PIC 40A: Homework 8 (due 6/9 at 5pm)

Like on previous homeworks, it is important that you meet the following requirements.

- You must upload your files to **Gradescope** before the deadline.
- You must upload your files to the **PIC server** in the appropriate directory before the deadline.
- Both submissions must be **identical** (down to the character).
  Never make changes to the PIC server submission after the deadline.
  (We can see when a file was last modified.)
- You must tell us (me and the grader) your **PIC username**.
- You must **validate** your HTML using https://validator.w3.org/.
  Ideally, with PHP, you should check that, after removing the PHP parts, the remaining HTML validates.

In this assignment you will submit twelve(!) files...

1. `README.txt`. This will contain your PIC username.
2. `login.php`, `h_password.txt`, `welcome.php`, `username.js`, and `welcome.js`.
   These are files that you made on previous homeworks. No changes are necessary.
3. `shut_the_box.php`.
   This is a file that you made on a previous homework. It'll probably need one additional line.
4. `shut_the_box.js`.
   This file will build upon the file of the same name that you submitted on a previous homework.
5. `scores.txt`. A blank text file which will be used for storing scores.
6. `score.php`. A script for writing a new score to `scores.txt`.
7. `scores.php`, `scores.js`. Files for a self-updating score page.

As mentioned above, you should submit all files to Gradescope before the deadline.
You should also submit the files to the PIC server. Save them in the directory

`/net/laguna/???...???/your_username/public_html/HW8`

(in the folder `HW8` within `public_html`). **You'll also need to create a folder called "sessions" with 755 permissions.** We should all be able to test your live webpage at

www.pic.ucla.edu/~your_username/HW8

Now, I am just left to tell you what I want these files to achieve. Go over the page for that!

1 `shut_the_box.html`, `shut_the_box.js`, `score.php`, and `scores.txt`

`scores.txt` should start off life as a blank text file. This is where users' scores will be stored. They will be stored in the simplest possible format...

```
user1 score1
user2 score2
user3 score3
```

When you submit your homework, make sure to reupload a blank text file to the server so that the grader does not see your previous scores from playing Shut The Box! Don't worry about submitting `scores.txt` to Gradescope.

In order to make scores appear in `scores.txt`, we need to change `shut_the_box.js` a little...

Before starting this assignment, you should have a function that looks something like

```
function finish() { // ... need to disable buttons
  alert(`Your score is ${unchecked_total}.`);
}
```

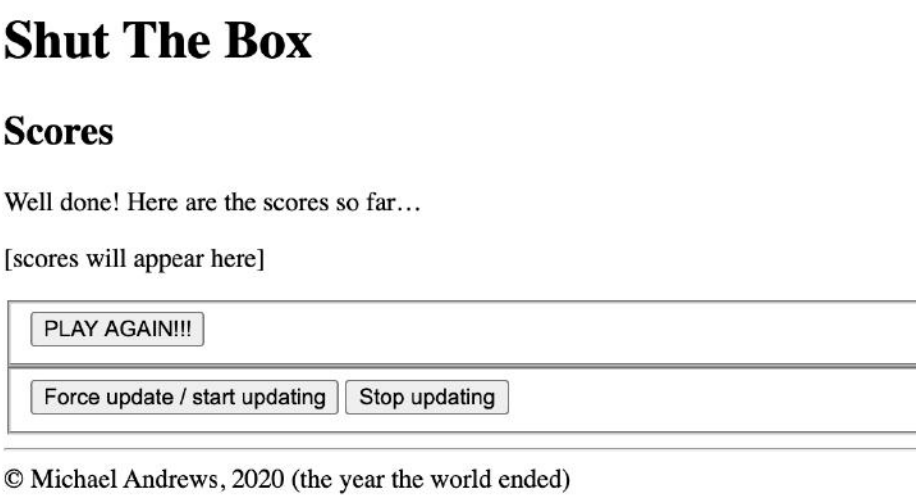
This is the function that is called by clicking "I give up / I can't make a valid move" in my incomplete solution to homework 3. This function should do more...

- It should make an AJAX request.
- The AJAX request should have its method set to POST and it should send the user's username and the user's score to the PHP file `score.php`.
- `score.php` should write the information it receives to the file `scores.txt`.

  (However, `score.php` should not write any information to `scores.txt` if opened directly.)
- **After all this has been done successfully,** we should be redirected to `scores.php`.

2 `scores.php` and `scores.js`

Here's a screenshot of what my `scores.php` looks like.



So, in terms of the HTML...

- The tab should be titled "Shut The Box".
- You should have a header displaying a heading "Shut The Box".
- You should have a section with a heading saying "Scores".
- The section should have a paragraph saying "Well done! Here are the scores so far..."
- The section should have a paragraph where the scores will be displayed. It is probably better to leave this paragraph empty originally. I just put "[scores will appear here]" for your benefit.
- There should be two `<fieldset>` elements.
  - The first should have a button saying "PLAY AGAIN!!!".
  - The second `<fieldset>` element should contain two buttons with the text indicated above.
- There should be a footer with copyright information.

In terms of the PHP...

- use PHP sessions so that if a user is not logged in, they are redirected back to `login.php`.

In terms of the JavaScript, `scores.js` should make sure the scores are updated every 8 seconds (without refreshing the page) and that the buttons have the correct functionality.

- The "PLAY AGAIN" button should redirect a user to `welcome.php`.

  We're redirecting to `welcome.php` instead of `shut_the_box.php` so that a user has an opportunity to change their username.
- "Force update / start updating" should update the scores immediately.

  After clicking this button, the scores should continue to update every 8 seconds.
- "Stop updating" should stop the scores from being automatically updated.

Bear in mind that I dealt with most of this logic in class (see `repeat.js`). You just need to implement an AJAX GET request to get the relevant information from `scores.txt` and avoid the use of the `onclick` attribute.

3 Grading

One point for each of the following...

- `scores.php` redirects to `login.php` before the user has logged in.
- After logging in, specifying a username, playing Shut the Box, and "giving up", the user is redirected to `scores.php`.
- After logging in, specifying a username, playing Shut the Box, and "giving up", something is written to the text file `scores.txt`.
- After logging in, specifying a username, playing Shut the Box, and "giving up", the correct information (the `username` and the score displayed by the `alert`) is written to the text file `scores.txt`.
- Visiting `score.php` directly does not write information to `scores.txt`.
- Having logged in, `scores.php` displays the scores so far (the contents of `scores.txt`).
- By playing another game using Chrome on another computer, one can see that the scores are updated automatically (without having to refresh) within 8 seconds.
- By playing another game using a different browser (such as Safari), one can also see that the scores are updated automatically (without having to refresh) within 8 seconds.
- The "Force update / start updating" button works.
- The "Stop updating" and "PLAY AGAIN" buttons work.

A remark about the 8-th bullet point...

I've picked Safari because it is good at revealing a potential issue with your AJAX request. When you code the user's score to be written to a text file and the redirection from `shut_the_box.php` to `scores.php`, you should assume the following...

- The user loves all web browsers equally.
- The server that the text file is stored on is very slow.
- The human playing Shut the Box is lightning fast, so they could easily click on the "OK" of an alert in a time much shorter than it takes for the AJAX request to finish.

All of this is a cryptic way to say, "use the `onload` or `onreadystatechange` property appropriately!!"