Shapes & Drawings



rect(x, y, width, height, radius\*);

Draws a rectangle with the top-left corner being the (x, y) coordinate and the width and height being how far right and down it extends. The radius value is optional, it determines how rounded the corners of the rectangle are



ellipse(x, y, width, height);

Draws an ellipse with the centre of the ellipse being the

(x, y) coordinate.



line(x1, y1, x2, y2);

Draws a line from the point (x1, y1) to the point(x2, y2).



triangle(x1, y1, x2, y2, x3, y3);

Draws a triangle using the points (x1, y1), (x2, y2), and

(x3, y3).



quad(x1, y1, x2, y2, x3, y3, x4, y4);

Draws a quadrilateral using the points (x1, y1), (x2, y2),

(x3, y3), and (x4, y4). Be sure to list the points *in order going clockwise*.



point(x, y);

Draws a point at the coordinate (x, y)

**Controlling Shapes & Drawings**

fill(r, g, b);

Sets the RGB color to fill shapes

stroke(r, g, b);

Sets the RGB color for lines, points and shape outlines

background(r, g, b);

Sets the background color

noFill(); or noStroke();

Disables coloring for filling shapes, or disables coloring for outline of shapes, lines and points

strokeWeight(thickness);

Changes the thickness for lines, points and shape outlines

**Variables**

**var** [**variableName**] = [**value**];

Declares a variable with the name and value above

[**variable**] **+=** [value]

Adds a value to a certain value. Also works with subtraction, multiplication and division

**variable ++**;

Adds 1 to the variable. Also works with subtraction

var [**colorName**] = **color**(**r**,**g**,**b**);

Declares a variable that stores a color based on the RGB value. Can be used in fill and stroke commands

**Functions**

**var** [**functionName**]= **function(parameters)** { };

Declares a function with the name above and contains code in the curly brackets. The code can use the parameters as variables within the function itself. If you don’t need parameters, leave the inside of the parentheses blank.

[**functionName**](**parameters**);

Calls a function using the parameters inside the parentheses. If the function requires no parameters, leave the inside of the parentheses blank.

**return** [**value**];

Used at the end of a function to return a value when the function is called. The value can be used wherever that type of value can be accepted (i.e. if it returns a number, the function can be used wherever a number is accepted, such as a coordinate for drawing a shape).

**Mouse Functions and Variables**

**mouseX**, **mouseY**

The X and Y coordinates of the mouse respectively. Can be called as a variable and do not have to be declared.

**mouseClicked** = **function()** { };

A function that is called whenever the mouse is clicked. Like all mouse functions, it has to be declared first, but does not need to be called within your code.

**mousePressed** = **function()** { };

A function that is called while the mouse is held down.

**mouseOut** = **function()** { };

A function that is called when the mouse moves outside of the canvas.

**mouseButton**

Which mouse button is being pressed (LEFT, RIGHT, CENTER). Called as a variable and does not have to be declared.

**mouseReleased** = **function()** { };

A function that is called when the mouse buttons are released (after being held down).

**Miscellaneous**

**draw** = **function**() { }

A function that runs its code several times per second, depending on the framerate (default is 60 FPS)

**frameRate**(30);

Changes the framerate of the draw function to the number in the parentheses

**random**(**low\_value**, **high\_value**);

Generates a random value from the low value up to but excluding the high value