



Week 2 Assessment

1. DOM Manipulation

Use this HTML for the following questions:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>My Site</title>
</head>
<body>
  <nav>
    <ul>
      <li><a href="/">Home</a></li>
      <li><a href="/about">About</a></li>
      <li><a href="/contact">Contact</a></li>
    </ul>
  </nav>
  <div class="section">
    <h1>Welcome to my site!</h1>
    
    <p>Words and <a href="/">links</a></p>
  </div>
  <div class="footer">
    <small>Copyright General Assembly</small>
  </div>
</body>
</html>
```

Using jQuery...

1.1 Replace "Welcome to my site!" with "Hello World!"

1.2 Replace the current image with this image: "earth.jpg"

1.3 Add a new link to the nav that points to the `"/blog"` section of the website.

1.4 When a user clicks on one of the `nav` links, do not redirect them. Instead, trigger an `alert` that says "Sorry, this page is under construction". **Bonus:** Display the page name ("Home", "About", etc.) in the alert.

How hard was this problem? Hard <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Easy	How confident are you in your solution? It's shaky <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Nailed it!
--	---

2. JavaScript OOP

2.1 Create a JavaScript object constructor for a `Person`. Every instance of `Person` should have the following properties:

- first name
- last name
- number of siblings
- states lived in

2.2 Using your `Person` constructor, create a `greet` method that returns a string, e.g. "Hi, my name is Steve Wozniak and I have 3 siblings!"

2.3 Create a new instance of `Person` , and demonstrate how you would call the `greet` method.

2.4 Create a method called `placesLived` that prints all the states a person has lived in, one per line.

How hard was this problem? Hard <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Easy	How confident are you in your solution? It's shaky <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Nailed it!
--	---

3. AJAX

3.1 Describe the purpose of each part of the following URL:

<https://www.youtube.com/watch?v=y8Kyi0WNg40&t=1s>

3.2 Juggler Supply Co. has an api with the following documentation:

API Endpoint: <http://jugglersupply.co>

Search: Returns all matching juggling supply products.

- Path: `/api/supplies/search`
- Parameters:
 - `q` - search query term or phrase.
 - `limit` - (optional) number of results to return. Default 10.
 - `offset` - (optional) results offset. Default 0.
 - `danger` - (optional) limit results to supplies with this danger level (`safe` , `medium` , `superdanger`).
- Example response:

```
js { "data": [ { "name": "Simple Balls", "danger": "medium"}, { "name": "Deceptively Simple Balls", "danger": "superdanger"} ] }
```

Random: Returns a random juggling supply product.

- Path: `/api/supplies/random`
- Example response:

```
js { "data": { "name": "Bunnies", "danger": "medium" } }
```

Top: Returns the most popular juggling supply products.

- Path: `/api/supplies/top`
- Parameters:
 - `limit` - (optional) number of results to return. Default 2.
 - `danger` - (optional) limit results to supplies with this danger level (`safe` , `medium` , or `superdanger`)
- Example response:

```
js { "data": [ { "name": "Frank's Flaming Knives", "danger": "superdanger" }, { "name": "Hilda's Hackkeysacks", "danger": "safe" } ] }
```

Give a jQuery code example of how a front-end developer would request the top 5 juggling supply products. **Bonus:** Use the response data to render a list of product names to the body of the page.

How hard was this problem?	How confident are you in your solution?
Hard <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Easy	It's shaky <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Nailed it!