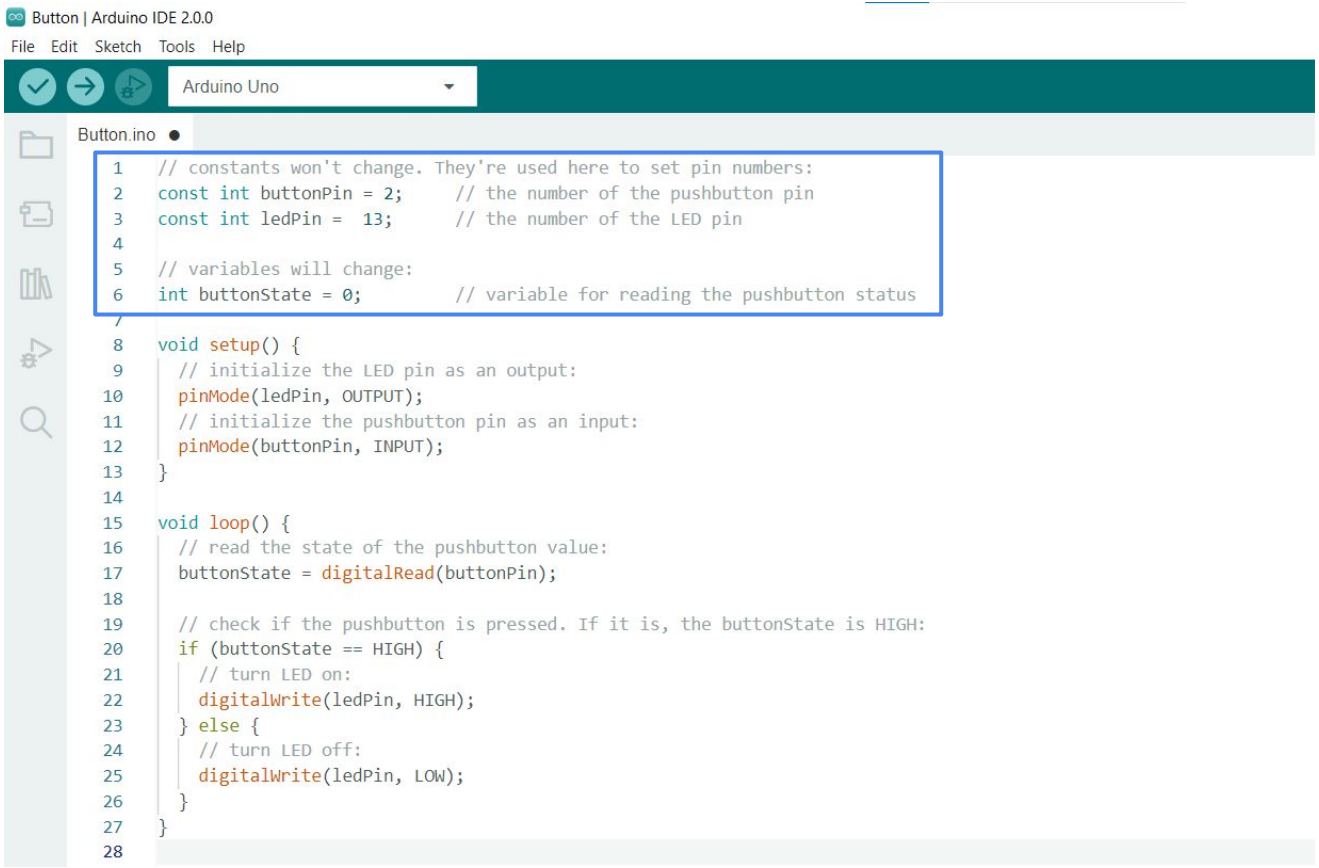
There are two ways to declare a constant, each with different syntax:

```
#define  Red_LED_Pin 3
```

or

```
const int Red_Led_Pin = 4;
```

The second option is preferred, so we'll use that one.
The declaration goes in the first part of the code.



Programming 101>> Data types

Each variable or constant has a type of information they are allowed to store.

In this case we're declaring a constant to hold a pin number. The number of a pin will always be a 'whole number', which is called integer, int for short:

```
const int Red_Led_Pin = 4;
```

Programming 101>> Data types

A constant could also hold a number with a decimal point, a float:

```
const float distance = 1.8;
```

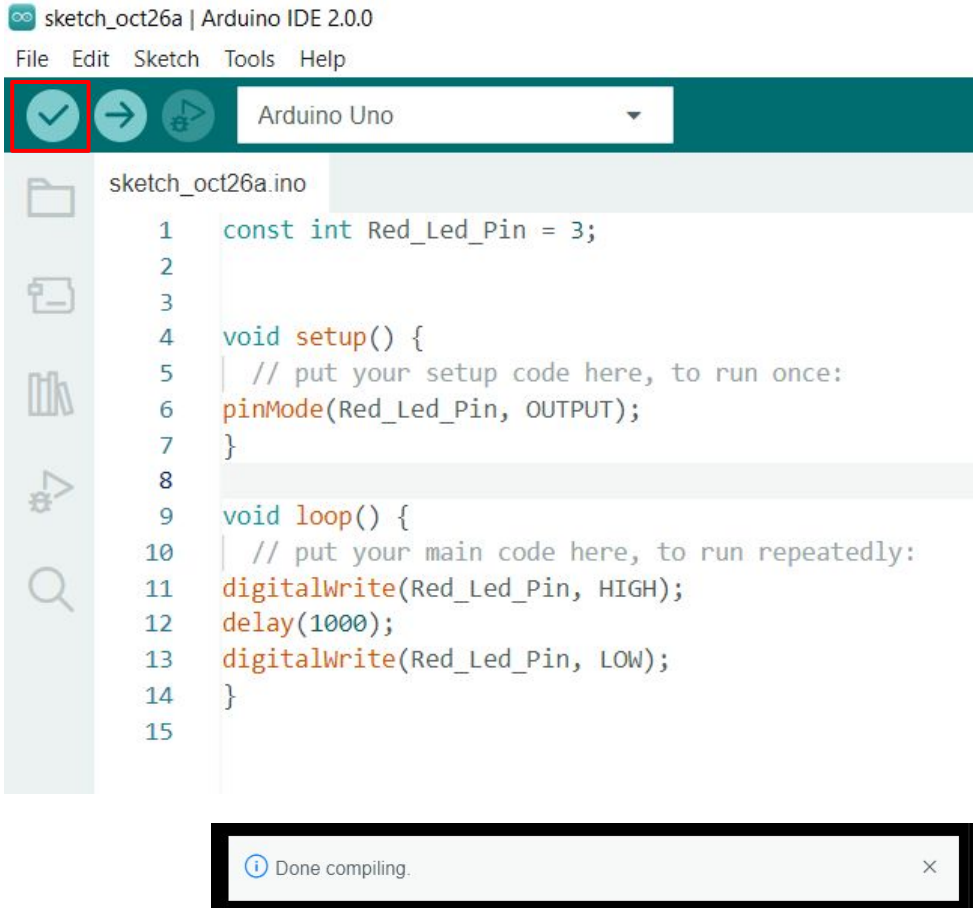
Another data type is a string, which means text.
For example, HIGH and LOW which we use to set pin modes, are string constants the are saved for the programme to use.

Programming 101>> Compiling and uploading

Once we've written a programme we need to make sure that it's written correctly, that we spelled things properly etc.

The verify button will run through the code and make sure it makes sense.

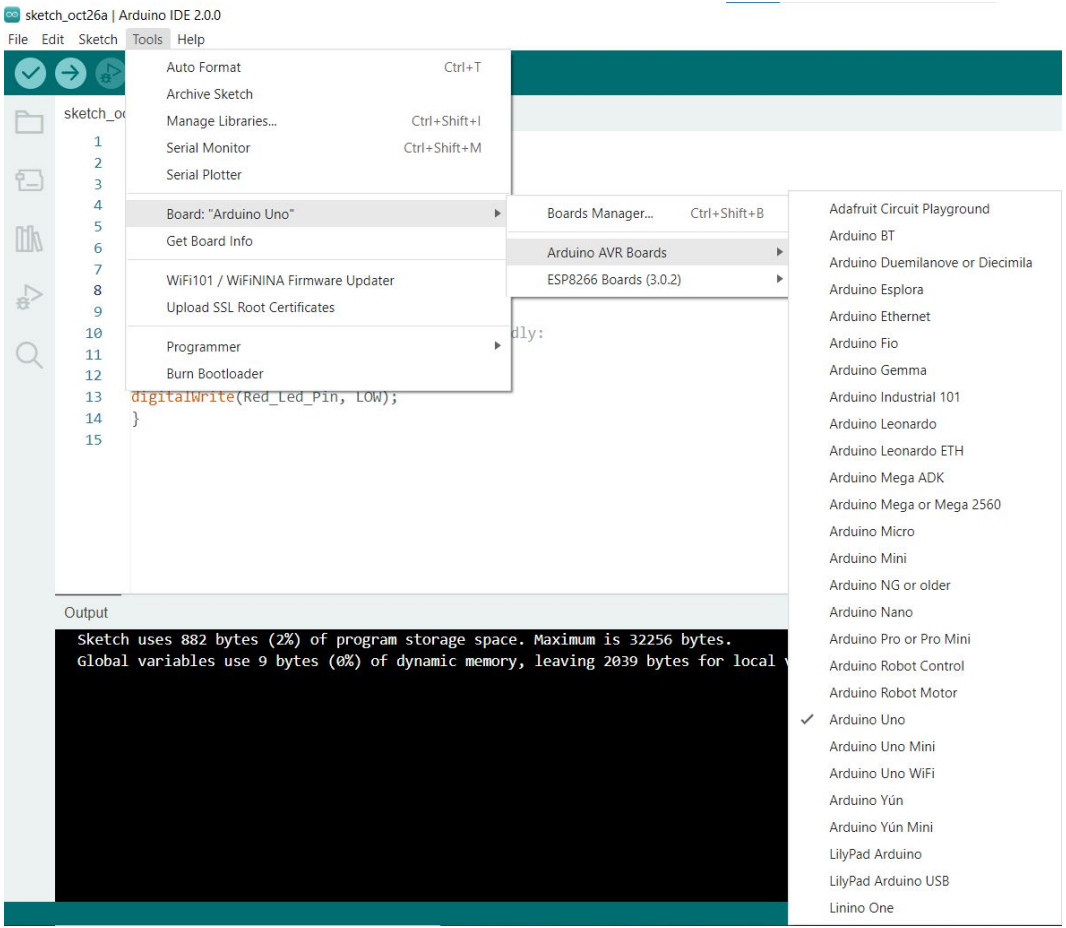
If everything is okay it will say: Done compiling.



Programming 101>> Compiling and uploading

We then need to upload the programme onto the Arduino board.

For that we need to first choose which board we are uploading to:
Tools> Board> Arduino AVR Boards> Arduino Uno



Programming 101>> Compiling and uploading

Then tell the IDE which USB port the Arduino board is connected to.