Accordion Solution

Vanilla JS

Yangshun Tay

Ex-Meta Staff Engineer

Easy 15mins 6.5k done

Solution

Config / API Design

Part of the complexity of building a component is designing the API for it. The `accordion` function accepts a root element, `$rootEl` , where the accordion sections will be inserted. The second parameter of the `accordion` function is an object used to store the configuration options. At the bare minimum, we will need the following options:

- `items` : A list of item objects. Each item is an object with the fields:
  - `value` : A unique identifier for the accordion item.
  - `title` : The text label to show in the accordion title.
  - `contents` : The contents to show when the section is expanded.

Unlike our typical Vanilla JS approach, we opt not to keep any state in JavaScript this time round and instead rely on the DOM to track each accordion section's state and expand/collapse the appropriate elements based on that accordion's state.

The `accordion` function calls the `init` and `attachEvents` functions to set up the accordion component by rendering the DOM elements and attaching the necessary event listeners.

`init`

This function sets up the DOM elements that remain throughout the lifecycle of the component, aka they will never be destroyed. The accordion's sections (titles and contents) are rendered.

`attachEvents`

Only a single event listener is necessary in this component, which is the `click` event, to be added to the root element. We make use of Event Delegation so that only a single event listener has to be added and will work for all of its child contents. However, we have to be careful about checking which element is being clicked on and make sure we're only responding when the title is clicked and not the contents.

When a click is triggered on the accordion title, we'll need to rotate the icon and toggle the `hidden` attribute on the accordion contents.

Test cases

- All the provided sections should be displayed.
- Clicking on a collapsed section's title should expand it.
- Clicking on an expanded section's title should collapse it.
- Test that all sections are allowed to expand and collapse independently.
- Test that you are able to initialize multiple instances of the component, each with independent states.

Accessibility

Interactive elements need to be focusable, so we'll use a `<button>` for the title.

The **ARIA Authoring Practices Guide for Accordion** has a long list of guidelines for the ARIA roles, states, and properties to add to the various elements of an accordion. **Accordion II** and **Accordion III** will focus on

- src
  - index.js
  - styles.css
- index.html
- package.json

index.html   index.js   styles.css

```
import './styles.css';

(() => {
  function accordion($rootEl, { sections }) {
    function attachEvents() {
      // Use Event Delegation.
      $rootEl.addEventListener('click', (event) => {
        const target = event.target;
        if (
          target.tagName !== 'BUTTON' ||
          !target.classList.contains('accordion-item-title
        ) {
          return;
        }

        // Find the icon and toggle the direction.
        const $icon = target.querySelector(
          '.accordion-icon',
        );
        $icon.classList.toggle('accordion-icon--rotated');

        // Find the accordion contents and toggle the
        // contents' visibility.
        const $accordionContents = target.nextSibling;
        $accordionContents.hidden =
          !$accordionContents.hidden;
      });
    }

    function init() {
      $rootEl.classList.add('accordion');

      const $accordionSections =
        document.createDocumentFragment();

      sections.forEach(({ value, title, contents }) => {
        const $accordionSection =
          document.createElement('div');
        $accordionSection.classList.add('accordion-item');

        const $accordionTitleBtn =
          document.createElement('button');
        $accordionTitleBtn.classList.add(
          'accordion-item-title',
        );
        $accordionTitleBtn.type = 'button';
        $accordionTitleBtn.setAttribute(
          'data-value',
          value,
        );

        const $accordionIcon =
          document.createElement('span');
        $accordionIcon.classList.add('accordion-icon');
        $accordionIcon.setAttribute('aria-hidden', 'true')

        $accordionTitleBtn.append(title, $accordionIcon);

        const $accordionSectionContents =
          document.createElement('div');
        $accordionSectionContents.classList.add(
          'accordion-item-contents',
        );
        $accordionSectionContents.hidden = true;
        $accordionSectionContents.textContent = contents;

        $accordionSection.append(
          $accordionTitleBtn,
          $accordionSectionContents,
        );
        $accordionSections.append($accordionSection);
      });
```