PIC 40A: Homework 4 (due 2/14 at 10pm)

Like on homework 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 three files together...

1. `README.txt`. This will contain your PIC username.
2. `merch.html`. This is the easier part of the assignment.
3. `merch.js`. I expect you'll spend longest on this part 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/merch.html`

Now, I am just left to tell you what I want `merch.html` and `merch.js` to achieve. See the next page!

Merch page

HTML

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

1. (a) The tab should be titled “Our Merchandise”.
   (b) You should have a header displaying a heading “Our Merchandise”.

2. You should have a section with...

   (a) a header with a general title of the items shown. I choose “Hand Knitted items.”

   (b) Then you should have a `<p>` element with the description:

   > Please have a look around. Our new members are awarded with $20,00 in credit. You can add credit at anytime with a coupon code. When you want to make a purchase, please select the checkboxes of the items you wish to purchase and click the "Checkout" button below.

   You can copy or you can improve on my description, whichever you prefer.

   (c) An empty additional `<p>` element.

   (d) A `<table>` element. There is no table head or table foot. Only a table body. There should be at least 4 `<td>` elements each with merchandise information each with the following...

   - A `<img>` element with the product image which sources a url (rather than locally for submission reasons).
   - A heading with the title of the item.
   - A checkbox `<input>` element.
   - An empty `<span>`.
   - A `<p>` element with a description of the item.

3. Below the table you should include a `<fieldset>` element with...

   (a) A label that says “Coupon code:” for a textbox meant for a coupon code.

   (b) An textbox for the coupon code.

   (c) A “Checkout” button.

   (d) A lone `<p>` element.

4. There should be a footer with copyright information.

JavaScript

Watch the demo video. I address the following aspects of the JavaScript…

1. When visiting the page for the first time (or if refreshing the page), the credit should be shown as $20.00. Store credit as a variable. You'll want to update the displayed credit if the credit should change.

2. Store your prices in a container. Populate the `<span>` elements in the table with the respective prices.

3. Clicking on the image or the corresponding checkbox causes the checkbox to become `checked` or unchecked. To make clicking on an image function correctly, you should add event listeners to some of your `<img>` elements. You should not type (or copy and paste) `addEventListener` or `on` 4 times; use a `for` loop or find another way to avoid code bloating.

4. The Checkout button should be given their functionality through the use of event listeners and event handlers, **not** the `onclick` attribute of a `<input>` or `<button>` element.

   Clicking on the Checkbox button will sum the checked prices and compute the tax using `sales_total` function you wrote. There are a few changes to this function.

   (a) It will no longer return anything (technically it will return `undefined`).

   (b) If the total after tax is more than the credit, the user will be alerted they do not have sufficient credit and nothing else about the page changes.

   (c) If the credit is sufficient...

   - Instead the relevant message with the total before and after tax will be added to the lone `<p>` element at the bottom of the page. Note that the formatting will change because we want it to display as parsed HTML. So there will be some necessary substitutions for the whitespace.
   - It will update the credit on the page.
   - The `checked` checkboxes will be changed back to their unchecked state and be `disabled` so that they will not be checkable anymore by clicking on them or the corresponding image.

5. The Coupon code box* works as follows...

   (a) Pressing the Enter Key while typing in the coupon textbox or pressing the Checkout button will validate the coupon code and add the corresponding credit. Then clear the coupon box.

   - If the string is "COUPON5" then $5 is added. If the string is "COUPON10" then $10 is added. If the string is "COUPON20" then $20 is added. The page should display the current credit.
   - Entering any (valid or invalid) coupon code alone with no boxes checked should remove any possible previous message of sales in the bottom `<p>` element.
     - If a valid coupon, the credit will be updated for the valid coupon and displayed on the page.

    - If an invalid coupon code is entered when no boxes are checked, then the only thing that happens is the coupon textbox is restored to it's default.
- Entering any (valid or invalid) coupon code with boxes checked will updated credit before the purchase is attempted.
    - If funds are sufficient, a sales message will be displayed as well as the post-sale credit updated on the page.
    - If funds are insufficient, any previous sales message will be cleared and the credit will be updated with the coupon amount.

I have provided `template.js` so you can see, to some extent, how I implemented this myself. I have included all the function names and global variable names that I used.

*As this is written, a clever person could look at our js source code and figure out they can get more credit onto their account. The majority of people would not think to look at the source code to find the coupon code though. Once we learn PHP, we will see more secure ways of having validating a coupon code or password.

Grading

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

- 1. is worth 1 point.
- 2. is worth 2 points.
- 3. is worth 2 points.
- 4. is worth 0 points, since "whatever".

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

- 1. and 2. are together worth 1 point.
- 3. is worth 2 points.
- 4. is worth 6 points.
- 5. is worth 6 points.