CoderDojo HTML Game Programming Week 2

We’re going to learn a bit about programming games in HTML. This week, we’ll build on last week’s lesson to implement a simple “Hangman” game, where you guess the word by entering letters.

To start, click the CoderDojo icon along the bottom of the screen.

Last week, we looked at variables and assigning them values, like numbers or strings.

Sometimes, it’s useful to have a variable contain a set of values, like, the names of players on a team, or the possible words to guess when playing hangman. In Javascript, variables holding sets of values are called ‘arrays.’ An array A containing the values 1, 2, and 3 is written

```
var A = [1, 2, 3];
```

Besides numbers, variables can contain strings, like words “hello,” “goodbye”, “whatever.” To put a string into a word, we write

```
var S;
S = “weirdo”;
```

We can also convert from a string, to an array, in Javascript, using the string method ‘split.’ If we had a string like:

```
var S = “this is a string”;
```

We can create an array containing 4 elements: “this”, “is”, “a”, “string” by writing

```
var A = S.split(“ “);
```

Finally, a bit of a trick. if we have a string “rat”, we can create an array R containing 3 elements “r” “a” and “t” by writing

```
var S = “rat”;
var R = S.split(“”);
```

Now, copy the following into a file, and run it to see how it works.

```
<html>
<title>Hangman</title>
<p>The computer will pick a word from a list</p>
<p>Your goal is to determine the word by guessing letters</p>
<p>Each time you guess, if you guess wrong, you lose a life!</p>
<p>Guess the word before running out of lives. If you don't, you lose!"</p>
<button onclick="playGame()">Press to start!</button>

<body>

    <script>
        function playGame() {
            var words = [
                "monkey",
                "dingle",
                "wombat",
                "knitting"
            ];
            //Pick a word at from "words" at random
            var originalWord = words[Math.floor(Math.random() * words.length)];
            //and turn it into an array
            word = originalWord.split("");
            var answerArray = [];
            for (var i = 0; i < word.length; i++) {
                answerArray[i] = "_";
            }
            var remainingLetters = word.length;
            var remainingLives = 5 + Math.floor(Math.random() * words.length);
            while (remainingLetters > 0 && remainingLives > 0) {
                alert(answerArray.join(" ") + " Remaining lives: " +
remainingLives);
                //get a guess!
                var guess = prompt("Guess a letter, or click Cancel to stop");
                if (guess == null) {
                    break;
                } else if (guess.length != 1) {
                    alert("Please enter a single letter!");
                } else {
                    var correctGuess = false;
                    //Check the guess, maybe the game's over
                    for (var j = 0; j < word.length; j++) {
                        if (word[j] == guess) {
                            correctGuess = true;
                            answerArray[j] = guess;
                            word[j] = " ";
                        }

                    }
                    if (correctGuess == false) {
                        remainingLives--;
                    } else {
                        remainingLetters--;
                    }
                }
            }
            if (remainingLetters == 0) {
                alert("Good job! The answer was " + originalWord);
            } else {
                alert("Sorry, the answer was " + originalWord);
            }
        }
    </script>
</body>

</html>
```

This program makes a lot of use of the ‘if’ statement. The **if** statement controls the flow of the program, the sequence of steps that the program will follow next, after testing the **condition** within the parentheses. So, in the program, we see

```
    if (remainingLetters == 0) {
                alert("Good job! The answer was " + originalWord);
            } else {
                alert("Sorry, the answer was " + originalWord);
            }
```

this means, if the contents of the variable remainingLetters equals zero, show the ‘good job!’ alert. Otherwise (**else**) show ‘Sorry.’

Easy: Change the words – add more words, what do you have to change for this?

Medium: Allow letters entered in upper-case (‘A’, ‘B’, … as compared to ‘a’, ‘b’ …) to work. You might need the method toLowerCasee. See if you can find this method on google by searching for ‘javascript toLowerCase method’

Hard: Allow letters entered in upper-case to work, but implement toLowerCase in a function using an array (or whatever works!)