Advanced Web Development Technologies (HTML5 Canvas and jQuery)

HTML5 Canvas[1]

Getting Started

The HTML5 canvas is an HTML element that allows you to dynamically render graphics onto a webpage. It provides an alternative to static images, and has many real life applications, such as making interactive maps, and as a digital drawing medium.

An HTML5 canvas can be added to a webpage using the “canvas” tags. You need both an opening and closing “canvas” tag, as the element supports an alternative display* if there is an issue with the canvas. You will also need to make use of its “width” and “height” parameters in order to define the area that the canvas should take up. I would also recommend using the “id” tag so that the canvas may be more easily retrieved in your application’s associated JavaScript file - as you can’t do anything with the canvas by using HTML alone.

*This alternative display can be added by inserting either text or an image between the “canvas” tags.

Drawing Shapes

Before you can draw anything on an HTML5 canvas, you must first retrieve its context. You can do this by locating the canvas element by its identifier, and calling the getContext(type) function on it. Canvases can either be 2D or 3D spaces, so you must use the “type” parameter to specify which you would prefer to draw on. The following functions are all methods of the HTML5 canvas context object.

When it comes to the 2D space, the HTML5 canvas has the ability to draw two things: Rectangles and paths. Rectangles are fairly self-explanatory. You can use strokeRect(x,y,w,h) to draw an outline of a rectangle, (starting at the x,y coordinate and giving it a width of w and a height of h,) you can use fillRect(x,y,w,h) to fill in a rectangular portion of the canvas with a certain colour, and you can use clearRect(x,y,w,h) to empty a portion of the canvas. If you want to draw something that isn’t a rectangle, then you must use a path.

Paths are a collection of lines and coordinates that can be used to form shapes. Paths serve as an outline for what to be drawn, so nothing is actually drawn onto the canvas while a path is being made. You can start to make a line by calling the function beginPath(), and you can put an end to a path by calling closePath(). Between beginPath() and closePath(), you can use a range of functions to move your cursor around and draw both straight and curved (arced) lines.

After a path has been made, you can then choose to either:

- Fill the shape, using fill() . This will automatically seal a shape that was left open by connecting its final point to its starting point, and will fill the shape with the colour stored in its “fillStyle” parameter, which you can set yourself.
- Trace over the path, using stroke() . This will represent the given path on the canvas as a line. You can select the colour of this line using the “strokeStyle” parameter, and its thickness using the “lineWidth” parameter. This option does not make any attempts to seal up potential shapes.

jQuery

Getting Started

jQuery[2] is a JavaScript library that attempts to simplify common JavaScript actions, including retrieving information from a website/web server and handling events. It shortens well-used functions like document.getElementBy[attribute]() from lengthy strings to a single character with a bit of new syntax. This makes JavaScript code faster to write and easier to read.

There are two ways to use jQuery in your web application. The first and most common way is to go to https://releases.jquery.com/ and select the version of jQuery that you wish to use. Then, click on the desired version, and insert the script-tag-link provided into the “head” tags of your application. The downside of this method is that it requires an internet connection - as it is retrieving the jQuery script from a web server - so it will not work for offline applications.

The second (and offline-friendly) method is to go to https://jquery.com/download/ and manually download the source code for the version of jQuery that you wish to use. You can then integrate this script into your application as you would with any other script you have created.

Do note that, with both methods, the jQuery link must be inserted before all other JS links - as it must be loaded before any other program attempts to use jQuery. Otherwise, your browser will not be able to interpret your jQuery code as valid JS, and will raise an error.

Sources

1. https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API
2. https://jquery.com/