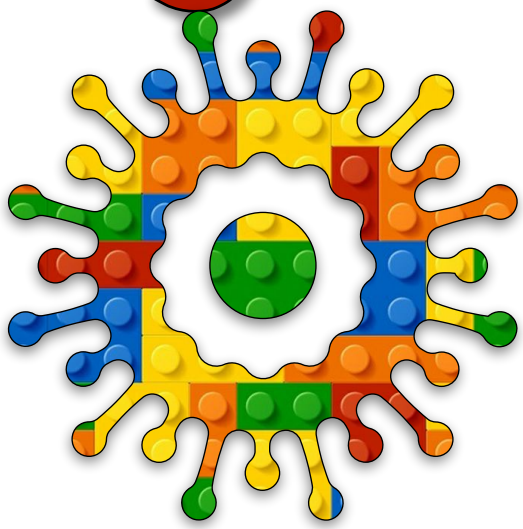
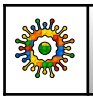
Coding Basics Module E Unit #1 creating & uploading files



Content

Module E Unit #1 creating & uploading files



Introduction to creating & uploading files

There aren't any sketches in this unit, just an explanation concerning the file system and the console. It is quite intuitive and straightforward but also needs practise. So if it goes horribly wrong, try, try again.

We will be mainly concerned with creating files. They help to keep the amount of one code in one place to a minimum by distributing across a few. This is particularly useful when using Object Orientated Programming with classes. You will see, otherwise you can have one very long sketch.

The purpose of the console

The `console` is the grey boxed section underneath where you enter your code. This is a very useful function for a number of reasons. It is where you will get error messages and where you can send information. It's main purpose is for debugging which is another word for problem solving.

Figure 1: the console

Error messages

If you editor picks up on some glaring and obvious errors in your code which can be anything from a missed comma or semicolon to an unknown variable appearing. Mostly you will get something useful and informative although sometimes it just flags a problem and leaves you to search for it.

Sometimes the error code may be highlighted in red and it may even give you the line number to identify where the problem is (or the following line number where it does become a problem).

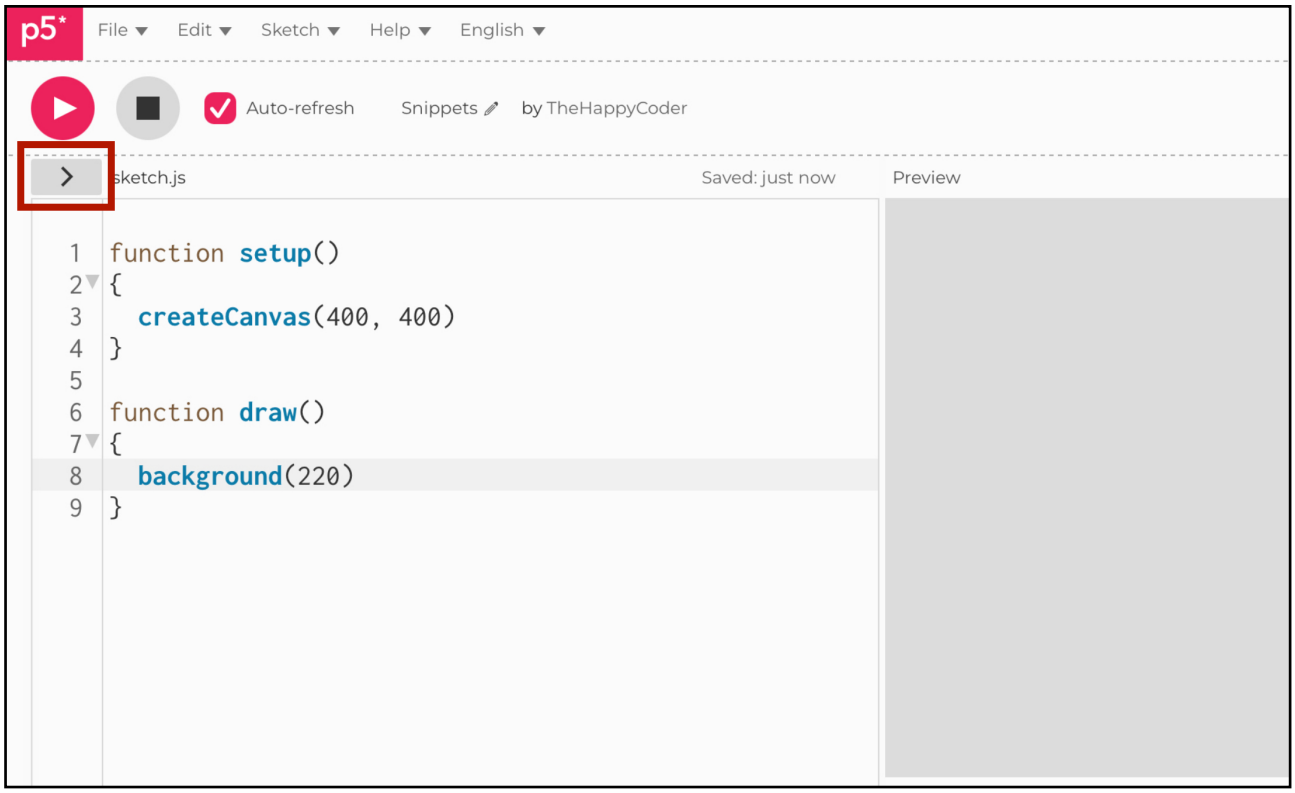
Console log

We can put a line of code in called `console.log()`. This means you are going to log something in the console. Sometimes it can be a piece of text such as `console.log('finished training')`, or it can be the value of a variable such as `console.log(counter)` which will give you the value of the variable at that time in the code. Another really helpful use is debugging a problem where you are not sure what is happening, for example, to an array.

Accessing the sketch files

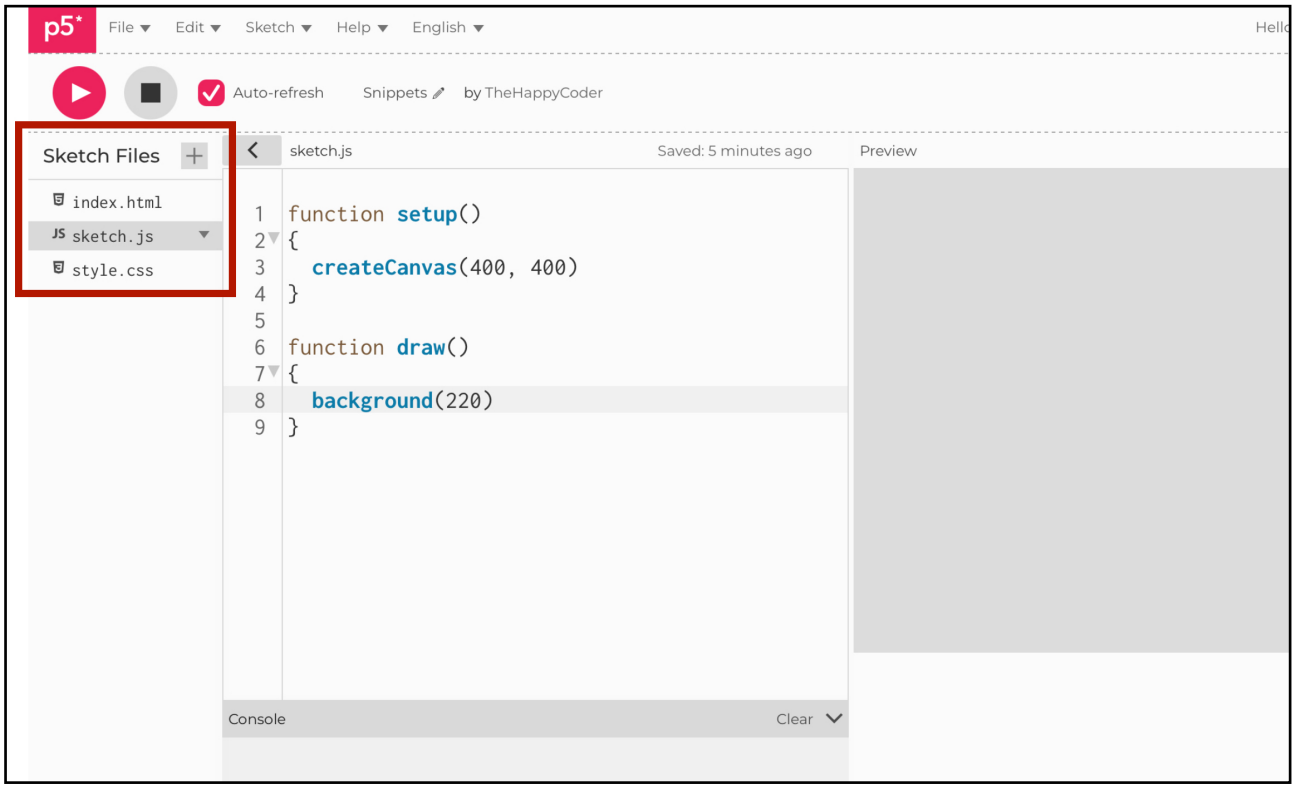
It is going to be critical to understand where the sketch files are and what files are used in creating a sketch. To the left of where you type your code you will find a grey box with a grey arrow. I have highlighted in red, click on it.

Figure 2: sketch files



When you click on it you will get a menu of Sketch Files associated with the editor. Those listed here are index.html, sketch.js and style.css. The main ones you will be interested in are the sketch.js and index.html. The style.css can be ignored (for now).

Figure 3: files menu

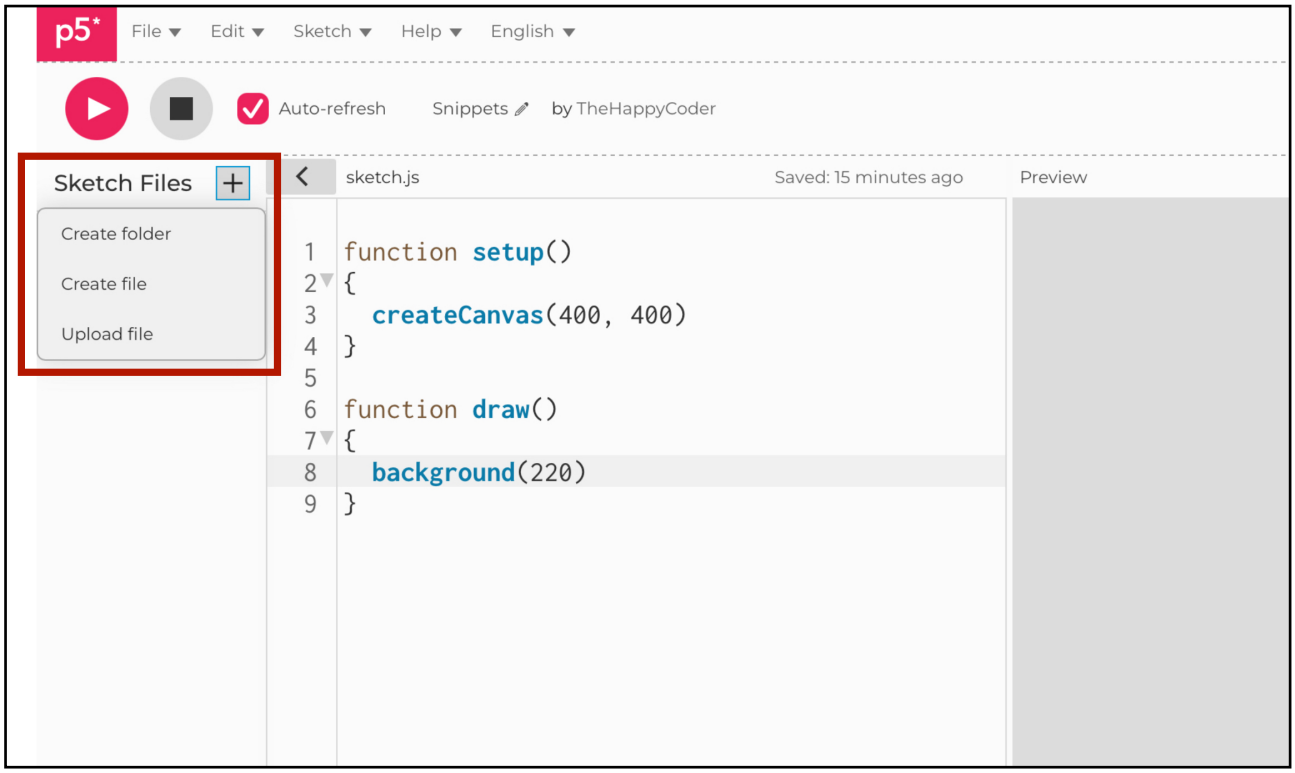


Adding files

Although I have avoided adding extra files in the majority of the coding there are times when adding them makes things neater and, on the whole, easier to manage. You have the `sketch.js` by default as your main coding file but you may want to add more. You will need to give it a name and file extension `.js`, for example `particles.js`.

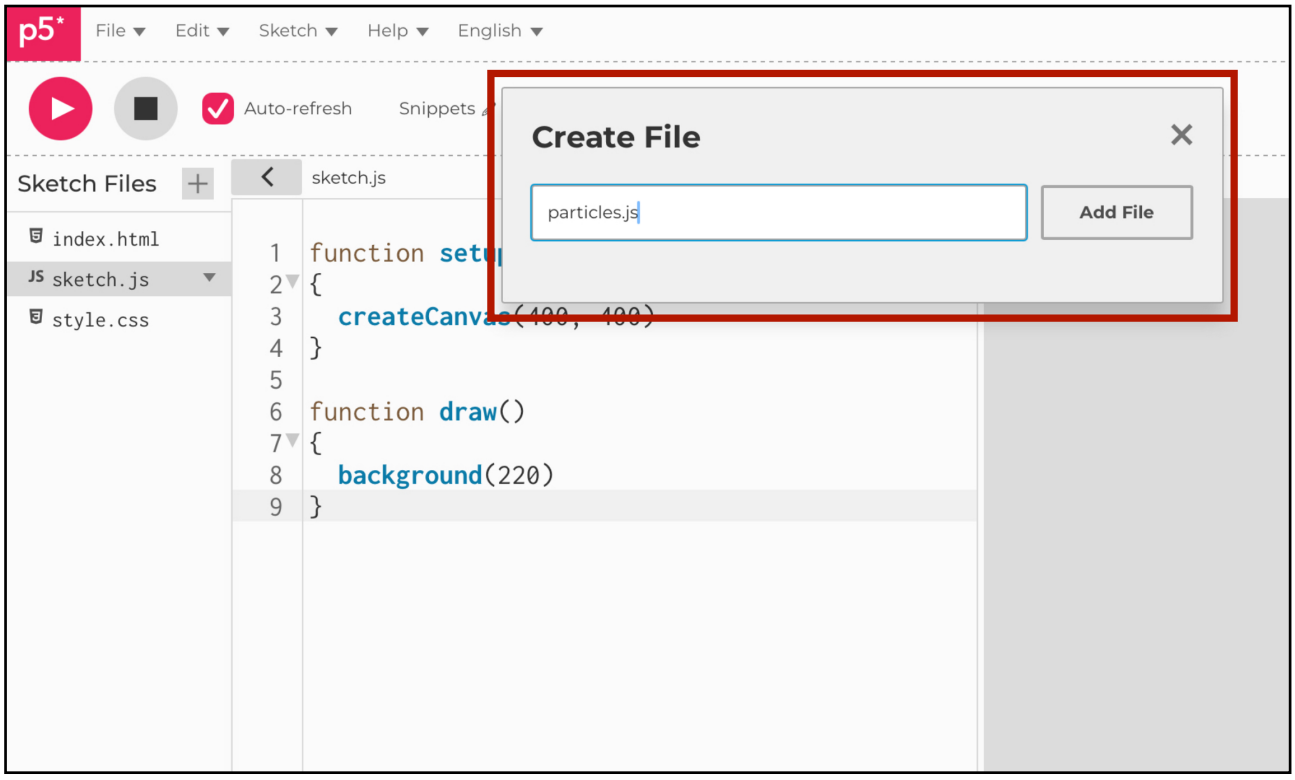
1 You add them by clicking on the grey + sign next to `Sketch File`.

Figure 4: adding files



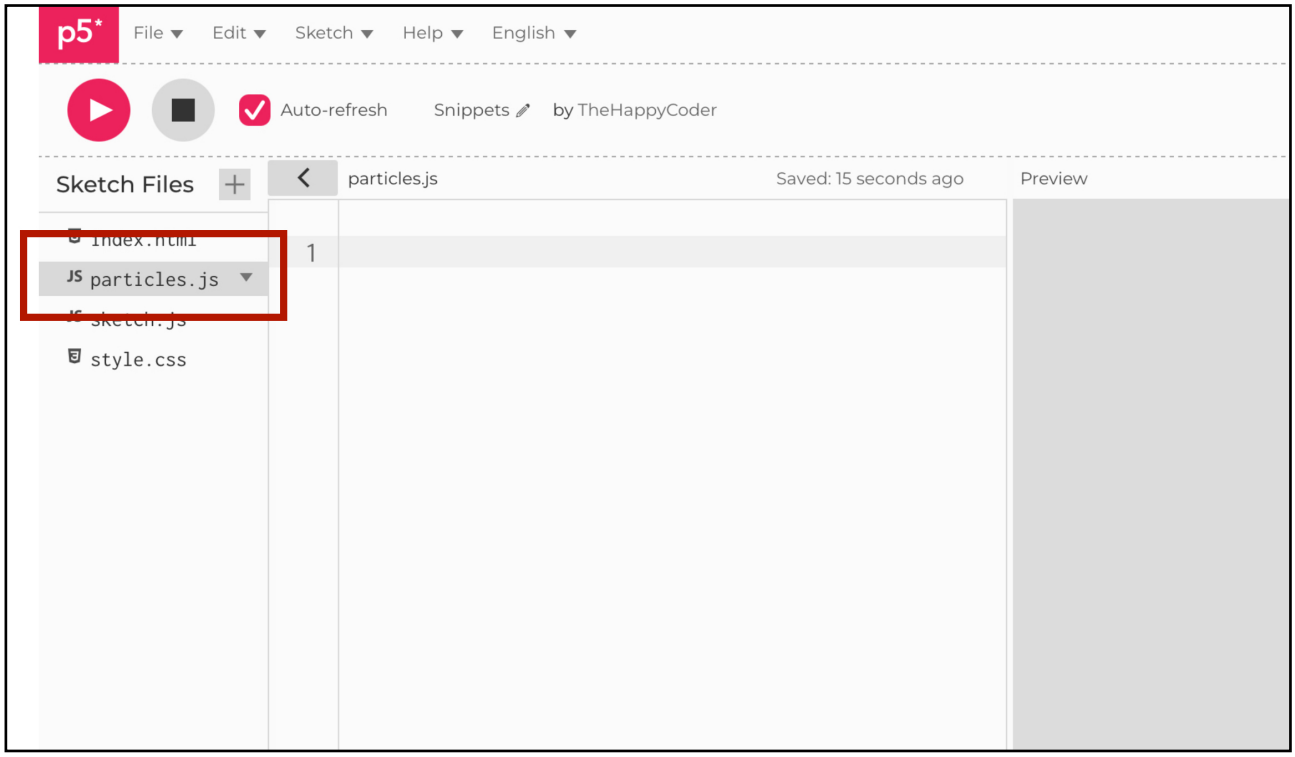
2️⃣ Now you click on `Create file`. You will get a box appearing waiting for you to put the name of the file in (important that it is case sensitive and must have the `.js` file extension at the end).

Figure 5: creating and naming files



3️⃣ Click on `Add File` and the file will appear in the `Sketch files`. Notice that the file is empty of any code. However, we aren't done yet, to use this new file we have to make reference to it in the `index.html` file.

Figure 6: adding the file

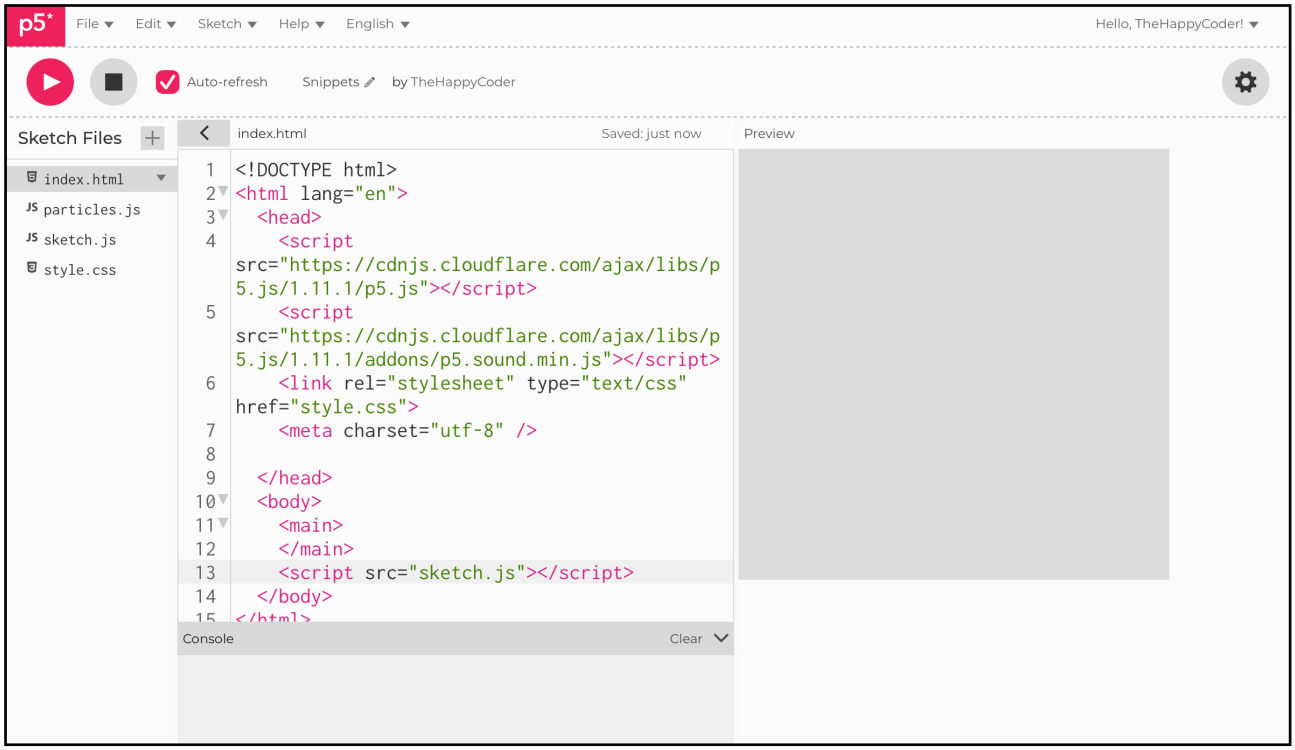


The index.html file

Although we are just adding another file we have created you will be adding other script tags which we use for links to `ml5.js` and `matter.js` etc, but for now I will show you how to add the file we have just created. This bit is often easy to forget to do and then you wonder why the code is throwing a wobbler at you.

4 This time we click on the `index.html` file to open a bewildering amount of code. Don't be put off if you are not familiar with html tags.

Figure 7: the index.html



This is the code within the `index.html` file. The index file is the core of the editor because it is where the editor interacts with the web browser. HTML is a tag based coding language and is often used to create web sites. All this is in the web editor by default. We don't need to install the p5.js code instead we access the library through a link. Here is a brief explanation:

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.11.7/p5.js"></script>
```

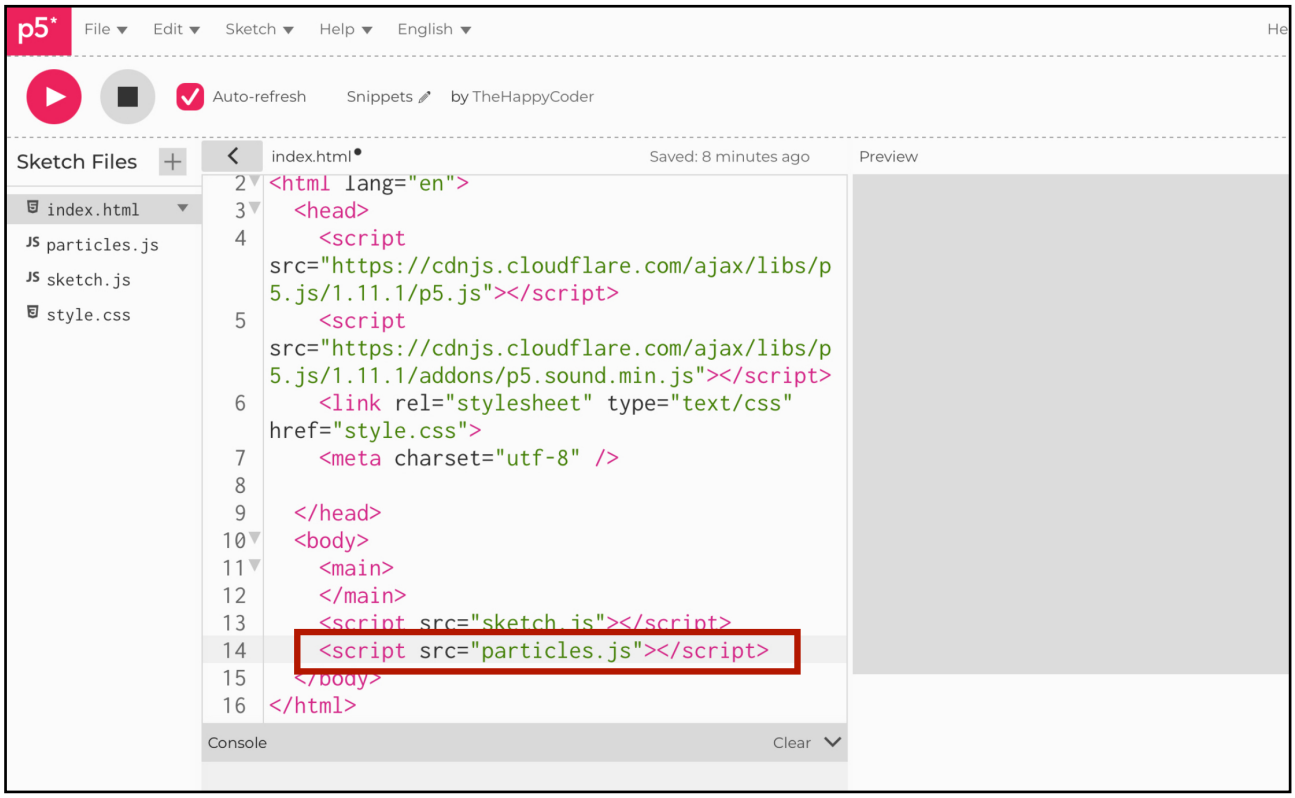
The code we use is accessed using cloudflare, we are using p5.js version 1.11.7 (as of writing). Also we have another library so we can use sound:

```
<script  src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.11.7/
addons/p5.sound.min.js"></script>
```

Notice that these lines have <...> symbols, these are called tags. There is a lot I could share about JavaScript, CSS and HTML, how they work together to create websites, how they give text style and functionality, but that is a topic all on its own and not essential for this tutorial, although might include one at some time in the future.

5️⃣ The key point here is the line of code that shows the `sketch.js` tags, what I do is copy and paste it underneath and change the second `sketch.js` to the new name of the file, in this case `particles.js`. It has to match the name exactly.

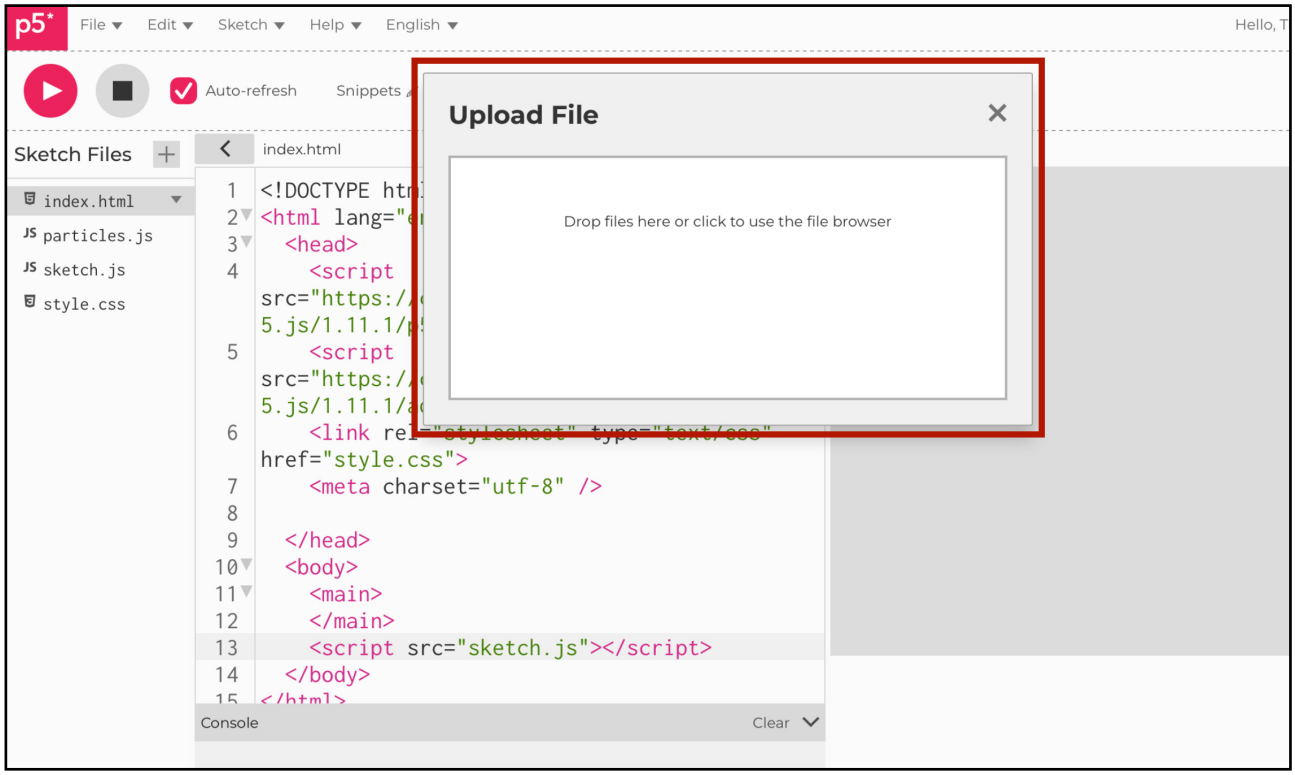
Figure 8: index.html new file



Uploading files

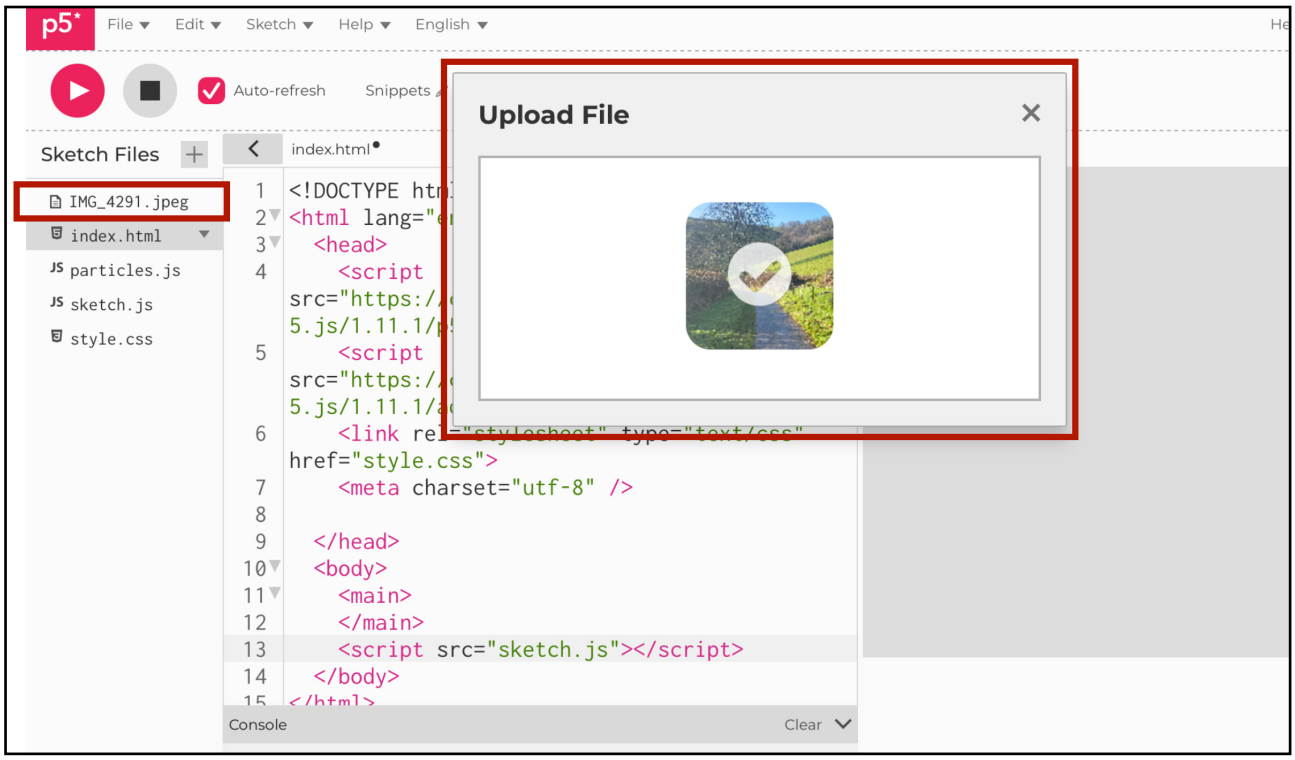
Going back to the `Add files` bit we can also `Upload files`. These tend to be files that contain images, data, videos, models, fonts etc. the process is quite simple but you do need to have the files ready to either access through browsing or to drag and drop from your desktop.

Figure 9: uploading files



1️⃣ In this instance I uploaded an image of a photo I took out walking one day in Devon. You will notice that the image has a file name on the right in the Sketch files. We can change this to something more memorable. Just note that there is a limit to the size of file and the first one I chose was just too big, the limit is 5MB, so make sure you choose a file that will meet those limitations.

Figure 10: uploading image



2️⃣ Let's change the name to path.jpeg. It is important to keep the same file extension but we can change the name.

Figure 11: renaming file

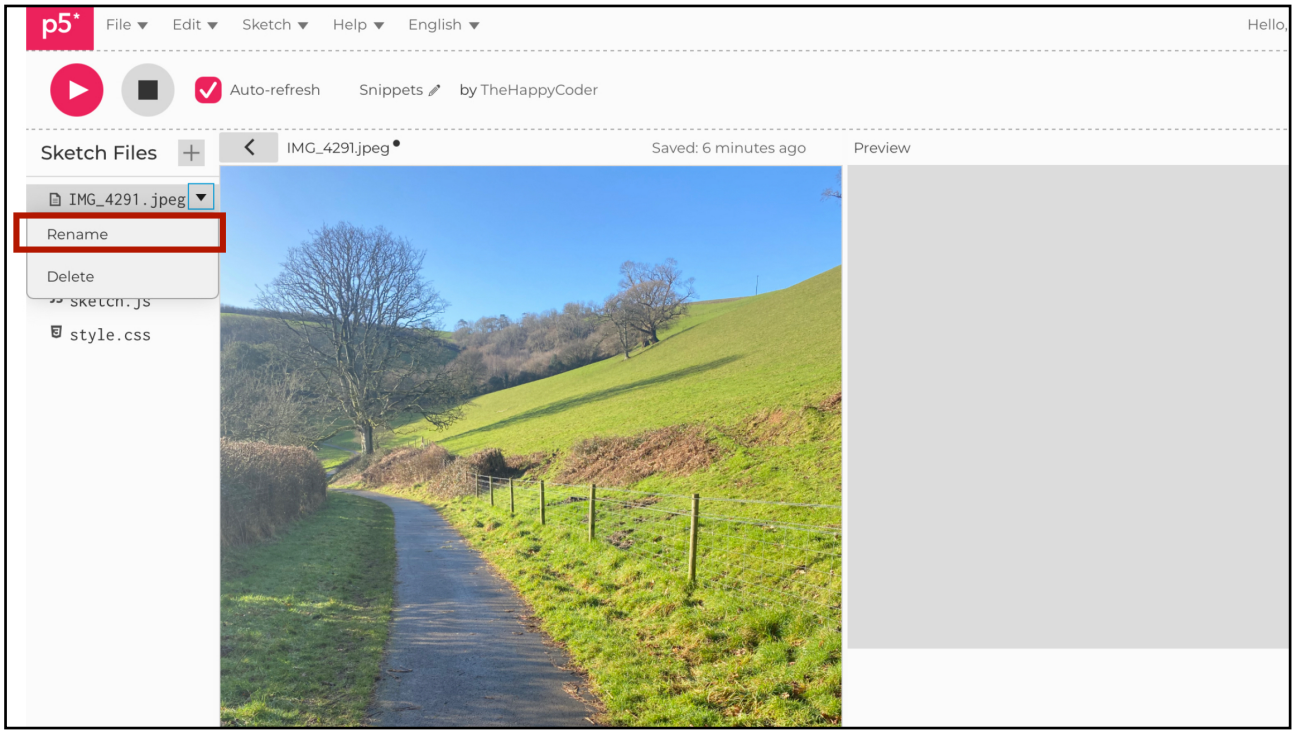
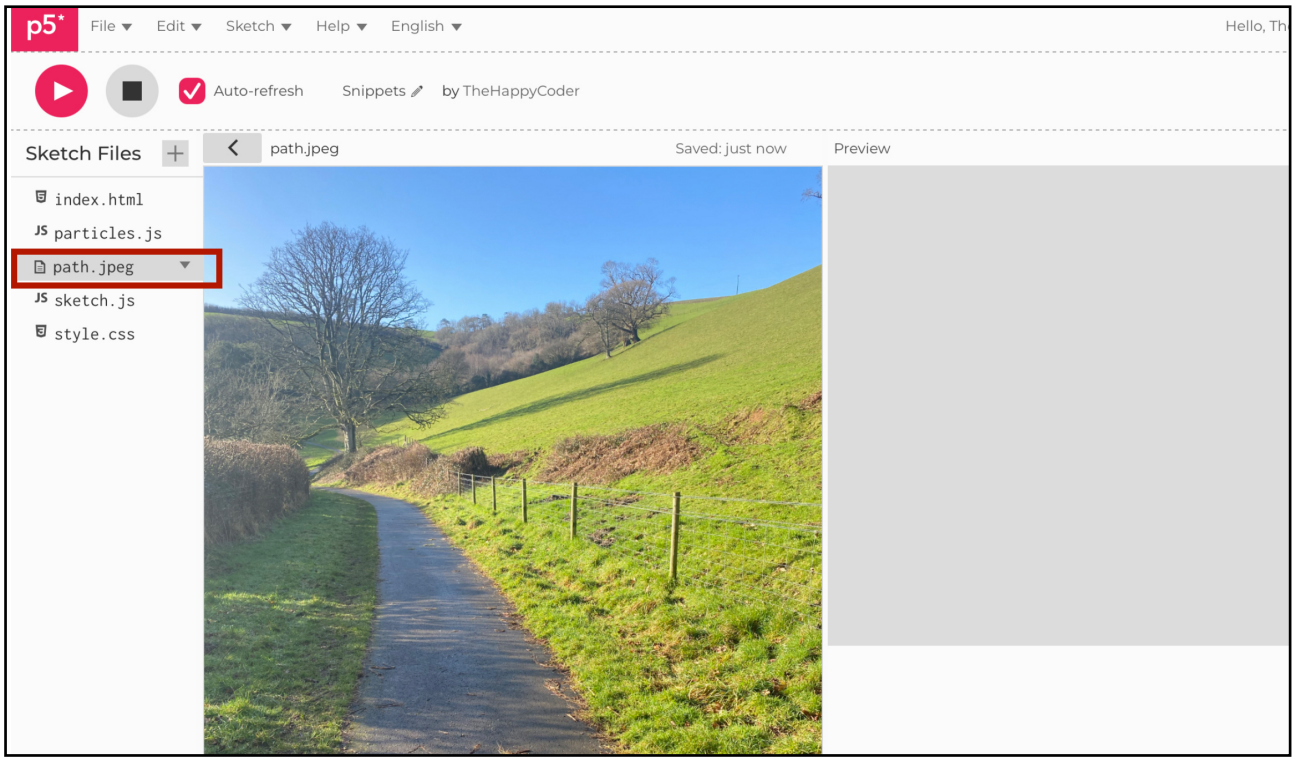


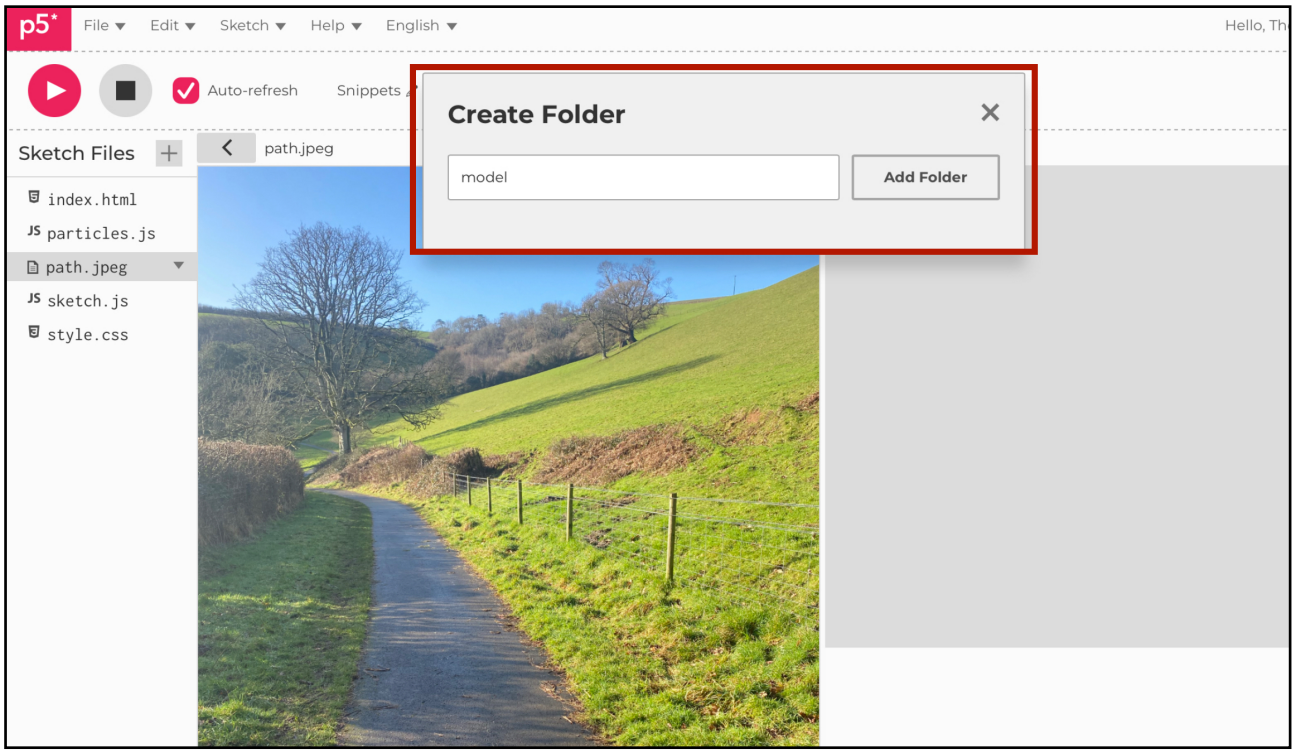
Figure 12: renamed file



Creating folders

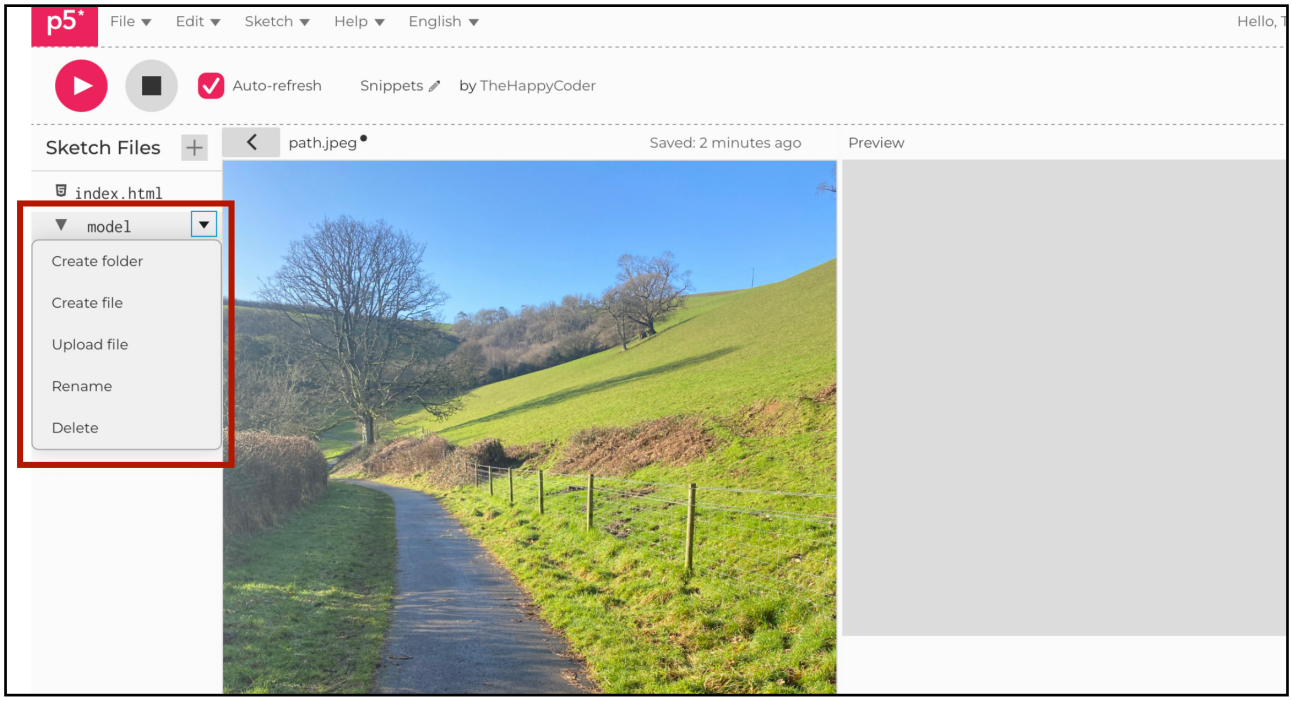
There are instances where you might want to group some images together or as in the case of saving an AI model you need to put all the files together in one folder. To do that we can create a folder and give it a name. Once it is created we can create files or upload files into that folder.

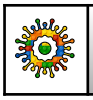
Figure 13: folders



As you can see you can even create a folder within a folder etc.

Figure 14: folder menu





Final thoughts

On the whole there is still a lot more to say about the web editor but most of it is fairly intuitive, the best way is to play and explore but for now you can have a read through to see what options are available and the use they may or may not be.

I will be showing you how to use created files as well as upload files, such as images etc in this module