Overview

The goals of this assignment are:

1. Practice responding to events through JavaScript DOM manipulation.
2. Practice retrieving data from remote systems using AJAX.
3. Processing JSON encoded data retrieved from a remote system and displaying it using JavaScript DOM manipulation.

For the required AJAX call, the use of jQuery or XMLHttpRequest objects is not permitted, please use the fetch() method as demonstrated in notes and class.

The academic integrity policy will be enforced, as usual. Please ask if you are uncertain.

NOTE: if the assignment is not submitted to both CSUnix and to Canvas it will receive a grade of 0.

Content

In this assignment we will simulate the kind of processing you will do next semester when you retrieve records from databases using PHP. From our point of view, we don't care how the JSON string was created, we only need to work with it once it arrives in our client-side system.

We haven't done an exact example of this together, but it has a lot in common with printing variables from an array in a loop that we did in first semester. What does "<td>Hello</td>" mean? Nothing in a Python program, and not much to JavaScript either, it's a string. When it becomes part of the DOM, the browser turns it into a cell in a table. This is the idea we've discussed a few times: **syntax** versus **semantics** – the structure versus the meaning.

You are going to be doing exactly that: turning a string into HTML. Except where noted you may create empty HTML elements and change the innerHTML to achieve your goals but top marks are reserved for using CreateElement and manipulating the DOM directly. You may still need to modify attributes on those elements, those details are up to your ingenuity.

Create a page with 3 buttons labelled "First", "Second", "Third", and an input element on it. You can pick the type of input element you feel best solves the problem.

When any button is pressed you will be sending an AJAX request to https://csunix.mohawkcollege.ca/~adams/10259/a6_responder.php below you will find the details of what each request should do. To work correctly with CSUNIX the request must originate on CSUNIX. If you try to test your work from your local computer you will see a CORS error, a cross site scripting error, will happen. To develop your solution, I suggest putting the URLs for GET requests into your brower's URL bar and putting the returned strings into a variable. This way you can test your code without uploading it. For the POST request you can use a similar technique or simply test immediately on CSUNIX.

When the **first button** is pressed send the AJAX request with no parameters. You should expect the response to be text. Create a h1 element with this text and your student number. Make the text centered in a container that fits across the entire width of the page.

When the **second button** is pressed send an AJAX request along with the value of the input element as a GET parameter named “choice”. The valid values allowed by the server script are “mario” or “starwars”. For example:
https://csunix.mohawkcollege.ca/~adams/10259/a6_responder.php?choice=mario?choice=mario You should expect the response to a JSON string. The string will encode an array of objects. You will receive at least 1 and at most 3 elements in the array. For each element in the JSON array create a div to hold the content. Each attribute of the object will be a string, to keep things simple: a name, a url, and a series. At the top of the div put the series string in an h2 tag inside the div. Below that show the image that is linked in the url. Finally, below that put the name. The divs should fill 100% of the width of the screen: if there are 2 elements then the divs will be 50% each, if there’s 3, 33% each. Be warned you may have to provide height and width parameters to force the images to resize. Do not leave a blank div, fill the entire area. It should resemble (but does not have to be the same as) these examples:

My AJAX Assignment - Student # 000123456

Smash Bros	Yoshi's Wooly World
	
Dark Samus	Green Yarn Yoshi

Game trademarks and copyrights are properties of their respective owners. Nintendo properties are trademarks of Nintendo. © 2019 Nintendo.

Series	Name	Link
Super Smash Bros.	Donkey Kong	https://www.nintendo.com/amiibo/detail/donkey-kong-amiibo-super-smash-bros-series/
Super Smash Bros.	Fox	https://www.nintendo.com/amiibo/detail/fox-amiibo-super-smash-bros-series/
Super Smash Bros.	Kirby	https://www.nintendo.com/amiibo/detail/kirby-amiibo-super-smash-bros-series/
Super Smash Bros.	Link	https://www.nintendo.com/amiibo/detail/link-amiibo-super-smash-bros-series/
Super Smash Bros.	Mario	https://www.nintendo.com/amiibo/detail/mario-amiibo-super-smash-bros-series/
Super Smash Bros.	Marth	https://www.nintendo.com/amiibo/detail/marth-amiibo-super-smash-bros-series/
Super Smash Bros.	Peach	https://www.nintendo.com/amiibo/detail/peach-amiibo-super-smash-bros-series/
Super Smash Bros.	Pikachu	https://www.nintendo.com/amiibo/detail/pikachu-amiibo-super-smash-bros-series/
Super Smash Bros.	Samus	https://www.nintendo.com/amiibo/detail/samus-amiibo-super-smash-bros-series/
Super Smash Bros.	Villager	https://www.nintendo.com/amiibo/detail/villager-amiibo-super-smash-bros-series/
Super Smash Bros.	Wii Fit Trainer	https://www.nintendo.com/amiibo/detail/wii-fit-trainer-amiibo-super-smash-bros-series/

Game trademarks and copyrights are properties of their respective owners. Nintendo properties are trademarks of Nintendo. © 2019 Nintendo.

When the **third button** is pressed send an AJAX request with the value of the input element as a POST parameter. The valid values allowed by the server script are “mario” or “star wars”. You should expect the response to be a JSON string. The string will encode an array of objects. You will receive at least 1 and at most 15 elements. Put the elements returned in a table below the pictures, each object containing the values for 1 row. The table tag may be hard coded in your html, but the table rows and cells may not be hard coded and must be dynamically generated by your JavaScript code.

Below the table there should be a copyright notice that matches the content.

Place an appropriate copyright notice below the images depending on whether the input was for Star Wars or for Mario.

The copyright notice for “mario” content is:

Game trademarks and copyrights are properties of their respective owners. Nintendo properties are trademarks of Nintendo. © 2019 Nintendo.

The copyright notice for “starwars” is:

Star Wars © & TM 2022 Lucasfilm Ltd. All rights reserved. Visual material © 2022 Electronic Arts Inc.

My AJAX Assignment - Student # 000123456

Smash Bros	Animal Crossing	Yoshi's Wooly World
		
Dark Samus	Tom Nook	Green Yarn Yoshi

Game trademarks and copyrights are properties of their respective owners. Nintendo properties are trademarks of Nintendo. © 2019 Nintendo.

Series	Name	Link
Super Smash Bros.	Donkey Kong	https://www.nintendo.com/amiibo/detail/donkey-kong-amiibo-super-smash-bros-series/
Super Smash Bros.	Fox	https://www.nintendo.com/amiibo/detail/fox-amiibo-super-smash-bros-series/
Super Smash Bros.	Kirby	https://www.nintendo.com/amiibo/detail/kirby-amiibo-super-smash-bros-series/
Super Smash Bros.	Link	https://www.nintendo.com/amiibo/detail/link-amiibo-super-smash-bros-series/
Super Smash Bros.	Mario	https://www.nintendo.com/amiibo/detail/mario-amiibo-super-smash-bros-series/
Super Smash Bros.	Marth	https://www.nintendo.com/amiibo/detail/marth-amiibo-super-smash-bros-series/
Super Smash Bros.	Peach	https://www.nintendo.com/amiibo/detail/peach-amiibo-super-smash-bros-series/
Super Smash Bros.	Pikachu	https://www.nintendo.com/amiibo/detail/pikachu-amiibo-super-smash-bros-series/
Super Smash Bros.	Samus	https://www.nintendo.com/amiibo/detail/samus-amiibo-super-smash-bros-series/
Super Smash Bros.	Villager	https://www.nintendo.com/amiibo/detail/villager-amiibo-super-smash-bros-series/
Super Smash Bros.	Wii Fit Trainer	https://www.nintendo.com/amiibo/detail/wii-fit-trainer-amiibo-super-smash-bros-series/

Game trademarks and copyrights are properties of their respective owners. Nintendo properties are trademarks of Nintendo. © 2019 Nintendo.

NOTES: that there is no guarantee the user will not change the input between button presses, so how will you make the two pieces of content match one another? To simplify the overall situation, you may choose to disable the input tag once one of the 2nd or 3rd buttons is pressed to avoid tracking state and managing a more complicated situation.

The use of Bootstrap to make things look nicer is recommended, indeed, **you might find it simplifies the layout dramatically**.

Responsive design is not required for this assignment but is always welcome.

Handing In

To eligible for grading the following steps must be done:

1. Name your html file as **index.html** and save all files in your CSUnix **public_html/private/10259/a5** directory. Note that paths are case sensitive, a5 is not the same as A5.
2. Validate your HTML and CSS files are standard compliant.

   a. Check your HTML and CSS files using the W3C validators. Ensure that there are no errors. Minor warnings will not result in a penalty, but any serious warning should be corrected. Please use the discussion group to get help to resolve the warnings and errors, or to clarify if you are uncertain.
3. Ensure your JavaScript runs without errors output to the console. You may have debug statements output to the console without penalty (e.g. deliberate console.log() messages are ok)
4. Make sure you also follow the instructions in the Assignment Documentation Standard, including the student's name a number in each file you submit, including your CSS files.
5. Submit your work to the LMS by copying each of your html, css, and js files to add a ".txt" extension to them. For example, "index.html" becomes "index.html.txt". Do not archive your files (no renamed zip files).

Evaluation

Approximately:

Technical requirements: 60%

Aesthetics and presentation: 30%

Documentation and validation: 10%

See the rubric for details.

Late days may be used on this assignment if you have any remaining. Please direct all questions to your professor, or to the discussion group on the LMS.