Week 7 Lab: Using JSON Files

- Due May 19, 2024 by 11:59pm
- Points 100
- Submitting a text entry box or a file upload
- Available May 13, 2024 at 12am - Jun 13, 2024 at 11:59pm

This assignment was locked Jun 13, 2024 at 11:59pm.

Before you begin:

You have been provided with the HTML file (ch08-proj02.html) that includes the markup (as well as images and stylesheet) for the finished version. Preview the file in a browser. You will be replacing the markup for the three country boxes with two JavaScript loops (one contained within the other) and the document.write() function to output the equivalent markup.

The CSS styling has been provided. You only need to output the correct HTML. The three images are contained within <article> elements. The color blocks are elements whose background-color style is set via inline CSS using the hex property from the colors array in the JSON data. The image filename is contained within the filename property in the JSON data.

Video Walkthrough

Necessary Files:

You are currently logged into Student View

Resetting the test student will clear all history for this student, allowing you to view the course as a brand new student.

Reset Student

Leave Student View

1. In the file ch08-proj02.js, convert the JSON string in photo-data.js into a JavaScript array object using JSON.parse().

```
// first transform JSON data into javascript array
const photos = JSON.parse(content);
```

2. Then write a loop that iterates through the photos array and calls outputCard(), which you will create in the next step. Pass a single photo object to outputCard().

```
3   // now loop thru this array and output cards for each photo.
4   for (let p of photos) {
5       outputCard(p);
6   }
7
```

3. Create a function named outputCard() that is passed a single photo object. This function is going to generate the markup (using document.write) for a single photo card (a card is a term often used to describe a rectangle containing an image then text below it).

```
7
8   function outputCard(photo) {
9       document.write('<article>');
10      document.write(`<img src="images/${photo.filename}" alt="${photo.title}" />`);
11      document.write('<div class="caption">');
12      document.write(`<h2>${photo.title}</h2>`);
13      document.write(`<p>${photo.location.city}, ${photo.location.country}</p>`)
14      outputColors(photo.colors);
15      document.write('</div>');
16      document.write('</article>');
17
18  }
19
20
```

This function will call two other functions (described below): outputColors() and constructColor().

4. Create a function named outputColors() that is passed the colors array for a single photo. It will loop through the colors and call constructColor() for each color. The string returned from constructColor() will be passed to document.write() .

You are currently logged into Student View

Resetting the test student will clear all history for this student, allowing you to view the course as a brand new student.

Reset Student

Leave Student View

```
function outputCard(photo) {
    document.write('<article>');
    document.write(`<img src="images/${photo.filename}" alt="${photo.title}" />`);
    document.write('<div class="caption">');
    document.write(`<h2>${photo.title}</h2>`);
    document.write(`<p>${photo.location.city}, ${photo.location.country}</p>`)
    outputColors(photo.colors);
    document.write('</div>');
    document.write('</article>');

        function outputColors(colors) {
        document.write('<h3>Colors</h3>');
        for (let c of colors) {
            document.write(constructColor(c));
        }
    }

}
```

5. Create a function named constructColor() that is passed a single color object. It will return a string containing the markup for a single color. It will also call constructStyle() for the background and text color.

```
        function outputColors(colors) {
        document.write('<h3>Colors</h3>');
        for (let c of colors) {
            document.write(constructColor(c));
        }
    }

    function constructColor(color) {
    let style = constructStyle(color);
    let tag = `<span style="${style}">${color.name}</span>`;
    return tag;
    }
```

6. Create a function named constructStyle() that is passed a single color object. It will return a string containing the CSS for the background and text color. The text color will only need to be specified if the luminance property value is less than 70. In that case, change the text color to white.

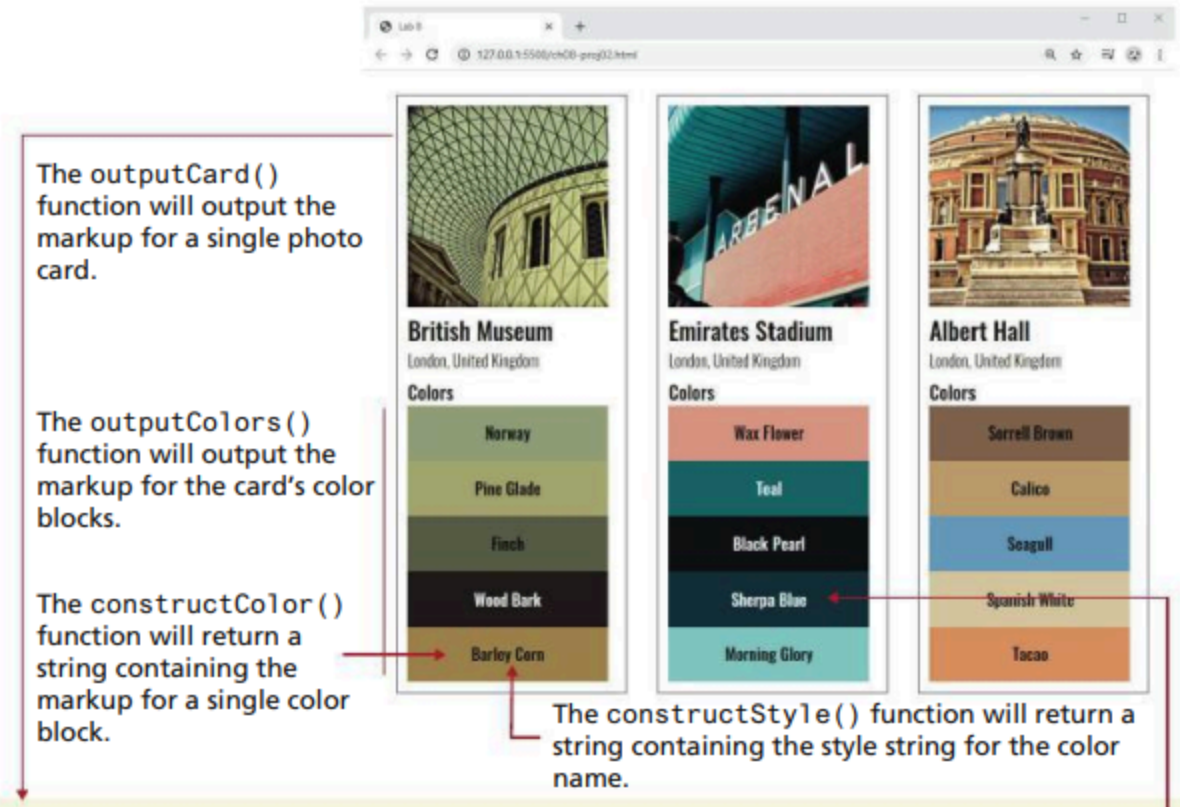
```
    function constructColor(color) {
    let style = constructStyle(color);
    let tag = `<span style="${style}">${color.name}</span>`;
    return tag;
    }

    function constructStyle(color) {
        let spanStyle = `background-color: ${color.hex}; `;
        if (color.luminance < 70)
        spanStyle += 'color: white; ';
```

You are currently logged into Student View

Resetting the test student will clear all history for this student, allowing you to view the course as a brand new student.

Reset Student

Leave Student View




```
<article>
  <img src="images/5855729828.jpg" alt="British Museum">
  <div class="caption">
    <h2>British Museum</h2>
      <p>London, United Kingdom</p>
      <h3>Colors</h3>
      <span style="background-color:#a9b490">Norway</span>
      <span style="background-color:#bab984">Pine Glade</span>
      <span style="background-color:#71735c">Finch</span>
      <span style="background-color:#332625;color:white">Wood Bark</span>
      <span style="background-color:#b99a5d">Barley Corn</span>
  </div>
</article>
```

Change the text color of the color name based on the `luminance` property.

You are currently logged into Student View

Resetting the test student will clear all history for this student, allowing you to view the course as a brand new student.

Reset Student

Leave Student View