

SE 3316A Web Technologies Lab #1: Due Friday, September 22, 5:00 pm

Deadlines:

Submission deadline: Friday, Sep. 22, 5:00 pm. Late submissions accepted until Sunday Sep. 24 11:55pm (2% penalty).

Demonstration deadline: End of assigned lab section.

Change Log

1. Added more details and a submission link to Github classroom in "Submission Instructions". 2:45pm, Sep-10
2. Added command to checking server logs to "Tips". 3pm, Sep-13.
3. Added a howto video to use Github PAT, 3:30pm, Sep-13.

Objectives:

- A. Create a HTML document that shows any 20 countries from the data provided.
 - a. Show the flag, name, currency and all regions for each country.
 - b. Use an unordered list to display each item and use CSS to visually group each item (e.g. boundary or a fill colour) without the typical bullet in a list.
 - c. Show the Wikipedia page as a link for each country.
 - d. Entries for the 20 countries must be hand-coded in HTML. They should not be created dynamically with JavaScript.
- B. Display two search boxes at the top of the page. One to search for countries by the name and the other to search by currency code.
- C. Use JavaScript to validate input for the two search boxes. Box 1 should accept no more than 20 characters and not accept numbers. Box 2 should only accept three upper case letters A-Z.
- D. "Enter" key or the "Search" button on each box should display a pop-up message with country names and currencies of all matches (up to 5). You may create a JSON data structure with relevant data for this purpose.
- E. Ensure that all CSS and JavaScript are in separate files.
- F. Use Git to keep track of your work and upload to Github as a private repository.
- G. Exclude project assets (images, data etc) from Git repository.
- H. Create a server on AWS and set up a HTTP server to deploy your HTML page via a public URL.

If you do not have a Github account, please sign-up with github.com using your Western email. All your repositories on Github must be private repositories. Each commit should be meaningful (adding specific functionality) and must be accompanied by a meaningful commit message.

Rubric (out of 100):

1. Basic HTML page shows flags, names, regions and currency for 20 countries: 30
 - a. Not adhering to HTML5 standard: -5
 - b. Not using list elements: -5
 - c. Not providing workable links for information from external sources: -5
2. CSS styling to make each item visually distinct: 15
 - a. Inline or embedded CSS: -5
 - b. Not separating structure from presentation: -5
3. HTML+JavaScript for search functionality: 25

- a. Inline or embedded JavaScript: -5 (minimal inline code needed to attach callback functions to events is acceptable. See appendix for an example)
- 4. Access the page with full functionality on a public URL: 30
- 5. Code management with Git
 - a. Less than 5 commits: -5
 - b. No meaningful commits: up to -10
 - c. No meaningful commit messages: up to -5
 - d. Not adhering to Git repository naming convention: -5
 - e. Not using a proper **.gitignore** file to ignore images: -5
 - f. Images included in Git repository: -10
 - g. Not using Git pull to deploy code to server: -5
- 6. Logistics
 - a. Repository name not in required format: -5
 - b. No zip file or git log: not graded.
 - c. Code is not attached as a zip file or it contains content that is not in the Git repository¹ or is missing content that is in the repository: -10
 - d. Images in zip file: -10

Workflow:

1. Create a new private Git repository on Github called "se3316-lab1-xyz" (all lowercase) where "xyz" is your Western email ID (without @uwo.ca part).
2. Clone that repository on your workstation/laptop to create a local working directory.
3. Copy the skeleton HTML file to the working directory and make an initial commit.
4. After each unit of work, test it using your browser and commit at the end.
5. Push your project to Github.
6. Use "Howto" documents and videos to set up a HTTP server using AWS. Pull your Github project to the folder where HTML files are being served.
7. Make modifications on the server (if necessary) and push your changes to Github.

Schedule:

You may complete it at home. TAs will be available in the lab during lab hours to answer any questions that you may have.

Submission Instructions:

Please carefully read the instructions and strictly follow them. Your grade depends on it.

1. Ensure that your repository is private.
2. Use a proper **".gitignore"** file so that only the files that you edit are in your repository.
3. Make frequent commits with an appropriate commit message.
4. Ensure that you understand the principles behind your code.
5. Ensure that Github contains the latest version of your code.
6. Copy the output of command **"git log"** and paste that onto the submission page (Assignments section) on Owl.
7. Download your repository as a zip file from Github and submit as an attachment. Please don't create a zip file of your desktop copy. Please don't use any other archive format.
8. Submit your lab to <https://classroom.github.com/a/LgA9HPW9> (See [detailed instructions](#)).

¹ If you zip the file on desktop, it may often include hidden files (particularly on Mac) or the .git folder itself.

9. Demonstrate your lab (on a public URL) before the demonstration deadline. You may set up the public URL after submission on Owl. As long as the changes only involve deployment issues, it is acceptable.

Penalties will apply for not following the naming convention or any of the submission steps.

References

- 1.

Implementation Tips

- Add an array of countries with name and currency to JavaScript code to implement search functionality
- Creating a **".gitignore"** file: On Github, select "none" as the gitignore template. Edit this file and add the line **"images/"** to ignore all files in the "images" folder. A line such as **"*.log"** ignores all files at the top level with names that end with **".log"**. See [documentation on gitignore](#) for more details.
- Troubleshooting: For a public URL, if it times out, it is a network or the endpoint issue. Check if the AWS server is provisioned and running, the web server is installed and running.
- Troubleshooting: Checking access and error logs on AWS server can yield important diagnostic info. Use **"sudo tail -n 10 /var/log/nginx/error.log"** to see server errors. In the same location, **"access.log"** contains a log of all page accesses. This command shows the last 10 lines of the specified file.
- How to use Github commands using a Personal Access Token (PAT):
<https://www.youtube.com/watch?v=kHkQnuYzwoo>

Frequently Asked Questions

Q: Images are not in my git repository. How can I get them to the server since "git pull" will not get them?

A: Use other file copy software such as FileZilla (Win or Mac), WinSCP or scp (Mac) to copy images.

Q: Since images are not in my repository, they will not be in my submission. Is that OK?

A: Yes.

Q: I pulled my repository to the server. But it is not showing the page and/or I get an error.

A: Most likely cause is that the file paths are wrong. Please review the guide on mapping URL to file paths and see if you can answer the questions. Review your setup based on your new knowledge.

Q: Can I use [insert a method for manipulating DOM] to show results?

A: No. Please use a JavaScript Alert window to show results.

Appendix

Sample HTML5 file:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>SE 3316 Lab 1</title>
<meta name="description" content="">
```

```
<meta name="author" content="">
<link rel="stylesheet" href="">
</head>
<body>
<!-- Place your content here -->
<h1>SE 3316 Lab 1</h1>
<p>Hello World!</p>

<!-- Embedded JavaScript go here -->
</body>
</html>
```

Sample JavaScript for attaching an event handler

```
<input type="text" onkeypress="myFunction()">
```