**Lab 1: Working with Visual Studio Code**

The aim of this lab is to use Visual Studio Code and to run a simple JavaScript program.

**Step 1: Start up Visual Studio Code**

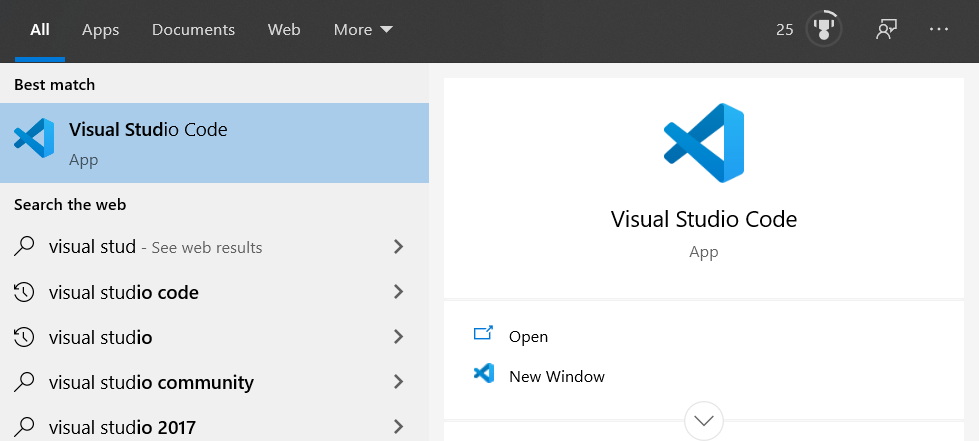
To do this look on your desktop as you may have a short cut to visual Studio configured for you. It will look like this:



Double Click on this icon – this will start the Visual Studio Code (aka VSC) running.

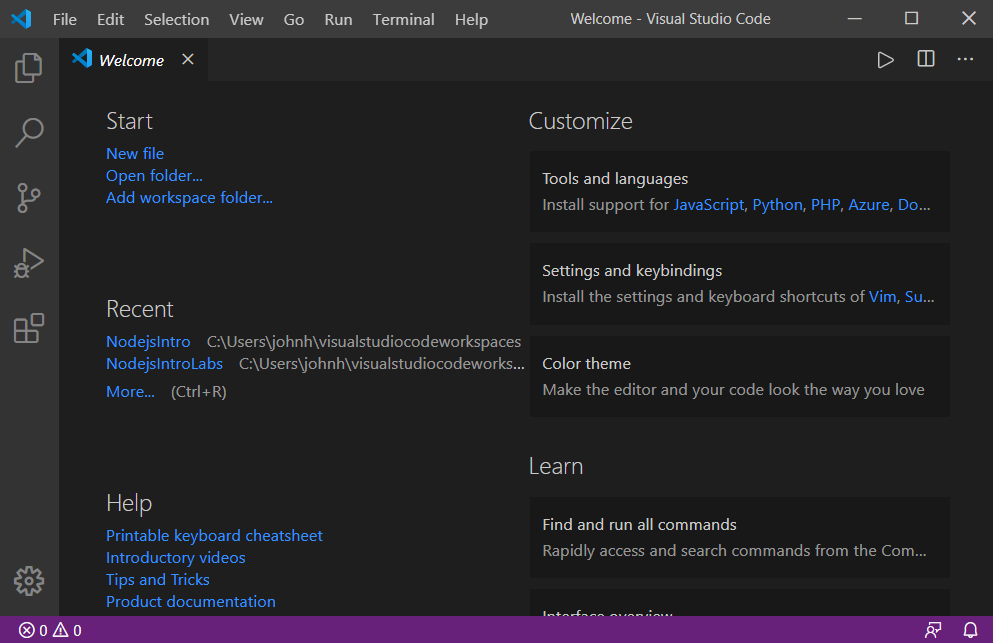
This is a lightweight, yet very powerful, IDE (integrated Development Environment / Editor) that you can use to create JavaScript applications. As an aside it is itself mostly written using JavaScript.

If you do not have a short cut icon such as the above on your desktop then go to the search bar at the bottom of the screen and start to type in Visual Studio Code. Windows should find this application before you finish typing the whole name and should make it available as an option in the right hand side area of the search results as shown below. Select to ‘open’.



If this is the first time that Visual Studio Code is started then it will display an empty editor.

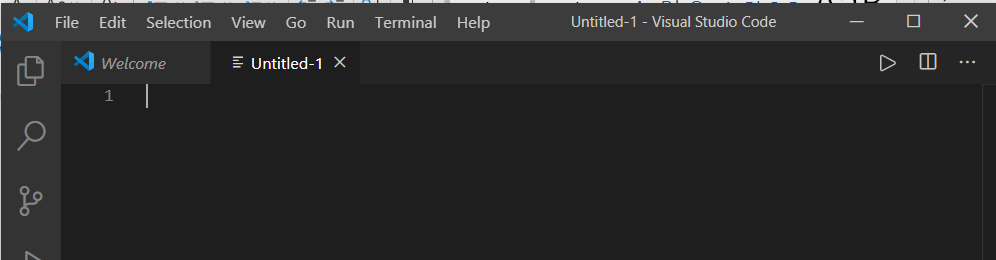
This will look similar to that shown below:



This can be a little off-putting but it is really trying to help you providing a range of options.

As this is the start of the work we will be doing we will select the ‘New file’ option which is shown in blue on the left hand side of the main window under the heading ‘Start’.

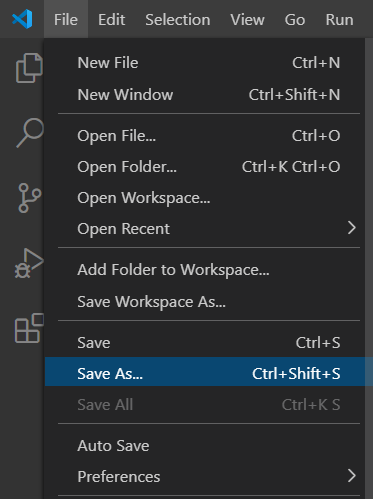
This will open an empty (currently unsaved file) within the editor as shown below:



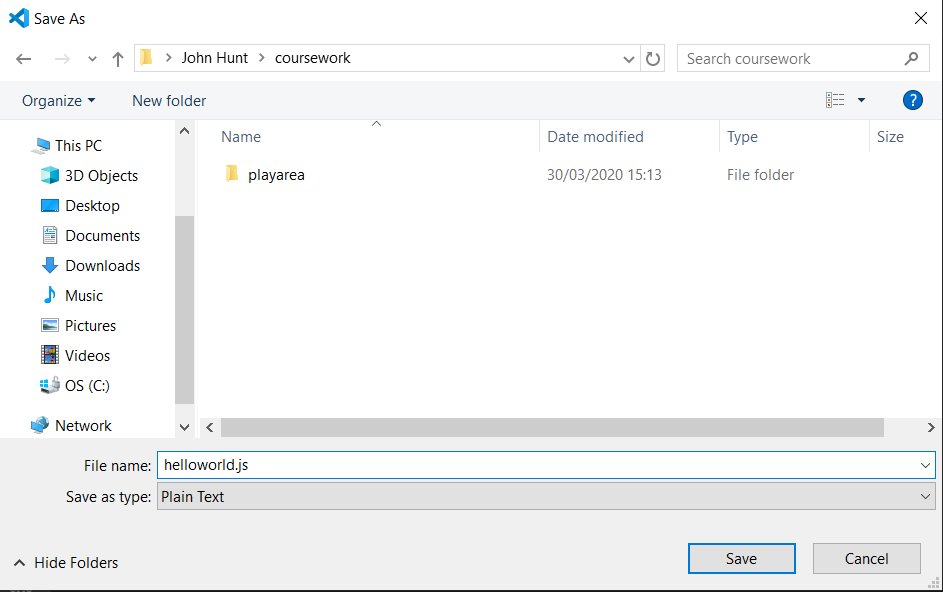
We would like Visual Studio Code to know that we are working in JavaScript and colour code our text appropriate (to highlight syntax words etc) and to check that what we are typing looks valid.

To do this we need to save the file with a .js extension; for example using the filename ‘helloworld.js’. In computer programming languages it is common to have a file extension which indicates to editors and compilers that the contents of a particular file is written in a particular programming language. For example if we were writing a Python program we would use a .py extension; here we are writing JavaScript and thus we use a .js extension.

To do this go to the File menu and select the Save As.. option as shown below:



Find your coursework directory from last week and save the file in there under the name helloworld.js; for example:



When the screen refreshes you should see the tab at the top of the editor know has the name helloworld.js:



The editor may also tell you that it will be colour coding the JavaScript.

**Step 2: Write a Program**

We will now write a simple JavaScript program which will initially print out a message saying ‘Hello World!’. This is a traditional first program in computer programming languages as it does several things; 1. It checks that your environment is working; 2. It checks that you can run the program you have written and 3. It checks that you can see the output of your program.

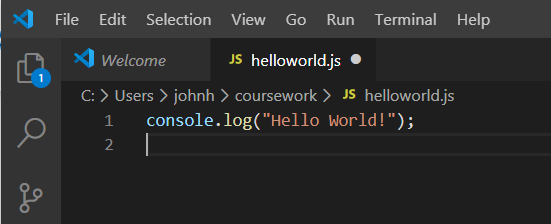
It is called Hello World because way back in 1972 an early book on Computer Programming used the ‘Hello World’ program to get students started (the language used was a programming language called B which is no longer used today but the idea behind the program itself is still with us).

To write Hello World in JavaScript we will use the console.log() function. This function will output (or log) the text it is given to the console. The console is where such output is written; if we run JavaScript form the command line then the window in which you run the program is the console. If you run the JavaScript in a web browser then the console is hidden within the browser (we will see this later in the course).

Add the following line to your file:

console.log("Hello World!");

This shown below:



Notice how different colours are being used. The blue indicates that this is a part of the JavaScript world; the yellow indicates that log() is a function that you can call and the brown indicates that this is a string.

Also note that you need to be very careful with this code. The string must be opened and closed using the double quotes or single quotes – but you can’t mix them. So “Hello World” is valid ‘Hello World’ is valid but “Hello world’ is not!

Also not that the line is terminated by a ‘;’ – this indicates that one statement (or if you like sentence) in the program has finished and anything coming after this is another statement (or sentence).

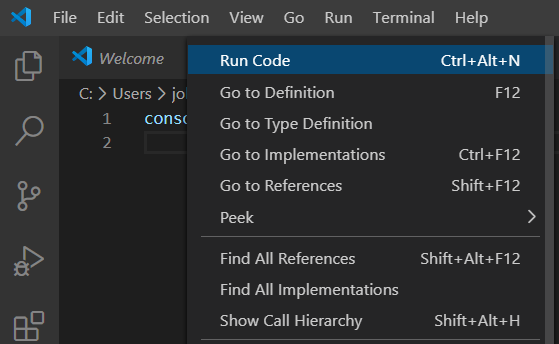
Now save the file – either go the File menu and select Save or use the keys CTRL-S.

You need to make sure you save the program / file every time you want to make a change so that when Visual Studio Code runs the program it picks up the latest version (otherwise it will run the last version of the file that was saved).

**Step 3: Run the Program**

Next bring up the right mouse menu while your cursor is within the file you just created.

This menu has a lot of options on it but at the top of the file you should see the option to ‘Run Code’:



Select this option and Visual Studio Code will run the code within the current file for you.

(If you do not have this option available then you will need to install a plugin to visual Studio Code that will do this for you. This is needed because Visual Studio Code tries to be lightweight but can be used to write programs in many different languages. To avoid increasing the amount of functionality included by default when you install Visual Studio Code you can decide which languages to support and what operations.

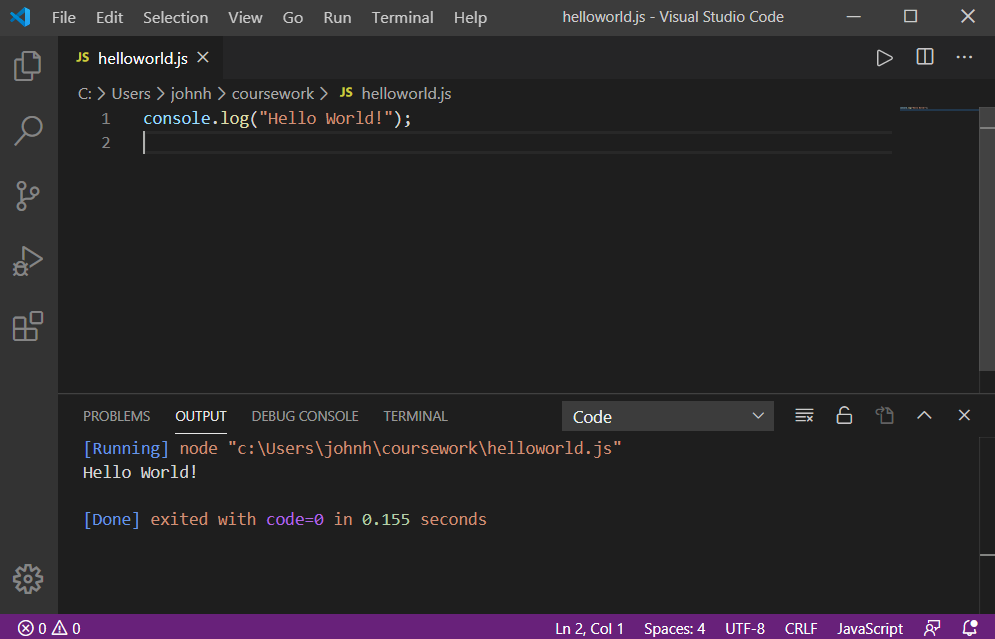
To add the ability to run JavaScript you will need to go to the following link:

<https://marketplace.visualstudio.com/items?itemName=formulahendry.code-runner>

On this page click on the Install button. Ok through the remaining screen which will then start to install the runner in your Visual Studio Code installed on your machine. Now try the right mouse button menu and select Run Code).

When you select to run the helloworld.js file using the JavaScript code runner from within Visual Studio Code you are actually using Node.js to execute your JavaScript for you. Node.js is a runtime environment for JavaScript; that is, it can execute JavaScript programs. Another runtime environment for JavaScript is embedded within a Web browser.

In my case the result of running the helloworld.js file is shown below:



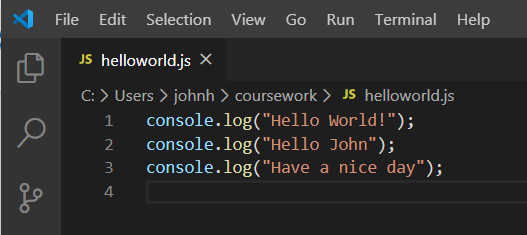
This opens an output window below your editor window with your file in it. This shows the output that has been logged to the console.

Why is this called logging? It is called logging because it is often used to trace or log the steps a program has taken so that if something goes wrong; later investigate can reconstruct the execution path or operation of that program. This allows developers to then debug the program and work out what went wrong.

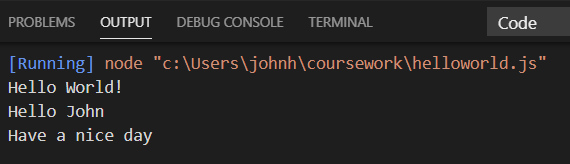
**Step 4: Extending your program**

You can now return to your program in the editor in the window above the output.

In this window type in some of your own messages to be logged; such as ‘Hello <whatever your name is’ and ‘Have a nice day’. Each of these can be printed (logged0 to the console using its own call to the console.log() function. For example:

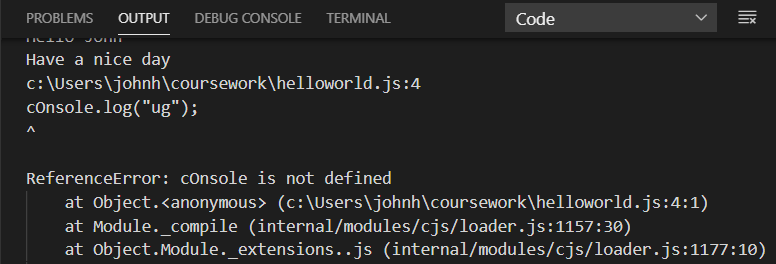


This three-line program would now generate the following output:



Again, be very careful with the syntax of the programming language; computer programs are very, very strict about these things. So make sure you open and close the string, make sure you finish each line with a ‘;’. Also make sure you use the same capitalization as I have for console.log(). In a programming language such as JavaScript console.log() is not the same thing as cOnsole.Log().

If you do miss-type any of the program statements then the JavaScript runtime will generate an error and list what was happening when the error occurred, for example if I had written cOnsole.log() then tried to run the program I would see the following in the output window:



This tells me that at line 4 in the file helloworld.js it found something that was not defined (that it did not understand).

To run the program you can also just click on the arrow like symbol at the top of the file as this just reruns the last program that it tried to run, for example look for:



At the top right hand side of the IDE and click on it.