

CSCI 202, Spring 2017, Lab # 5

Geoffrey Matthews

May 2, 2017

Due date: Midnight, Monday, May 15.

Goals: Learn about Javascript loops and arrays, and the DOM.

Zip your folder: Put all your files (including the supporting files) in a single folder and make a compressed (zip) archive of the folder.

Files to turn in:

- The `images` folder, and the HTML, CSS, and Javascript files, `penguins.html`, `penguins.css`, and `penguins.js`.
- The `images` folder, the HTML file `penguins.html`, and the CSS file, `penguins.css` are NOT TO BE CHANGED from the ones supplied on the git repository.
- You will create a javascript file, `penguins.js`, that will do all of the creation and styling of the penguins (and the yeti).

Lab steps:

- This lab will be a continuation of the previous lab. The idea is to make a *lot* of penguins (and one yeti), but with much less work. This version will support any number of penguins, and will place the one yeti in a random position, different every time the page is loaded.
- Examine the `penguins.html` and `penguins.css` files. You'll notice there's very little in them.
- Notice that the width of this version the full page, so we can have many more penguins in a row. We also had to fix the styling of the title image.
- Notice that there are *no* penguins (or the yeti) in the HTML file, and *no* styling of the penguins (or the yeti) in the CSS file. Javascript will do everything.
- Your Javascript file will declare and initialize the following variables at the top:

```
var penguinVisited = [];  
var yetiVisited = false;  
var numPenguins = 18;  
var moundImages = ['url("images/mound_1.png")', ...]  
var hoverImages = ['url("images/mound_1_hover.png")', ...]  
var penguinImages = ['url("images/penguin_1.png")', ...]
```

- It will also set the `window.onload` function to `loadPenguins`.
- You will also define the following functions:

```
function loadPenguins()  
function penguinOver()  
function penguinOut()  
function penguinClick()  
function yetiOver()  
function yetiOut()  
function yetiClick()
```

- The `loadPenguins` function (which also loads the yeti), will be the most substantial. It has the following tasks:
 1. Find the `gameholder` element of the web page.
 2. Pick a random number between 0 and `numPenguins - 1`.
 3. In a `for` loop, it will create each penguin object. If the loop index matches the random number, it will create a yeti instead.
 4. For each penguin (or yeti), do the following:
 - (a) Set its `visited` state to `false`. (Use the appropriate variable or array.)
 - (b) Create a `div` element.
 - (c) Set the appropriate `mouseover`, `mouseout` and `click` functions.
 - (d) Set the background image.
 - (e) Set the `width`, `height`, and `float` style properties.
 - (f) Append the penguin (or the yeti) to the `gameholder` element.
 - (g) If it's a penguin (not a yeti), set a `num` property of the penguin object to `i`, the `for` loop variable. This will give each penguin a unique number.
- The remaining six Javascript functions should be obvious, except that we have to pick a different image from the eight provided images for each penguin. If we have more than eight penguins (which we do), we will have to recycle the numbers. We do this finding the `num` property of the penguin, for which we can use `this.num`, and using `(this.num % 8)` to index the arrays.

Acknowledgements: Supporting files and ideas for this lab came from
<https://googlecreativelab.github.io/coder-projects/>