Once you have your editor set up, you can start coding in JavaScript.

**Basic Syntax Rules**

“Syntax” just means the way the code is written, with how everything is ordered and which commands go where. Here’s a few basic syntax rules that will be extremely useful:

* Every single command should end with a semicolon. This is *the most* important rule when coding in JavaScript
* A double slash (//) makes a “comment” - a line of code that isn’t run by the computer. Use comments to organize and explain your code so that it makes sense when you read it again
* You can also make multi-line comments. To start, use (/\*), and to end it, use (\*/). Everything in between the two slash-stars will not be run by the computer

**Drawing Shapes**

Before you can start drawing, you need to set up the canvas you’ll be drawing on. This is simply changing the WIDTH and HEIGHT variables in the code, and putting a background command to change the color of the background.

To start drawing shapes, all you have to is call a command and input the parameters you want to customize it. Here’s how it should look:



For your parameters, you would input actual numbers/values. For example:

rect(100, 200, 50, 100);

This will draw a rectangle at the point (100, 200) that is 50 pixels wide and 100 pixels tall.

You can see a full list of the shapes you can draw in the documentation on the classroom.

**Changing Colours**

Now that you can draw shapes, you’ll probably want to customize the colour of those shapes. This is done using two commands:

fill(r, g, b); and stroke(r, g, b);

The fill command changes the colour setting for filling shapes, while the stroke command changes the colour setting for shape outlines, lines and points. In order to change the colour of the shape or line, use the respective fill or stroke command beforehand with the desired RGB values, then draw the shape or line. For example:

fill(0, 0, 0);

stroke(160, 200, 0);

ellipse(100, 100, 40, 80);

This would draw an ellipse filled black with a yellowish-green outline. You can also draw shapes that are *not* filled (transparent) or have no coloured outline. This is done with the commands:

noStroke();

noFill();

The last thing you can change is the thickness of the lines and shape outlines you draw. This is done with the following command:

strokeWeight(thickness);

Note that with all the above commands, the commands need to be written *before* the shape you want to draw.

**Practice Exercises**

1. Draw a magenta circle with a white outline in the centre of the canvas. Make the thickness of the outline be 4.
2. Draw a light blue circle perfectly contained within a dark red rectangle (edges should touch, no part of the circle should be outside rectangle)
3. Draw two arrow shapes (⇨ and →). The first should be drawn using only shapes and be transparent (no fill colour). The second should be drawn using only lines.

Challenge: Draw a basic house using various shapes in the documentation file. You can make your house look like whatever you want, but the house should have a door, a window, a roof and a chimney. Try to use several different colours and make your house look unique.