CM1005 - Introduction to Programming I

BSc Computer Science

Nathan Donovan

University of London

April 2024

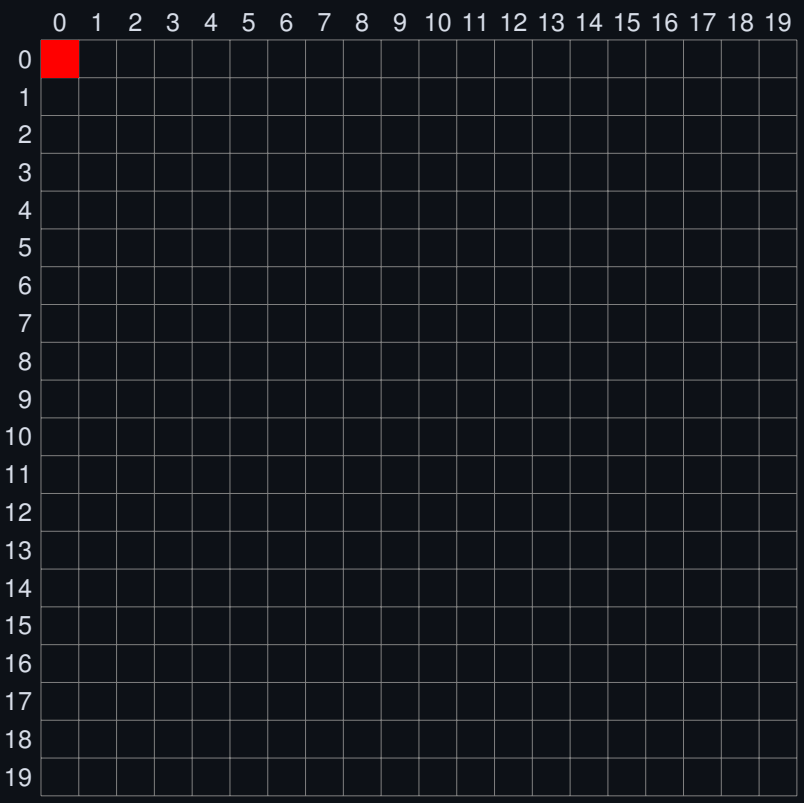
1 p5.js - JavaScript Library for Creative Coding

Using p5.js

- Projects / programs must be opened at the folder level in Brackets.io / other IDE for live preview to work.
- Work on sketch.js file only.
  - index.html is boilerplate for the sketch to run as a web page.
  - p5.min.js is the p5.js library code.

Function used to set up the sketch, width and height are specified in pixels with the origin (0,0) at the top left hand corner of the browser window:

```
function setup() {
    createCanvas(width, height);
}
```



Origin pixel highlighted in red at (0, 0), example canvas is 20 x 20 pixels.

Drawing and Shape Functions

Function used to contain the drawing commands in the sketch:

```
function draw() {
    // Various functions available as part of the p5.js library
}
```

Example p5.js shape functions used:

```
// rect() syntax
rect(x, y, width, height, [tl], [tr], [br], [bl]);
/*
tl = top-left radius
tr = top-right radius
br = bottom-right radius
bl = bottom-left radius
*/

// ellipse() syntax
ellipse(x, y, width, [height]); // height is optional

// point() syntax
point(x, y);    // point() is a single pixel unless modified with strokeWeight()
                // Can only be colored with stroke(), not fill()

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

// vertex() syntax
vertex(x, y);   // Can be used to construct complex shapes
                // Used exclusively with beginShape() and endShape()
// Example:
beginShape();   // fill(), stroke() etc. to be used before beginShape() called
vertex(x1, y1);
vertex(x2, y2);
vertex(x3, y3);
endShape();

// beginShape() syntax
beginShape([kind]);
/*
kind = POINTS, LINES, TRIANGLES, TRIANGLE_FAN, TRIANGLE_STRIP, QUADS, QUAD_STRIP or TESS

examples shown in p5.js reference documentation
*/
```

Colour Functions

Example p5.js color functions used:

```
// stroke() syntax
stroke(red, green, blue, [alpha]); // alpha is optional
// red, green, blue and alpha values are between 0 and 255

// fill() syntax
fill(red, green, blue, [alpha]); // alpha is optional
// red, green, blue and alpha values are between 0 and 255

// noFill() syntax
noFill();   // Disables fill color for shapes
```