<Beyond The Canvas/>

GAD405/DAT405 – Creative Coding

<Beyond the Canvas/>

GAD405/DAT405 – Creative Coding

- **How to Wrap and Position the p5 Canvas**
- **How to Create other HTML Elements with p5**

<Beyond the Canvas/>

How to Wrap and Position the p5 Canvas

As we are aware, createCanvas creates an HTML5 Canvas, which we then draw to using the draw function in our sketch.js

However, p5.js can also be used to create and interact with HTML elements outside of the graphics canvas.

First we need to make sure that we are calling, the p5.dom add-on library, alongside our other scripts

<Beyond the Canvas/>

How to Wrap and Position the p5 Canvas

For this example, I am going to use the Interactive Colour Grid that made up the optional task in last weeks homework task.

If you do not have it, Stavros has uploaded the completed code here

https://github.com/stavrosdidakis/DAT-GAD-405_2017/tree/master/Session3/12_Exercise05-InteractiveColorGrid_(Solved)

You can use any of the Generative Pieces that we worked on last week, or you can use your own

<Beyond the Canvas/>

How to Wrap and Position the p5 Canvas

Once you have the code, you will see that p5.dom.js is already being loaded in our header file

```
6       <script src="libraries/p5.js" type="text/javascript"></script>
7       <script src="libraries/p5.dom.js" type="text/javascript"></script>
8       <script src="libraries/p5.sound.js" type="text/javascript"></script>
```

If you are using your own project, and don't have the p5.dom.js file, then you can get it from one of the projects from last week, and move it into your folder, it is best practice to keep them in a folder, called libraries

<Beyond the Canvas/>

How to Wrap and Position the p5 Canvas

When we call the createCanvas function, we define a width and a height, and the script creates a canvas for us to draw to

One thing that we may have noticed, is that our projects usually start in the top left corner, and doesn’t look like it forms part of a webpage.

If we call our createCanvas function in as a var or let, then we can access some additional functionality, or methods, that p5.dom has added for us

<Beyond the Canvas/>

How to Wrap and Position the p5 Canvas

First things first, lets put our createCanvas, into a var

```
3  function setup() {
4    var canvas = createCanvas(500, 500);
5  }
```

This is now referred to as a reference or pointer, and now that we have a pointer, we can call methods to it, a full list of which can be found here

https://p5js.org/reference/#/p5.Element

Lets take a look at some of the more popular methods

<Beyond the Canvas/>

How to Wrap and Position the p5 Canvas

Here we are calling the methods “position” and “class” on our pointer.

```
3   function setup() {
4     var canvas = createCanvas(500, 500);
5
6     canvas.position(300, 50);
7     canvas.class("myCanvas");
8   }
```

The way in which the syntax works for this is the pointer name followed by a . followed by the name of the method we wish to call, most of the methods available will then need additional parameter to function correctly

If you save this, and open the index.html file in your browser, you should see that the sketch as moved 300px from the left, and 50 pixels from the top

If you right click and inspect element on the canvas, you should see in the developer tools window, that the canvas element, now has a class applied to it

Which we could then use as a selector, in our CSS

It is worth being careful with the position method, as this utilises the CSS position property, and applies absolute positioning, which can cause some sleepless nights

<Beyond the Canvas/>

How to Wrap and Position the p5 Canvas

Something else to take note of here, is that the canvas is a generated tag, by our p5 script, and is just placed inside the body tag, usually after the elements that already exist in our page

This is going to give us some problems, you we wish to create a more aesthetically pleasing look and feel to our page, by included other HTML elements, such as a header or footer, and we wanted to have our canvas in, in between the two

This is where we use the parent method

<Beyond the Canvas/>

How to Wrap and Position the p5 Canvas

In our HTML doc, lets create a div, with the id of myContainer, inside our body tag

```
13    <body>
14      <div id="myContainer"></div>
15    </body>
```

This element can be placed anywhere in our body, so as mentioned before between a header and footer tag, or even inside a header and footer tag

<Beyond the Canvas/>

How to Wrap and Position the p5 Canvas

In our set up function, lets call this method to our pointer

```
3   function setup() {
4     var canvas = createCanvas(500, 500);
5     canvas.parent("myContainer");
6   }
```

We pass into this method, the id name we have given to the element that we want our canvas to be nested into

<Beyond the Canvas/>

How to Wrap and Position the p5 Canvas

If you then go to your browser and refresh, the right click>inspect (if it has closed), you should see in the HTML, that our canvas, is now inside the container div

Just to prove the point, add a h1 tag with some text in, either side of the container div we created, save it

```
13    <body>
14      <h1>This should be at the top</h1>
15      <div id="myContainer"></div>
16      <h1>This should be at the bottom</h1>
17    </body>
```

You should see that the text appears, as desired, above and below our container element.

If you go to our setup function, and remove the .parent method, save and refresh, you should see that the order will change, and the elements defined in our HTML document, will appear first

<Beyond the Canvas/>

How to Create other HTML Elements with p5

Lets look at how we create some HTML elements with our P5 sketch.

There are whole host of methods that we can use to create and manipulate items in the DOM (Document Object Model)

The DOM, in very basic terms, is a logical structure of a document, and in the case of a webpage, it is what our browser generates, so that a programming language, such as javascript can read and update the document

A full list of the methods available, can be found here:

https://p5js.org/reference/#/libraries/p5.dom

<Beyond the Canvas/>

How to Create other HTML Elements with p5

We are just going to create a very simple div, with some text inside for this demonstration, and we will look more into some of the other elements, and their interaction capabilities, when we look at creating a Graphical User Interface or GUI, next week

<Beyond the Canvas/>

How to Create other HTML Elements with p5

This is a very similar process to creating the canvas, which we can also put into a variable, and use as a pointer, to call methods in

```
7    var text = createDiv('This is an HTML string!');
```

If you save this, and reload your html doc in the browser, you will see that it appear at the bottom, but we can use the position or the parent methods here, to rectify that

```
7    var text = createDiv('This is an HTML string!');
8    text.position(50,50);
```

This will also prove, that absolute positioning, can be tricky

<Beyond the Canvas/>

How to Create other HTML Elements with p5

Workshop time!

I’d like you to now start combing the things we have learnt in todays session, and start to create a full page to house a p5 sketch, using HTML, CSS and p5

Please feel free to utilise any prior knowledge you have of any of the above areas, and utilise me, and other online resources to start experimenting with CSS

We have only scratched the surface

Before the end of the session, I will stop you all, and show you how you can host your creation online for free, with GitHub Pages

<Beyond the Canvas/>

Where can I Find out more?

Online Resources

P5 Reference Library

https://p5js.org/reference/

P5 Beyond the Canvas

https://github.com/processing/p5.js/wiki/Beyond-the-canvas

Let’s Chat!

GAD405/DAT405 – Creative Coding