1. Introducing the Canvas App Template

Sam Scott, Sheridan College, 2013

First Steps

1. Download `CanvasAppTemplate.zip` from the web site[1]
2. Unzip the files and open `index.html` in any browser other than Internet Explorer
3. Go to the Developer Console (e.g. F12 in Chrome, ctrl-shift-k in FireFox, or search menus).
4. At the console prompt, type `alert("Hello, World!");` and hit enter.

What Did I Just Do?

You just executed a JavaScript statement. JavaScript is a fully functional programming language with a C-like syntax that runs in the context of a web page. Despite the surface appearance to C and Java, there are some paradigmatic differences between those languages and JavaScript. One such difference is that JavaScript supports pure **imperative programming**, which means that statements can be executed outside of any method, function, or class (like in Python). A browser's Developer Console allows imperative programming on the fly, using the current page as the context.

More Dialogs!

- Try `confirm("Are you OK?");` and `prompt("What's your name?");` and note the return values.
- Try hitting cancel when using `prompt` to see what happens.
- Try `alert("Hello "+prompt("What's your name?"));`
- Try `if (confirm("Are you OK?")) alert("Glad to hear it."); else alert("Oh");`

TIP: Use shift-enter for multi-line commands.

TIP: Use the up arrow to get back previous commands.

Now Try Drawing on the Canvas

- `canvas.strokeRect(100,100,50,50);` // (params are x, y, width, height)
- `canvas.fillRect(100,200,10,100);`
- `canvas.fillArc(50,50,25,0,360);` // (x, y, radius, startAngle, endAngle)
- `canvas.strokeArc(100,50,25,0,90);`
- `canvas.strokeLine(0,0,700,500);` // (x1, y1, x2, y2)
- `canvas.fillText("Hello, World!",200,200);`
- `canvas.strokeText("What does this do?", 300,300);`
- Can you use `fillArc` to draw Pac Man?

[1] http://www-acad.sheridancollege.ca/staff/scottsam/javascript2013

How Does This Work?

This Canvas App environment is designed to get students quickly creating cool stuff they can put on the web. The focus is on programming, and students are protected from having to know HTML, CSS, the Document Object Model, or some of the messy details of drawing on the HTML5 canvas.

The template consists of four files, but students only need to change the JavaScript code in `myCanvasApp.js`. The other files are a set of training wheels for the novice JavaScript programmer. Together, they provide:

1. Automatic access to several HTML objects including an HTML5 canvas.
2. Canvas extensions for easier drawing.[2]
3. Easy customization of the interface
4. A `debugOut` command for easier debugging. Try `debugOut("Something went wrong!");`
5. Easy code hookups for keyboard, mouse, button, and timer events.

Colors and Fonts

- Try `canvas.strokeStyle = "red";` and then try one of the stroke commands.
- Try `canvas.fillStyle = "DarkGoldenRod";` and then try a fill command.
- Go to http://www.w3schools.com/cssref/css_colornames.asp for a full list of colors.
- You can also specify colors by specifying red, green, blue (and alpha) components, like this: `"rgb(red, green, blue)"` or `"rgba(red, green, blue, alpha)"` where read, green and blue are 0->255 and alpha is 0.0->1.0. You can also use hexadecimal to specify colors like this: `"#FFCC99"`.
- Try `canvas.font = "50px monospace bold";` and then try `fillText`. Web safe font names are monospace, serif, and san-serif (bold and italic are optional).

More Power!

The Canvas also allows you to define more complicated polygons and curved shapes, as well as fills that use color gradients and other patterns. See http://www.w3schools.com/html/html5_canvas.asp and http://www.w3schools.com/tags/ref_canvas.asp for more info.

Debugging Issues

If you misspell something: e.g. `canvas.fint` instead of `canvas.font`, it will often fail silently. Also, since fonts and colors are strings you can easily mess those up too with no error from JavaScript (though FireFox does report these). You might also notice that semicolons are often optional, but it's considered bad practice not to use them and most IDEs will issue a warning if you miss one.

[2] Without `strokeLine`, drawing a simple line requires four different statements. Without `strokeArc` and `fillArc` students would need to know about radians.