COMP6080: Web Front-End Programming

Tutorial 9

Sidney Pham
sidney.pham@unsw.edu.au

UNSW

Week 9, Term 3 2020

Feedback

I've spent the last week marking Assignment 2 — overall everyone did great for such a long and tedious assignment. Remember that it was *meant* to be painful!

Feedback

I've spent the last week marking Assignment 2 — overall everyone did great for such a long and tedious assignment. Remember that it was *meant* to be painful! There are a few things I've spotted that I think would be worthwhile to discuss:

Feedback

I've spent the last week marking Assignment 2 — overall everyone did great for such a long and tedious assignment. Remember that it was *meant* to be painful! There are a few things I've spotted that I think would be worthwhile to discuss:

- Forms should submit when you press Enter (use `<form>` with an `input[type=submit]` or a `button[type=submit]` for this submission behaviour by default).

Feedback

I've spent the last week marking Assignment 2 — overall everyone did great for such a long and tedious assignment. Remember that it was *meant* to be painful! There are a few things I've spotted that I think would be worthwhile to discuss:

- Forms should submit when you press Enter (use `<form>` with an `input[type=submit]` or a `button[type=submit]` for this submission behaviour by default).
- Time should be in `<time>` tags. See Reddit's timestamps:

```
<time
    title="Sun Nov 8 00:33:17 2020 UTC"
    datetime="2020-11-08T00:33:17+00:00"
>
    9 hours ago
</time>
```

Feedback

I've spent the last week marking Assignment 2 — overall everyone did great for such a long and tedious assignment. Remember that it was *meant* to be painful! There are a few things I've spotted that I think would be worthwhile to discuss:

- Forms should submit when you press Enter (use `<form>` with an `input[type=submit]` or a `button[type=submit]` for this submission behaviour by default).
- Time should be in `<time>` tags. See Reddit's timestamps:

```
<time
    title="Sun Nov 8 00:33:17 2020 UTC"
    datetime="2020-11-08T00:33:17+00:00"
>
    9 hours ago
</time>
```

- Timestamp only shows the date, not time.

Feedback (cont.)

- Empty feed should say that it's empty and possibly prompt to follow other users.

Feedback (cont.)

- Empty feed should say that it's empty and possibly prompt to follow other users.
- Having both a Follow & Unfollow button or both a Like & Unlike button.

Feedback (cont.)

- Empty feed should say that it's empty and possibly prompt to follow other users.
- Having both a Follow & Unfollow button or both a Like & Unlike button.
- Alt tags not useful. Something like 'Photo posted by [username] with description [description]' would be good.

Feedback (cont.)

- Empty feed should say that it's empty and possibly prompt to follow other users.
- Having both a Follow & Unfollow button or both a Like & Unlike button.
- Alt tags not useful. Something like 'Photo posted by [username] with description [description]' would be good.
- Clickable items should have affordances like `cursor: pointer` on the `:hover` pseudo-class. This should be done by using the appropriate HTML elements (e.g. `<button>` or `<a>`)!

Feedback (cont.)

- Empty feed should say that it's empty and possibly prompt to follow other users.
- Having both a Follow & Unfollow button or both a Like & Unlike button.
- Alt tags not useful. Something like 'Photo posted by [username] with description [description]' would be good.
- Clickable items should have affordances like `cursor: pointer` on the `:hover` pseudo-class. This should be done by using the appropriate HTML elements (e.g. `<button>` or `<a>`)!
- No feedback upon successful actions (e.g. editing profile, deleting post).

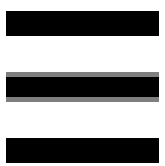
Exercise: Hamburger Menu Icon

We're going to do a simple SVG exercise: recreating a hamburger menu icon!

Exercise: Hamburger Menu Icon

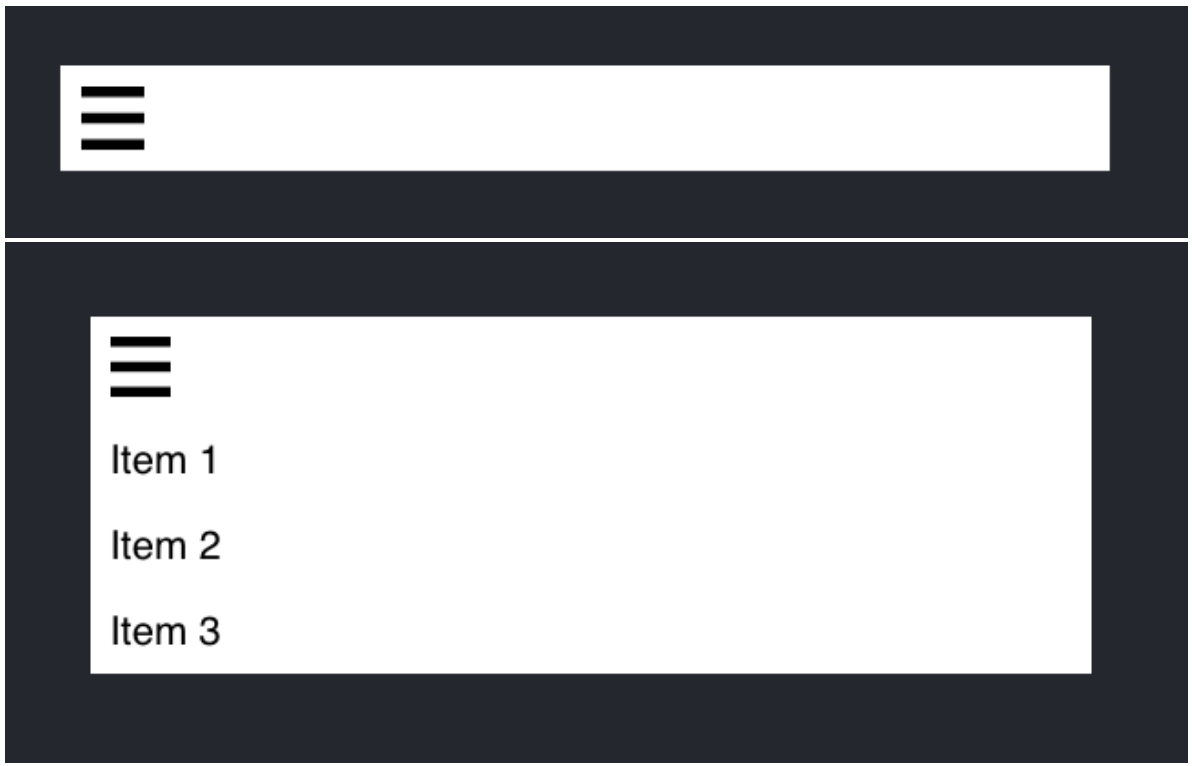
We're going to do a simple SVG exercise: recreating a hamburger menu icon!

> *Create an SVG element in `exercise/src/svg.html` for a hamburger menu icon that looks identical to this. The viewbox you should use is `0 0 100 100`, and the SVG should have dimensions 50 x 50 pixels.*



Exercise: A Simple Menu Bar

For this exercise, we will be building a menu bar. It consists of a hamburger icon that, when clicked, opens a menu with a list of links. Build the `MenuButton` and `MenuItem` components with accompanying component tests.



`MenuButton`

This component will take two inputs:

- `open` boolean value which indicates whether the menu is open or closed
- `onClick` event handler which is triggered when the button is clicked

`MenuButton`

This component will take two inputs:

- `open` boolean value which indicates whether the menu is open or closed
- `onClick` event handler which is triggered when the button is clicked

To ensure the menu is accessible, the button should have an `aria-label` attribute (since we are using an image button without text). It should also have an `aria-expanded` attribute that indicates whether the menu is expanded or collapsed.

`MenuButton`

This component will take two inputs:

- `open` boolean value which indicates whether the menu is open or closed
- `onClick` event handler which is triggered when the button is clicked

To ensure the menu is accessible, the button should have an `aria-label` attribute (since we are using an image button without text). It should also have an `aria-expanded` attribute that indicates whether the menu is expanded or collapsed.

You may use your SVG from before or `images/hamburger.png`.

`MenuItem`

This component will take two inputs:

- `title` string value which is the text to be displayed in the menu item
- `onClick` event handler which is triggered when the button is clicked, with `title` as an input parameter

Note: the `onClick` event handlers of `MenuButton` and `MenuItem` are **not** the same.

Finally, the `Menu` Component

Now, use the previously constructed `MenuButton` and `MenuItem` components to construct the full menu. This component only takes one input:

- `items` list of string values which correspond to the menu items

The menu should be closed by default.

The Tests First

We'll be using the tests from the solution, but first let's think about a few things to test for each of `MenuButton`, `MenuItem` and `Menu`. Here's what the solutions have:

The Tests First

We'll be using the tests from the solution, but first let's think about a few things to test for each of `MenuButton`, `MenuItem` and `Menu`. Here's what the solutions have:

- `MenuButton`
  - triggers `onClick` event handler when clicked
  - `aria-label` attribute is defined
  - sets `aria-expanded` to false when closed
  - sets `aria-expanded` to true when open
- `MenuItem`
  - triggers `onClick` event handler with title when clicked
  - renders with custom title
- `Menu`
  - is closed by default
  - creates a `MenuItem` for every provided item

One Possible Directory Structure

Before we go and implement this, I want to show you a simple approach to folder structures.

One Possible Directory Structure

Before we go and implement this, I want to show you a simple approach to folder structures.

Essentially, you'll have something like:

```
- src/
  - index.js
  - index.css
  - components/
    - Button/
      - index.js
      - index.test.js
      - styles.module.css
    - ...
```

One Possible Directory Structure

Before we go and implement this, I want to show you a simple approach to folder structures.

Essentially, you'll have something like:

```
- src/
  - index.js
  - index.css
  - components/
    - Button/
      - index.js
      - index.test.js
      - styles.module.css
    - ...
```

Notice that you can import `Button` like
`import Button from '../path/to/components/Button'`.