

PIC 40A: Homework 4 (due 5/15 at 5pm)

Like on homeworks 1 and 3, 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/>.

In this assignment you will submit six files...

1. `README.txt`. This will contain your PIC username.
2. `shut_the_box.html`, `shut_the_box.js`. These are files that you made for the last homework. You can improve on last week's submission if you wish, but this assignment does not require that you do so and improvements will not receive extra credit.
3. `welcome.html`, `username.js`, `welcome.js`. These are new files and the focus of the assignment.

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/HW4`

(in the folder `HW4` within `public_html`). We should all be able to view your live webpage at

www.pic.ucla.edu/~your_username/HW4/welcome.html

Now, I am just left to tell you what I want `welcome.html`, `username.js`, `welcome.js` to achieve. Go over the page for that!

Welcome

HTML

Watch the demo video. I address the following aspects of the HTML...

1. The tab should be titled “Shut The Box”.
2. You should have a header displaying a heading “Welcome! Ready to play “Shut The Box”?”
3. You should have a section with a heading saying “Choose a username”.
 - (a) The section should have a paragraph saying “So that we can post your score(s), please choose a username.”
 - (b) There should be a `<fieldset>` element containing a `<label>` element, a textbox, and a button with `value` attribute set to "Submit". The `<label>` element should be associated with the textbox.

These elements will allow a user to submit their username choice.
4. There should be a footer with copyright information.

Grading

Your HTML will be graded out of 3 points.

- Validation is worth 1 point.
- 3.(b) is worth 1 point.
- Everything else is worth 1 point.

JavaScript

1. `username.js` should contain a single function definition.

The function should be called `get_username` and have no parameters.

It should extract from `document.cookie` the value corresponding to the name `username` or return the empty string if there is not such a name.

Notes:

- `get_username` should account for when `document.cookie` returns `'username; username=mjandr'`. The similar HW2 question did not ask you to account for this scenario. Recall that the first name-value pair in this example has an empty name and value equal to `username`.
 - In a future assignment, another HTML file will need to make use of `get_username` without the clutter of the functions that you are going to define in `welcome.js`. That's why we're putting `get_username` in a separate file.
2. `welcome.js` should fill the textbox with the user's username if it is stored in `document.cookie`. (You will want to use `username.js`'s function, so you should include your JavaScript files in a sensible order.)
 3. `welcome.js` should make use of two event listeners to accomplish the following...
 - (a) Clicking on the "Submit" button triggers the behavior described in 4.
 - (b) Pressing the Enter Key while typing in the textbox triggers the behavior described in 4.(Turn the page for 4.)

4. Upon clicking on the “Submit” button or pressing the Enter Key while typing in the textbox, up to three things should happen.

- (a) The specified username should be checked to see if it is of a desired form:
- between 5 and 40 characters (inclusive);
 - does not include spaces, commas, semicolons, =, or &;
 - each character is either an alphanumeric or contained in the following string:
`!@#$%^&*()-_+[]{}: '|`~<.>/?`

When a user submits a bad username one of two things will happen:

- If the specified username is problematic because it violates one of the first two bullet points, the user should be alerted in a useful manner. For complete details, watch the demonstrational video. One example: if ", , = is submitted, the alert should say:
`Username must be 5 characters or longer.`
`Username cannot contain commas.`
`Username cannot contain =.`
- If the specified username is problematic because it violates **only** the last bullet point, the user should be alerted with the following message:
`Username can only use characters from the following string:`
`abcdefghijklmnopqrstuvwxyz`
`ABCDEFGHIJKLMNOPQRSTUVWXYZ`
`0123456789`
`!@#$%^&*()-_+[]{}: '|`~<.>/?`

Provided that an appropriate username has been chosen, it should also...

- (b) Create a new cookie with `name` equal to the string "username" and `value` equal to what the user typed. This cookie should expire in an hour. It should have the default `path`, i.e. do not specify anything about the path; `path=/` is incorrect.
- (c) Redirect to `shut_the_box.html`.

Grading

Here's how your JavaScript will be graded...

- 1. is worth 2 points.
- 2. is worth 1 point.
- 3. is worth 2 points.
- 4.(a) is worth 3 points.
- 4.(b) is worth 2 points.
- 4.(c) is worth 1 point.