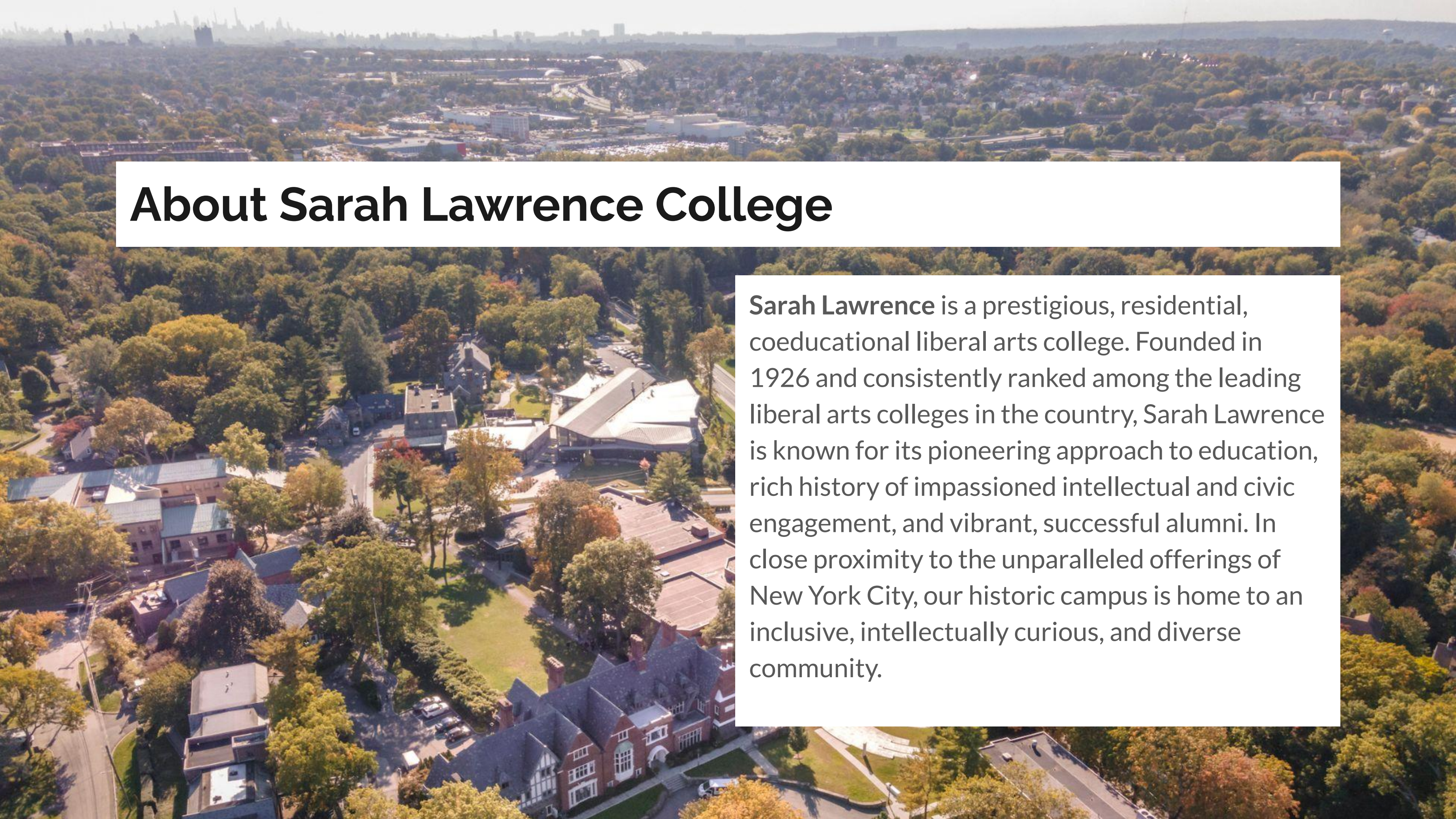
Accessible Accordions

Using data definitions, Velocity, and a little JS to get there



About Sarah Lawrence College

Sarah Lawrence is a prestigious, residential, coeducational liberal arts college. Founded in 1926 and consistently ranked among the leading liberal arts colleges in the country, Sarah Lawrence is known for its pioneering approach to education, rich history of impassioned intellectual and civic engagement, and vibrant, successful alumni. In close proximity to the unparalleled offerings of New York City, our historic campus is home to an inclusive, intellectually curious, and diverse community.

About your presenter(s)

- Winston Churchill-Joell, Director of Digital Services at Sarah Lawrence College since 2001
- Cascade Server for nearly 15 years
- Responsible for all things Cascade
- Passionate about digital accessibility, semantic markup, CSS, vanilla JS, and dogs
- Leo is the cute one with the slightly bigger ears; he’s passionate about anything that squeaks

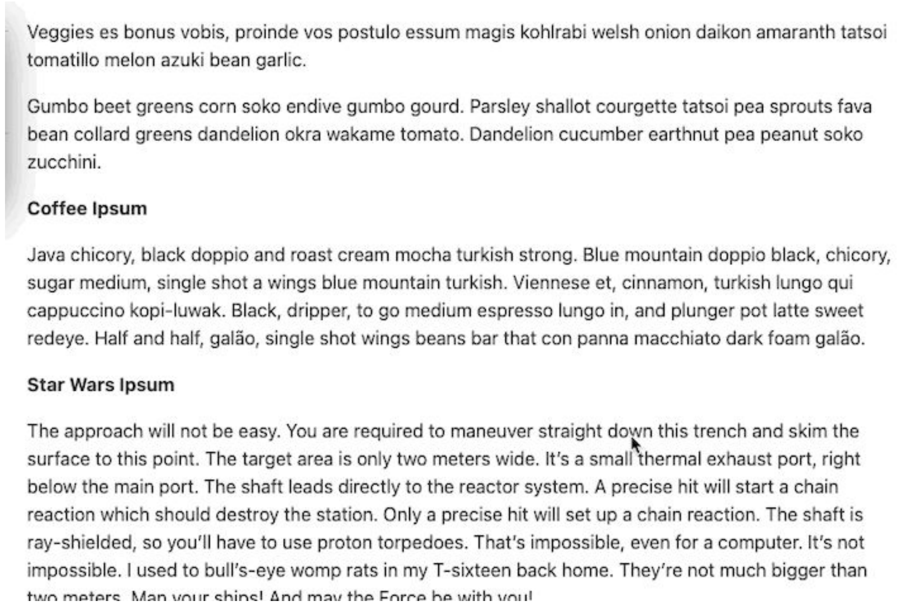
Project Files and Resources

- Github repo with all HTML, CSS, and JS files
- Download Cascade Server site archive
- Reference materials for this talk
- Authors and content creators I follow

slc.edu/csuc2023

Collapsible Sections: What they are

- **Semantic headings** that control the visibility of their content sections
- **Content by default**: no JS? No problem!
- **Keyboard accessible**: controls can be reached by tab key and activated by space and enter keys
- **Screen reader accessible**: semantic content, landmarks, and controls reveal structure and announce interaction state
- **Visually accessible**: unambiguous controls clearly show focus and interaction state



Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

Gumbo beet greens corn soko endive gumbo gourd. Parsley shallot courgette tatsoi pea sprouts fava bean collard greens dandelion okra wakame tomato. Dandelion cucumber earthnut pea peanut soko zucchini.

Coffee Ipsum

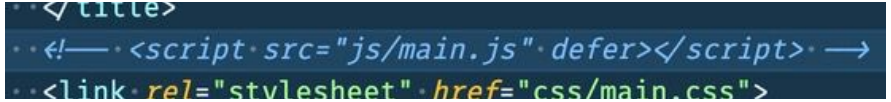
Java chicory, black doppio and roast cream mocha turkish strong. Blue mountain doppio black, chicory, sugar medium, single shot a wings blue mountain turkish. Viennese et, cinnamon, turkish lungo qui cappuccino kopi-luwak. Black, dripper, to go medium espresso lungo in, and plunger pot latte sweet redeye. Half and half, galão, single shot wings beans bar that con panna macchiato dark foam galão.

Star Wars Ipsum

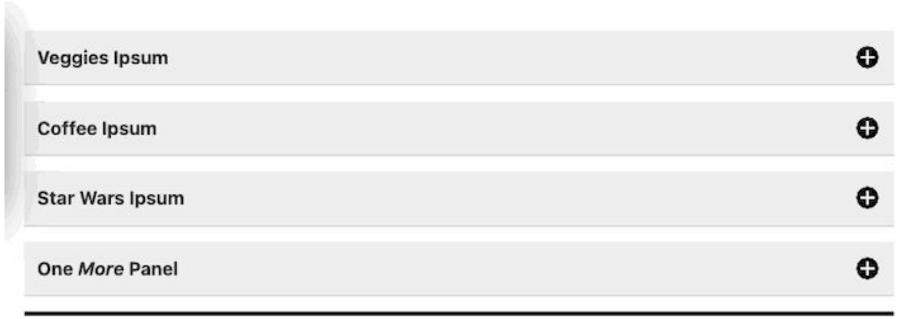
The approach will not be easy. You are required to maneuver straight down this trench and skim the surface to this point. The target area is only two meters wide. It's a small thermal exhaust port, right below the main port. The shaft leads directly to the reactor system. A precise hit will start a chain reaction which should destroy the station. Only a precise hit will set up a chain reaction. The shaft is ray-shielded, so you'll have to use proton torpedoes. That's impossible, even for a computer. It's not impossible. I used to bull's-eye womp rats in my T-sixteen back home. They're not much bigger than two meters. Man your ships! And may the Force be with you!

Collapsible Sections: What they are

- **Semantic headings** that control the visibility of their content sections
- **Content by default**: no JS? No problem!
- **Keyboard accessible**: controls can be reached by tab key and activated by space and enter keys
- **Screen reader accessible**: semantic content, landmarks, and controls reveal structure and announce interaction state
- **Visually accessible**: unambiguous controls clearly show focus and interaction state

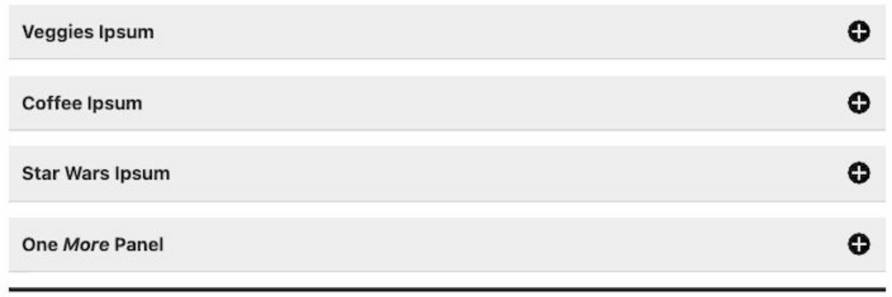


```
  </title>
  <!-- <script src="js/main.js" defer></script> -->
  <link rel="stylesheet" href="css/main.css">
```



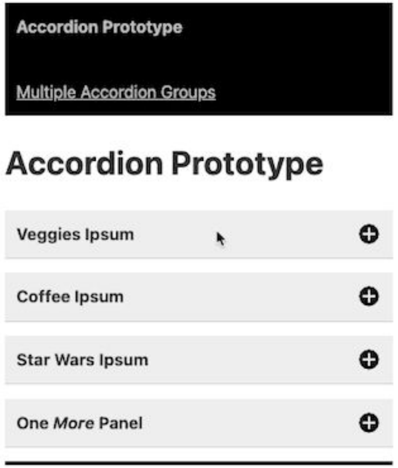
Collapsible Sections: What they are

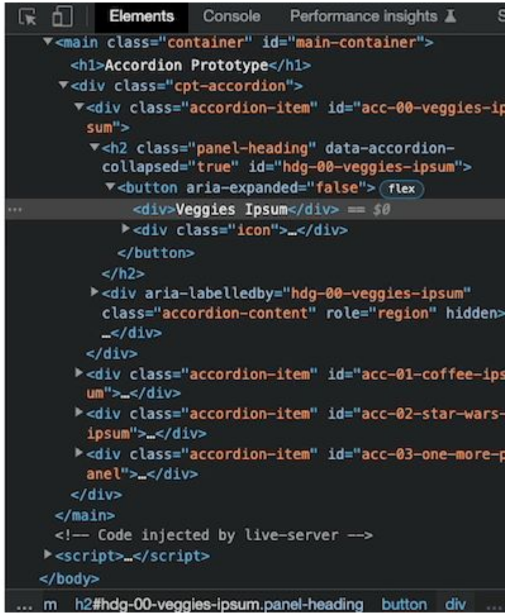
- **Semantic headings** that control the visibility of their content sections
- **Content by default**: no JS? No problem!
- **Keyboard accessible**: controls can be reached by tab key and activated by space and enter keys
- **Screen reader accessible**: semantic content, landmarks, and controls reveal structure and announce interaction state
- **Visually accessible**: unambiguous controls clearly show focus and interaction state



Collapsible Sections: What they are

- **Semantic headings** that control the visibility of their content sections
- **Content by default**: no JS? No problem!
- **Keyboard accessible**: controls can be reached by tab key and activated by space and enter keys
- **Screen reader accessible**: semantic content, landmarks, and controls reveal relationships and announce state
- **Visually accessible**: unambiguous controls clearly show focus and interaction state

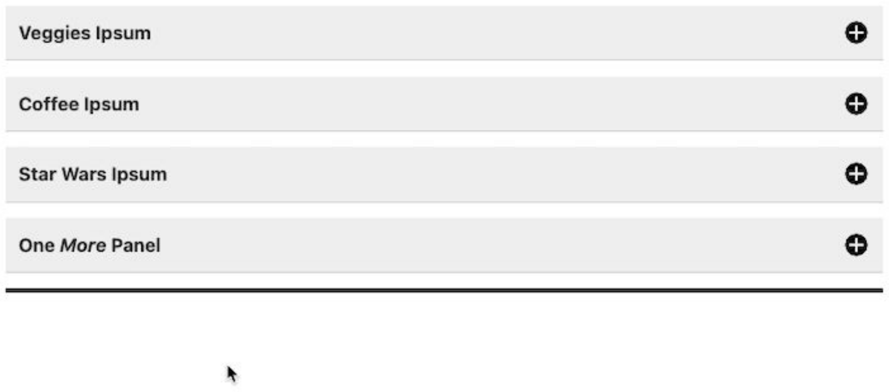




```
▼<main class="container" id="main-container">
   <h1>Accordion Prototype</h1>
  ▼<div class="cpt-accordion">
    ▼<div class="accordion-item" id="acc-00-veggies-ip
     sum">
      ▼<h2 class="panel-heading" data-accordion-
       collapsed="true" id="hdg-00-veggies-ipsum">
        ▼<button aria-expanded="false"> flex
...         <div>Veggies Ipsum</div> == $0
          ▶<div class="icon">…</div>
         </button>
       </h2>
      ▶<div aria-labelledby="hdg-00-veggies-ipsum"
       class="accordion-content" role="region" hidden>
       …</div>
     </div>
    ▶<div class="accordion-item" id="acc-01-coffee-ips
     um">…</div>
    ▶<div class="accordion-item" id="acc-02-star-wars-
     ipsum">…</div>
    ▶<div class="accordion-item" id="acc-03-one-more-p
     anel">…</div>
   </div>
 </main>
 <!-- Code injected by live-server -->
▶<script>…</script>
</body>
... m  h2#hdg-00-veggies-ipsum.panel-heading  button  div  ...
```

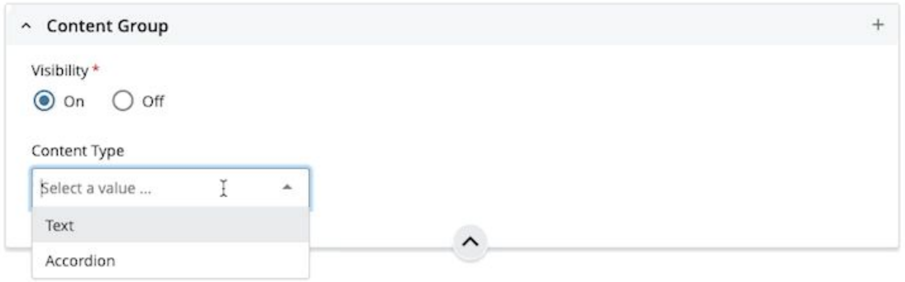
Collapsible Sections: What they are

- **Semantic headings** that control the visibility of their content sections
- **Content by default**: no JS? No problem!
- **Keyboard accessible**: controls can be reached by tab key and activated by space and enter keys
- **Screen reader accessible**: semantic content, landmarks, and controls reveal structure and announce interaction state
- **Visually accessible**: unambiguous controls clearly show focus and interaction state



Collapsible Sections: Data Definition

- **Consistency**: structured data to provide an easy, repeatable way to create content
- **Heading hierarchy**: settings to easily manage heading levels for accessibility
- **Rich formatted headings**: a streamlined inline-only WYSIWYG
- **Keeping it simple with schema**: limit the options for panel content and avoid inaccessible content scenarios
- **Use shared fields** for portability



Content Group

Visibility *

[✓ On] Off

Content Type

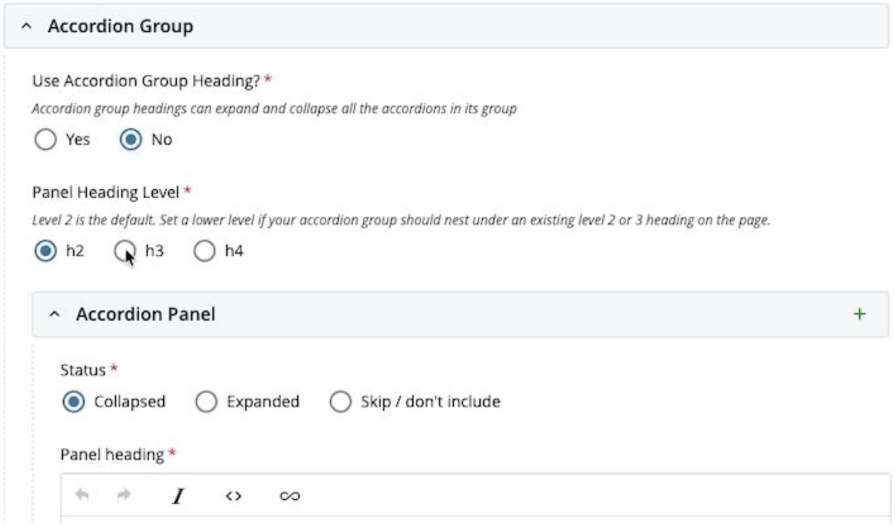
Select a value ...

Text

Accordion

Collapsible Sections: Data Definition

- **Consistency**: structured data to provide an easy, repeatable way to create content
- **Heading hierarchy**: settings to easily manage heading levels for accessibility
- **Rich formatted headings**: a streamlined inline-only WYSIWYG
- **Keeping it simple with schema**: limit the options for panel content and avoid inaccessible content scenarios
- **Use shared fields** for portability



Accordion Group

Use Accordion Group Heading? *

Accordion group headings can expand and collapse all the accordions in its group

Yes [✓ No]

Panel Heading Level *

Level 2 is the default. Set a lower level if your accordion group should nest under an existing level 2 or 3 heading on the page.

[✓ h2] h3 h4

Accordion Panel

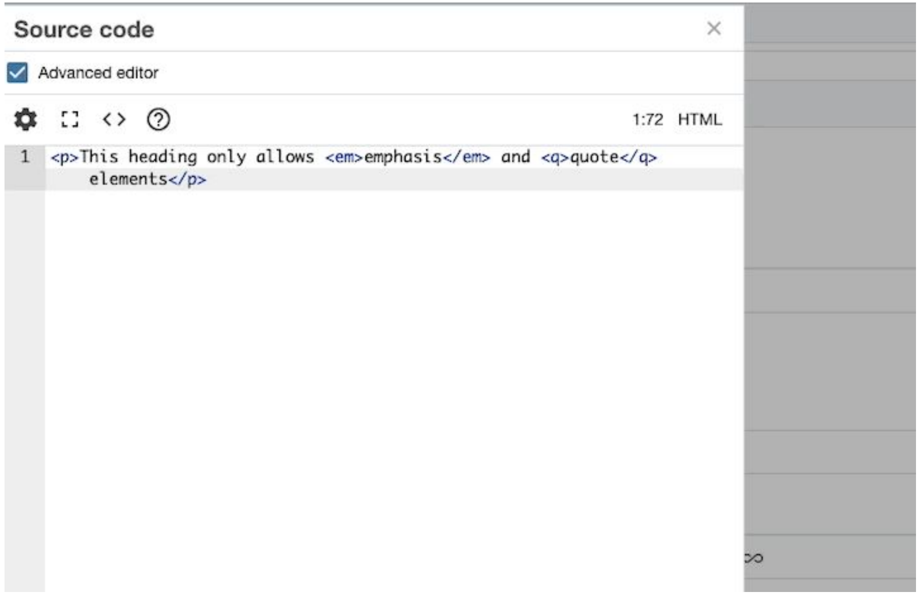
Status *

[✓ Collapsed] Expanded Skip / don't include

Panel heading *

Collapsible Sections: Data Definition

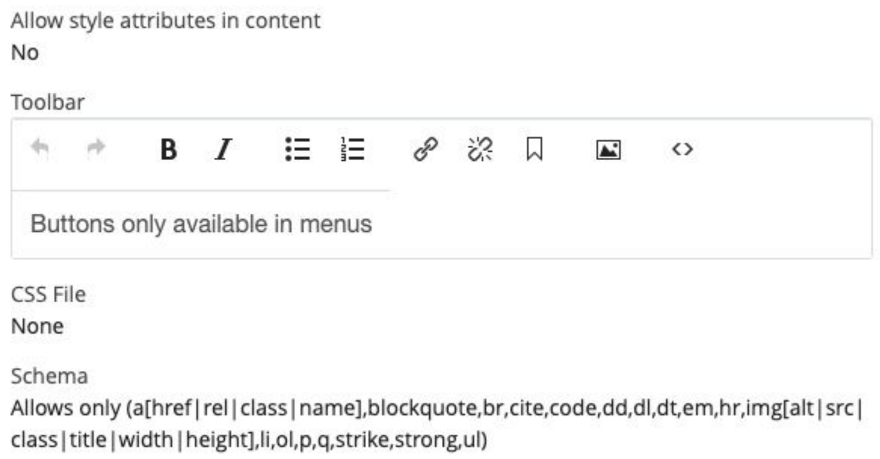
- **Consistency**: structured data to provide an easy, repeatable way to create content
- **Heading hierarchy**: settings to easily manage heading levels for accessibility
- **Rich formatted headings**: a streamlined inline-only WYSIWYG
- **Keeping it simple with schema**: limit the options for panel content and avoid inaccessible content scenarios
- **Use shared fields** for portability



```html
<p>This heading only allows <em>emphasis</em> and <q>quote</q>
    elements</p>
```

Collapsible Sections: Data Definition

- **Consistency**: structured data to provide an easy, repeatable way to create content
- **Heading hierarchy**: settings to easily manage heading levels for accessibility
- **Rich formatted headings**: a streamlined inline-only WYSIWYG
- **Keeping it simple with schema**: limit the options for panel content and avoid inaccessible content scenarios
- **Use shared fields** for portability



Allow style attributes in content
No

Toolbar

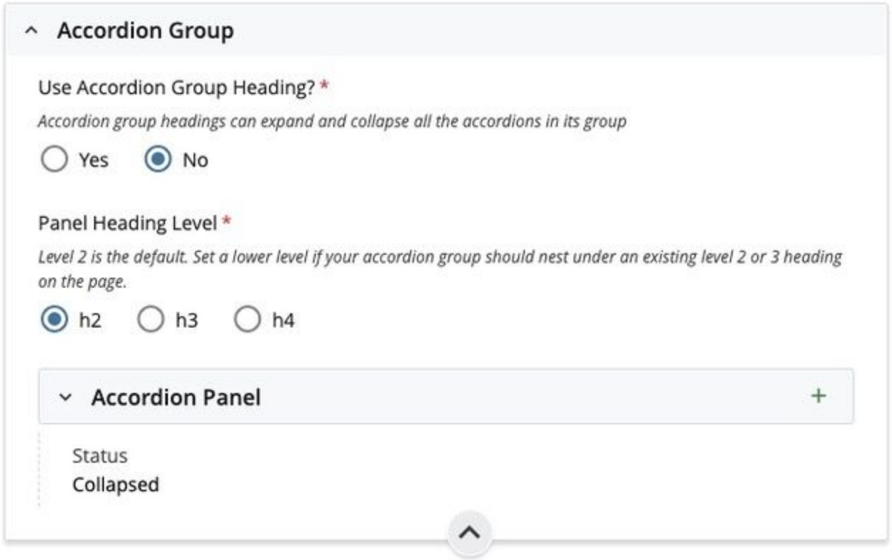
Buttons only available in menus

CSS File
None

Schema
Allows only (a[href|rel|class|name],blockquote,br,cite,code,dd,dl,dt,em,hr,img[alt|src|class|title|width|height],li,ol,p,q,strike,strong,ul)

Collapsible Sections: Data Definition

- **Consistency**: structured data to provide an easy, repeatable way to create content
- **Heading hierarchy**: settings to easily manage heading levels for accessibility
- **Rich formatted headings**: a streamlined inline-only WYSIWYG
- **Keeping it simple with schema**: limit the options for panel content and avoid inaccessible content scenarios
- **Use shared fields** for portability



Accordion Group

Use Accordion Group Heading? *

Accordion group headings can expand and collapse all the accordions in its group

Yes [✓ No]

Panel Heading Level *

Level 2 is the default. Set a lower level if your accordion group should nest under an existing level 2 or 3 heading on the page.

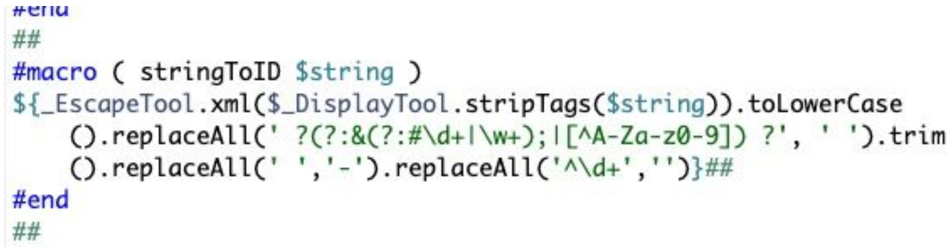
[✓ h2] h3 h4

Accordion Panel

Status
Collapsed

Collapsible Sections: Velocity

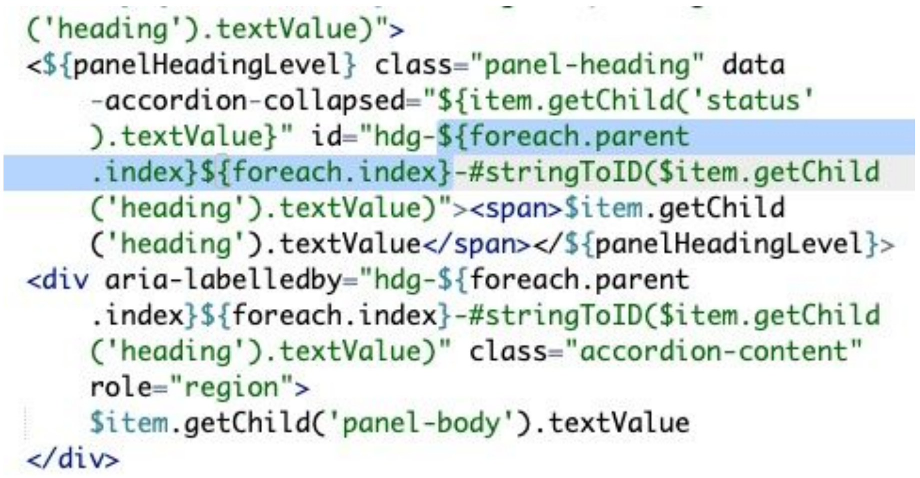
- **String methods** to make easy work of generating the required ARIA attributes and element IDs to set up accordions
- Using **`$foreach.index`** and `$foreach.parent` to ensure unique IDs
- **Helper macros** as utilities to handle complex string operations
- **Render semantic output** and nothing more; use JS to add interactive elements and icons



```
#end
##
#macro ( stringToID $string )
${_EscapeTool.xml($_DisplayTool.stripTags($string)).toLowerCase
    ().replaceAll(' ?(?:&(?:#\d+|\w+);|[^A-Za-z0-9]) ?', ' ').trim
    ().replaceAll(' ','-').replaceAll('^\d+','')}##
#end
##
```

Collapsible Sections: Velocity

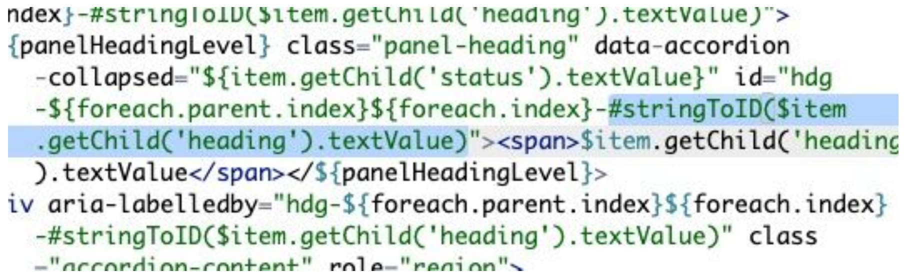
- **String methods** to make easy work of generating the required ARIA attributes and element IDs to set up accordions
- Using `$foreach.index` and `$foreach.parent` to ensure unique IDs
- **Helper macros** as utilities to handle complex string operations
- **Render semantic output** and nothing more; use JS to add interactive elements and icons



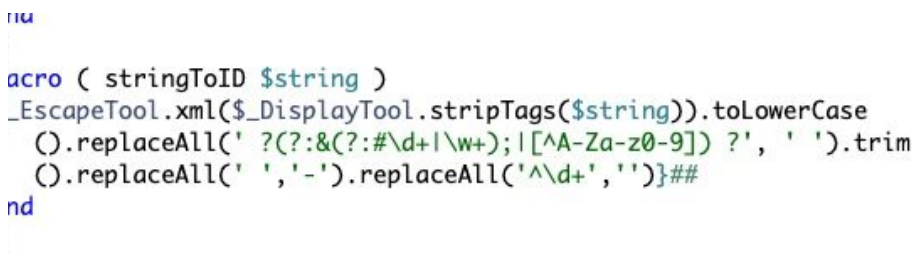
```
('heading').textValue)">
<${panelHeadingLevel} class="panel-heading" data
    -accordion-collapsed="${item.getChild('status'
    ).textValue}" id="hdg-${foreach.parent
    .index}${foreach.index}-#stringToID($item.getChild
    ('heading').textValue)"><span>$item.getChild
    ('heading').textValue</span></${panelHeadingLevel}>
<div aria-labelledby="hdg-${foreach.parent
    .index}${foreach.index}-#stringToID($item.getChild
    ('heading').textValue)" class="accordion-content"
    role="region">
    $item.getChild('panel-body').textValue
</div>
```

Collapsible Sections: Velocity

- **String methods** to make easy work of generating the required ARIA attributes and element IDs to set up accordions
- Using **`$foreach.index`** and `$foreach.parent` to ensure unique IDs
- **Helper macros** as utilities to handle complex string operations
- **Render semantic output** and nothing more; use JS to add interactive elements and icons



```
ndex}-#stringToID($item.getChild('heading').textValue)">
{panelHeadingLevel} class="panel-heading" data-accordion
  -collapsed="${item.getChild('status').textValue}" id="hdg
  -${foreach.parent.index}${foreach.index}-#stringToID($item
  .getChild('heading').textValue)"><span>$item.getChild('heading
  ).textValue</span></${panelHeadingLevel}>
iv aria-labelledby="hdg-${foreach.parent.index}${foreach.index}
  -#stringToID($item.getChild('heading').textValue)" class
  ="accordion-content" role="region">
```

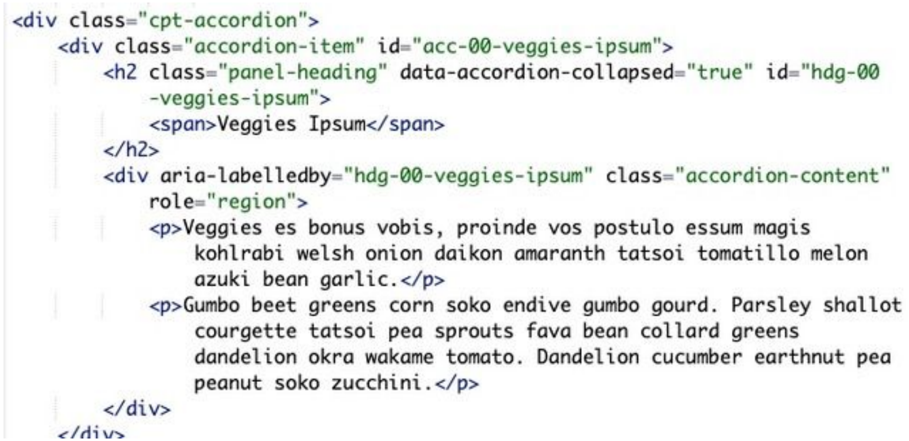


```
nd

acro ( stringToID $string )
_EscapeTool.xml($_DisplayTool.stripTags($string)).toLowerCase
  ().replaceAll(' ?(?:&(?:#\d+|\w+);|[^A-Za-z0-9]) ?', ' ').trim
  ().replaceAll(' ','-').replaceAll('^\d+','')}##
nd
```

Collapsible Sections: Velocity

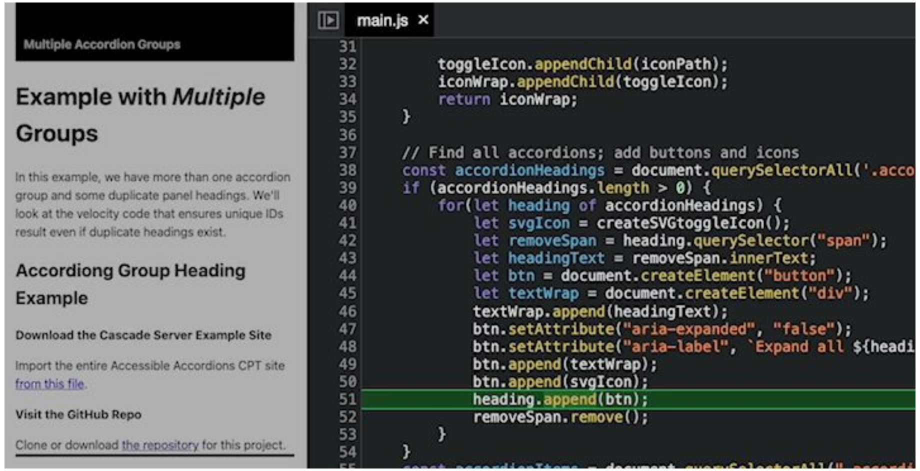
- **String methods** to make easy work of generating the required ARIA attributes and element IDs to set up accordions
- Using **`$foreach.index`** and `$foreach.parent` to ensure unique IDs
- **Helper macros** as utilities to handle complex string operations
- **Render semantic output** and nothing more; use JS to add interactive elements and icons



```
<div class="cpt-accordion">
    <div class="accordion-item" id="acc-00-veggies-ipsum">
        <h2 class="panel-heading" data-accordion-collapsed="true" id="hdg-00
            -veggies-ipsum">
            <span>Veggies Ipsum</span>
        </h2>
        <div aria-labelledby="hdg-00-veggies-ipsum" class="accordion-content"
            role="region">
            <p>Veggies es bonus vobis, proinde vos postulo essum magis
                kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon
                azuki bean garlic.</p>
            <p>Gumbo beet greens corn soko endive gumbo gourd. Parsley shallot
                courgette tatsoi pea sprouts fava bean collard greens
                dandelion okra wakame tomato. Dandelion cucumber earthnut pea
                peanut soko zucchini.</p>
        </div>
    </div>
```

Collapsible Sections: Progressive Enhancement

- **Add semantic controls** with JavaScript
- **Use buttons**, not DIVs; they ship with tab focus and keyboard interactivity for free
- **Use properties and attributes** instead of classes to keep track of state
- **Use the history API** to create URLs for deep linking to sections
- **Add support for search** on page (cmd/ctrl F)
- **Use event delegation** for performance
- **Execute first** in the call stack as soon as the DOM is *interactive* (`defer` or `readyState === 'interactive'`)



Example with *Multiple* Groups

In this example, we have more than one accordion group and some duplicate panel headings. We'll look at the velocity code that ensures unique IDs result even if duplicate headings exist.

Accordiong Group Heading Example

Download the Cascade Server Example Site

Import the entire Accessible Accordions CPT site from this file.

Visit the GitHub Repo

Clone or download the repository for this project.

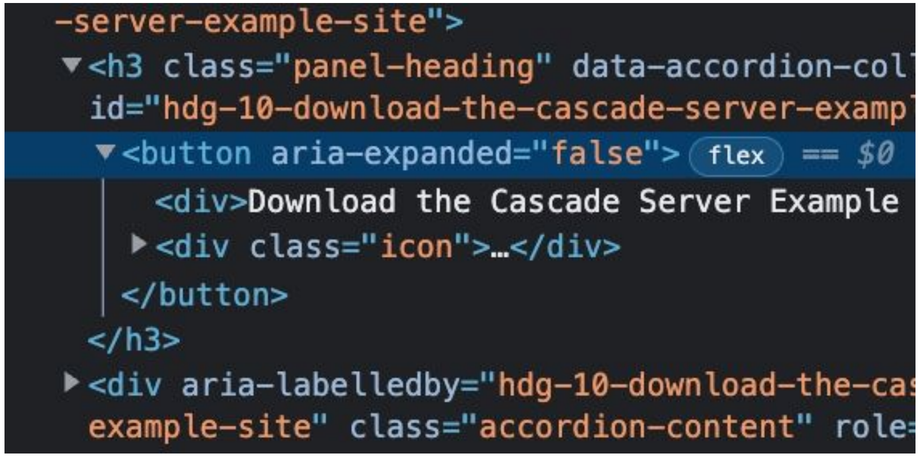
```

        toggleIcon.appendChild(iconPath);
        iconWrap.appendChild(toggleIcon);
        return iconWrap;
    }

    // Find all accordions; add buttons and icons
    const accordionHeadings = document.querySelectorAll('.acco
    if (accordionHeadings.length > 0) {
        for(let heading of accordionHeadings) {
            let svgIcon = createSVGtoggleIcon();
            let removeSpan = heading.querySelector("span");
            let headingText = removeSpan.innerText;
            let btn = document.createElement("button");
            let textWrap = document.createElement("div");
            textWrap.append(headingText);
            btn.setAttribute("aria-expanded", "false");
            btn.setAttribute("aria-label", `Expand all ${headi
            btn.append(textWrap);
            btn.append(svgIcon);
            heading.append(btn);
            removeSpan.remove();
        }
    }
```

Collapsible Sections: Progressive Enhancement

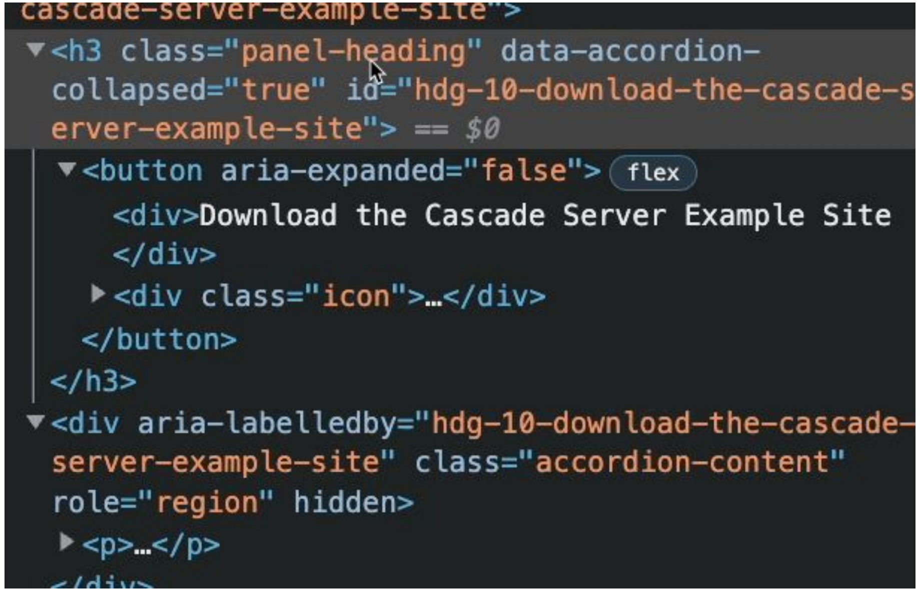
- **Add semantic controls** with JavaScript
- **Use buttons**, not DIVs; they ship with tab focus and keyboard interactivity for free
- **Use properties and attributes** instead of classes to keep track of state
- **Use the history API** to create URLs for deep linking to sections
- **Add support for search** on page (cmd/ctrl F)
- **Use event delegation** for performance
- **Execute first** in the call stack as soon as the DOM is *interactive* (`defer` or `readyState === 'interactive'`)



```
-server-example-site">
 ▼<h3 class="panel-heading" data-accordion-col
  id="hdg-10-download-the-cascade-server-examp
   ▼<button aria-expanded="false"> flex == $0
       <div>Download the Cascade Server Example
      ▶<div class="icon">…</div>
    </button>
  </h3>
 ▶<div aria-labelledby="hdg-10-download-the-cas
  example-site" class="accordion-content" role=
```

Collapsible Sections: Progressive Enhancement

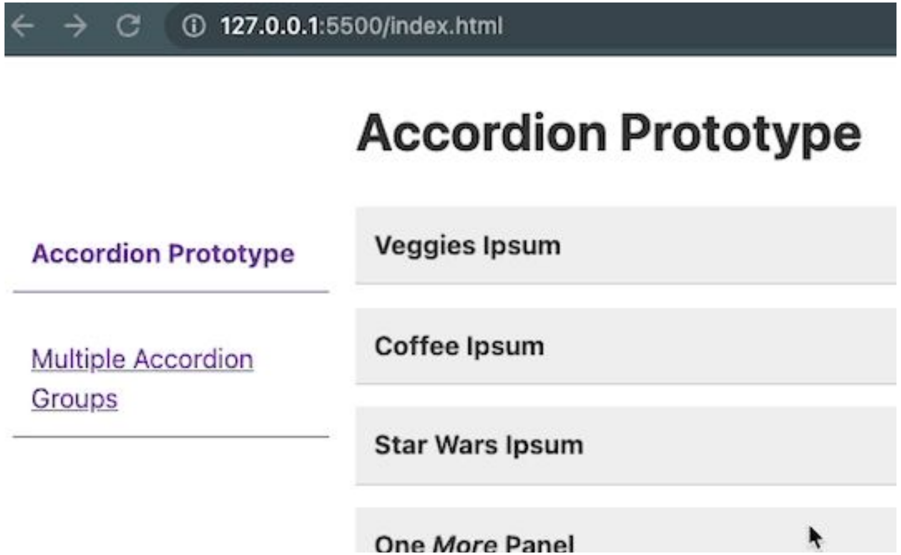
- **Add semantic controls** with JavaScript
- **Use buttons**, not DIVs; they ship with tab focus and keyboard interactivity for free
- **Use properties and attributes** instead of classes to keep track of state
- **Use the history API** to create URLs for deep linking to sections
- **Add support for search** on page (cmd/ctrl F)
- **Use event delegation** for performance
- **Execute first** in the call stack as soon as the DOM is *interactive* (`defer` or `readyState === 'interactive'`)



```
cascade-server-example-site">
▼<h3 class="panel-heading" data-accordion-
collapsed="true" id="hdg-10-download-the-cascade-s
erver-example-site"> == $0
  ▼<button aria-expanded="false"> flex
      <div>Download the Cascade Server Example Site
      </div>
    ▶<div class="icon">…</div>
    </button>
  </h3>
▼<div aria-labelledby="hdg-10-download-the-cascade-
server-example-site" class="accordion-content"
role="region" hidden>
  ▶<p>…</p>
  </div>
```

Collapsible Sections: Progressive Enhancement

- **Add semantic controls** with JavaScript
- **Use buttons**, not DIVs; they ship with tab focus and keyboard interactivity for free
- **Use properties and attributes** instead of classes to keep track of state
- **Use the history API** to create URLs for deep linking to sections
- **Add support for search** on page (cmd/ctrl F)
- **Use event delegation** for performance
- **Execute first** in the call stack as soon as the DOM is *interactive* (`defer` or `readyState === 'interactive'`)


127.0.0.1:5500/index.html

Accordion Prototype

Accordion Prototype

Multiple Accordion Groups

Veggies Ipsum

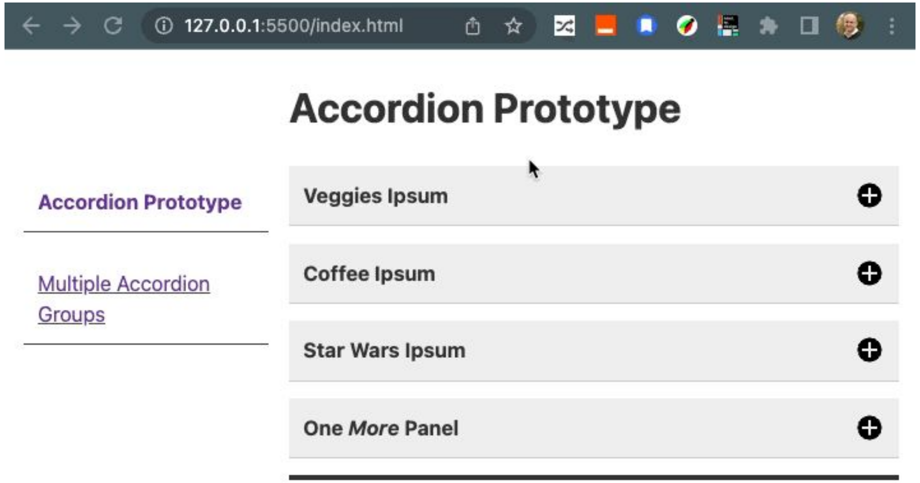
Coffee Ipsum

Star Wars Ipsum

One More Panel

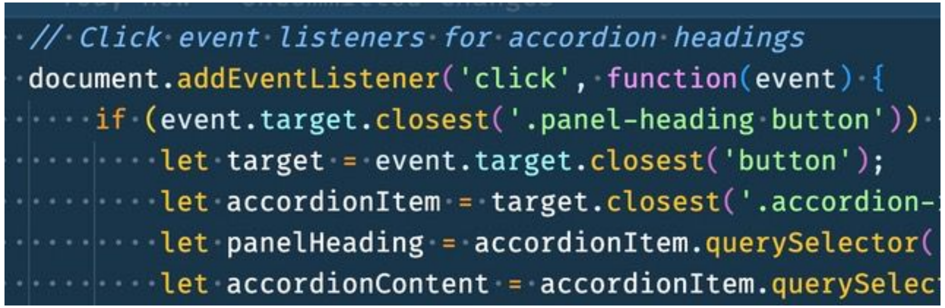
Collapsible Sections: Progressive Enhancement

- **Add semantic controls** with JavaScript
- **Use buttons**, not DIVs; they ship with tab focus and keyboard interactivity for free
- **Use properties and attributes** instead of classes to keep track of state
- **Use the history API** to create URLs for deep linking to sections
- **Add support for search** on page (cmd/ctrl F)
- **Use event delegation** for performance
- **Execute first** in the call stack as soon as the DOM is *interactive* (`defer` or `readyState === 'interactive'`)



Collapsible Sections: Progressive Enhancement

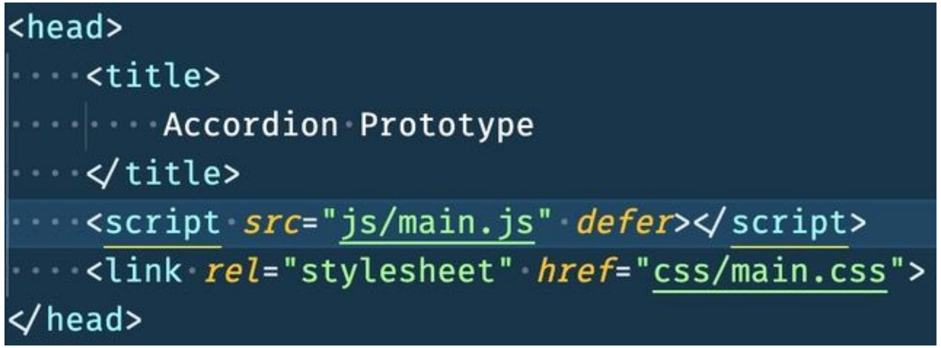
- **Add semantic controls** with JavaScript
- **Use buttons**, not DIVs; they ship with tab focus and keyboard interactivity for free
- **Use properties and attributes** instead of classes to keep track of state
- **Use the history API** to create URLs for deep linking to sections
- **Add support for search** on page (cmd/ctrl F)
- **Use event delegation** for performance
- **Execute first** in the call stack as soon as the DOM is *interactive* (`defer` or `readyState === 'interactive'`)



```
// Click event listeners for accordion headings
document.addEventListener('click', function(event) {
    if (event.target.closest('.panel-heading button'))
        let target = event.target.closest('button');
        let accordionItem = target.closest('.accordion-
        let panelHeading = accordionItem.querySelector(
        let accordionContent = accordionItem.querySelec
```

Collapsible Sections: Progressive Enhancement

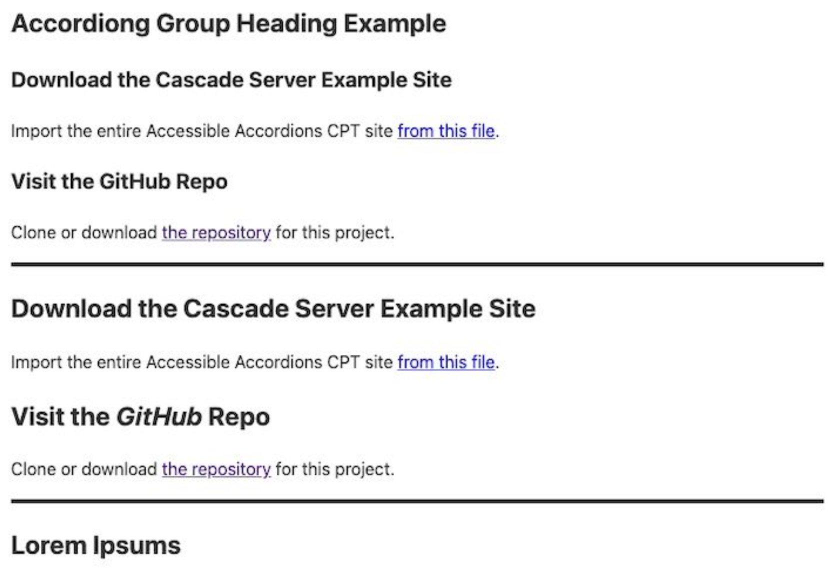
- **Add semantic controls** with JavaScript
- **Use buttons**, not DIVs; they ship with tab focus and keyboard interactivity for free
- **Use properties and attributes** instead of classes to keep track of state
- **Use the history API** to create URLs for deep linking to sections
- **Add support for search** on page (cmd/ctrl F)
- **Use event delegation** for performance
- **Execute first** in the call stack as soon as the DOM is *interactive* (`defer` or `readyState === 'interactive'`)



```
<head>
    <title>
        Accordion Prototype
    </title>
    <script src="js/main.js" defer></script>
    <link rel="stylesheet" href="css/main.css">
</head>
```

Collapsible Sections: Minimal CSS

- **Preserve intuitive heading appearance** when JS is not present
- **Normalize accordion and panel heading** appearance regardless of level
- **Add print support** with media queries
- **Focus rings** can be *beautiful* and they are *essential*
  - **Bonus**: let the user choose the color
- Use **`:hover`** and **`background-color`** for clear visual cues
- Use **`:target`** and **`scroll-behavior: smooth`** (responsibly) for deep links


Accordiong Group Heading Example

Download the Cascade Server Example Site

Import the entire Accessible Accordions CPT site from this file.

Visit the GitHub Repo

Clone or download the repository for this project.

Download the Cascade Server Example Site

Import the entire Accessible Accordions CPT site from this file.

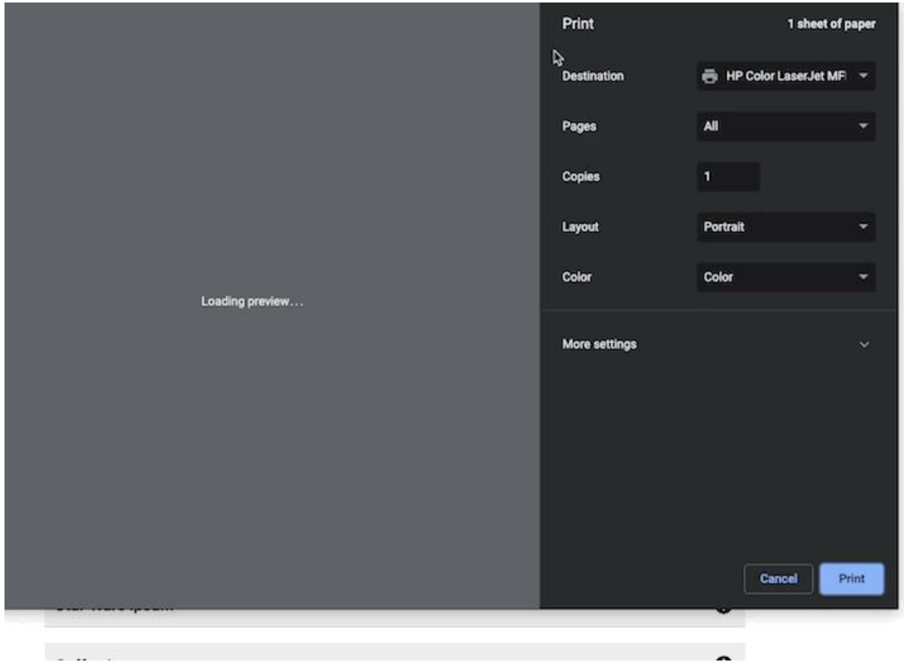
Visit the *GitHub* Repo

Clone or download the repository for this project.

Lorem Ipsums

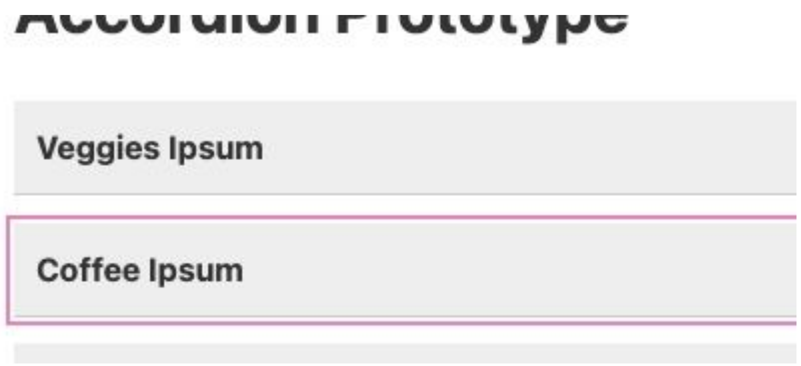
Collapsible Sections: Minimal CSS

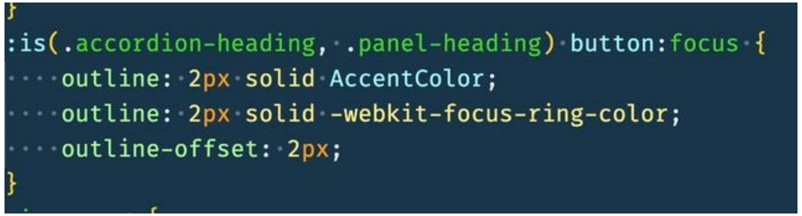
- **Preserve intuitive heading appearance** when JS is not present
- **Normalize accordion and panel heading** appearance regardless of level
- **Add print support** with media queries
- **Focus rings** can be *beautiful* and they are *essential*
  - **Bonus**: let the user choose the color
- Use **`:hover`** and **`background-color`** for clear visual cues
- Use **`:target`** and **`scroll-behavior: smooth`** (responsibly) for deep links


Print
1 sheet of paper
Destination: HP Color LaserJet MF
Pages: All
Copies: 1
Layout: Portrait
Color: Color
More settings
Loading preview…

Collapsible Sections: Minimal CSS

- **Preserve intuitive heading appearance** when JS is not present
- **Normalize accordion and panel heading** appearance regardless of level
- **Add print support** with media queries
- **Focus rings** can be *beautiful* and they are *essential*
    - **Bonus:** let the user choose the color
- **Use `:hover` and `background-color`** for clear visual cues
- **Use `:target` and `scroll-behavior: smooth`** (responsibly) for deep links

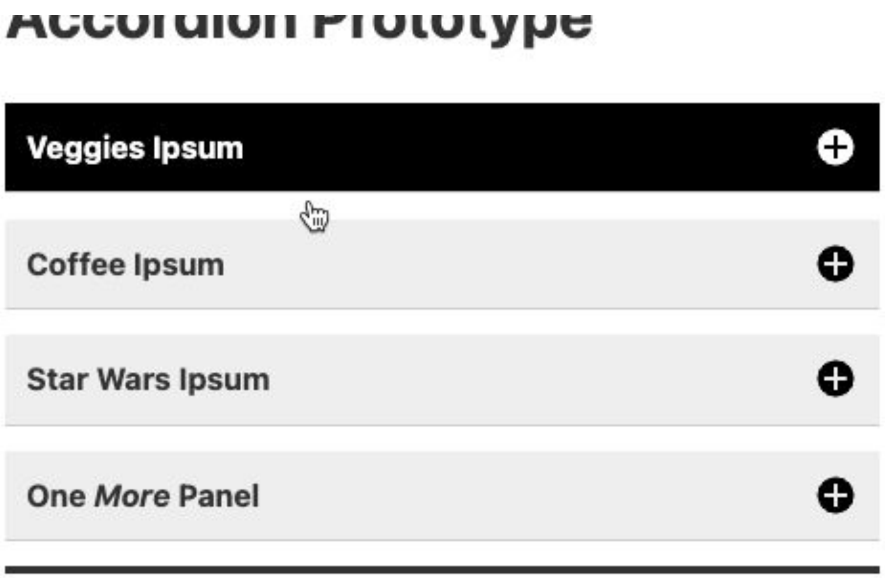




```
}
:is(.accordion-heading, .panel-heading) button:focus {
    outline: 2px solid AccentColor;
    outline: 2px solid -webkit-focus-ring-color;
    outline-offset: 2px;
}
```

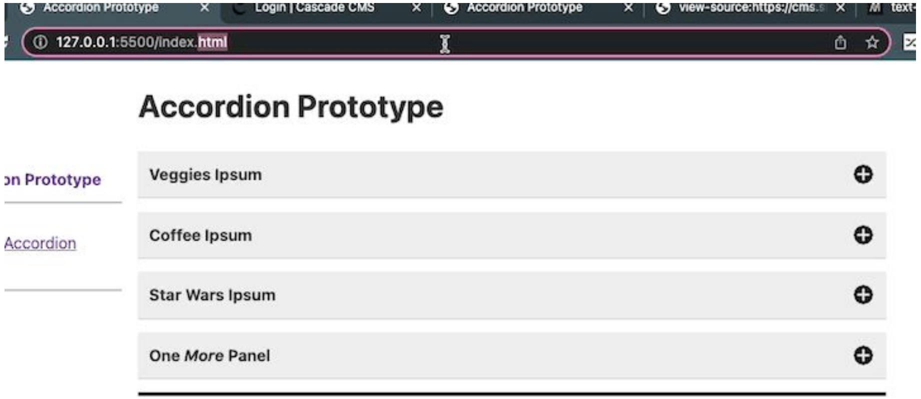
Collapsible Sections: Minimal CSS

- **Preserve intuitive heading appearance** when JS is not present
- **Normalize accordion and panel heading** appearance regardless of level
- **Add print support** with media queries
- **Focus rings** can be *beautiful* and they are *essential*
    - **Bonus:** let the user choose the color
- Use `:hover` and `background-color` for clear visual cues
- Use `:target` and `scroll-behavior: smooth` (responsibly) for deep links



Collapsible Sections: Minimal CSS

- **Preserve intuitive heading appearance** when JS is not present
- **Normalize accordion and panel heading** appearance regardless of level
- **Add print support** with media queries
- **Focus rings** can be *beautiful* and they are *essential*
  - **Bonus**: let the user choose the color
- **Use `:hover` and `background-color`** for clear visual cues
- **Use `:target` and `scroll-behavior: smooth`** (responsibly) for deep links



Accordion Prototype

Veggies Ipsum

Coffee Ipsum

Star Wars Ipsum

One *More* Panel

Demo time...

Thank you

And once again, project files and resources…

- Github repo with example HTML, CSS, and JS files
- Download Cascade Server site archive
- Reference materials for this talk
- Authors and content creators I follow

slc.edu/csuc2023