# 2. Your First Web App

*Sam Scott, Sheridan College, 2013*

## Customize Your App

1. Copy the canvas app template files to a new location and rename the folder.
2. Open `myCanvasApp.js` in a text editor (It is highly recommended that you use a programming editor like **notepad++.** Ideally, you can create a project in **NetBeans** or **Eclipse** and move the files there.)
3. Find the `init` function and make some changes to the *customization* section. Then save the file, load `index.html` into a browser (or hit refresh) to see the effect.

## The 411 on Variables and Types in JavaScript

- **Dynamic Typing:** Variables do not have types, but values do (`Number`, `String`, `Array`, `Object`, `Boolean`, `Function`, etc.). This is a feature JavaScript shares with Python, but not with C or Java. Because of dynamic typing, a variable can hold a String one moment, a Number the next, then an Object, and so on.
- **Weak Typing:** Unlike Python, which has strong typing, you will almost never get a type mismatch error from JavaScript. No matter what the operation, JavaScript will do its best to make it work through automatic type casting. This can be both a blessing and a debugging curse.
- **Declaring Variables:** Variables are declared with the `var` keyword (e.g. `var x = 34;`).
- **Automatic Variable Declaration:** If you forget to declare a variable, the system will do it for you, but the variable will have global scope. **Undeclared variables are bad programming practice.**
- **Declaring Functions**: Functions are declared with the `function` keyword (e.g. `function foo(a, b, c) {}`). You do not need to specify a return value or types for the parameters and you are free to mix and match return types or not return anything at all.
- **Functions are Values**: Functions in JavaScript are actually values stored in variables. They can be declared like variables, passed as parameters, reassigned, etc. For most of what we're doing here, this doesn't matter, but it might help you make sense of some of the helper code.
- **Objects:** JavaScript implements objects without classes. You can access an object's fields using dot notation (e.g. `canvas.fillStyle`) or associative array notation (e.g. `canvas["fillStyle"]`). Objects can be created from from **prototypes** (http://javascriptweblog.wordpress.com/2010/06/07/understanding-javascript-prototypes/), **constructor functions** (http://www.w3schools.com/js/js_objects.asp), or on the fly like this:

```
var o = [];                 // object is empty
o.name = "Sam";             // object now has a field
o.getName = function() {    // object now has a method
      return this.name;
};
```

GOTCHA: This is why errors like canvas.fillstyle="red" fail silently.

- **Arrays:** Arrays can mix types and can be extended on the fly, like this…

```
var a = [34, "sdf", false];
a[3] = -3.2;
a[4] = [];
```

GOTCHA: Arrays can have "holes" in them (e.g. a[0] and a[2] are defined but a[1] is undefined)

For more on arrays, see http://www.w3schools.com/js/js_obj_array.asp and http://www.w3schools.com/jsref/jsref_obj_array.asp.

## Ideas for Quick first Exercises

TIP: Use your browser's developer console to get the syntax of a command right, then paste it into the `init`

1. Add some code to the `init` method to implement hello world.
2. Write a program that draws a design on the canvas.
3. Write a program that asks the user their favorite color, then turns the canvas that color.
4. Write a program that repeatedly asks the user to pick a number from 1 to 10 until they guess the right number. Then give them a display on the canvas to celebrate their victory.
5. Write a program that has a dialog with the user, then displays a pay stub or receipt.

## String and Math Functions in JavaScript

- **Math:** JavaScript has a built-in `Math` object with all kinds of useful functions like `Math.random`, `Math.max`, and so on. http://www.w3schools.com/jsref/jsref_obj_math.asp
- **Strings:** JavaScript also has a built-in `String` object which all strings are extensions of. It provides methods like `s.charAt`. http://www.w3schools.com/jsref/jsref_obj_string.asp

## Images and Sound

If you want to use images and sounds you have to create an object for each one first. In the global variables section, try this:

```
var ping = new Audio("sounds/ping.wav");
var paddle = new Image();
paddle.src = "images/paddle.png";
```

Then in the `init` method, or in the developer console (after reloading) try this:

```
ping.play();
ping.loop=true;
ping.play();
ping.pause();
```

```
canvas.drawImage(paddle,100,100);
canvas.drawImage(paddle,100,200,200,50);
```

Note that if you have a long sound and you pause it, when you play again it will resume where it left off. The only way to reset the sound back to the beginning is to create a new object and reload the sound – e.g. `ping = new Audio("sounds/ping.wav");`.