

attention to <details>

PRESENTED BY

Molly Engelberg

follow along with demos!

madebymolly.uk/seminar/demos.html

anatomy of the <details> element

html + basics

anatomy of the <details> element

DEMO 01

summary The “summary” is the visible part of the dropdown.

content Then add content that you want to appear when the dropdown is clicked.

Summary + content = details

Details is for interactive dropdowns. Common examples are FAQs, or pages with a lot of information such as a Wikipedia page.

Semantically, the details element should only be used to collapse information elegantly.

It should not be used to take the place of other types of control over the page that already exist with other elements, like a navigation menu or footnotes.

anatomy of the <details> element

DEMO 01

```
<details>  
  <summary>This is my summary!</summary>  
  <p>This is my secret hidden information!</p>  
</details>
```

▶ This is my <summary>!

>

▼ This is my <summary>!

This is my secret hidden information!

Use a <p> or a for your content.

Headings inside <details> should be avoided where possible, but if you need to use a heading element inside a summary semantically, use `display: inline` to avoid spacing issues.

open vs. closed

DEMO 02

By default, details elements load with only the summary visible.

By adding "open" to the element, we can tell the browser to load the element already opened.

```
<details open>
  <summary>This details element starts as open.</summary>
  <p>This information is NOT secret.</p>
</details>

<details>
  <summary>This details element starts as closed.</summary>
  <p>This information IS secret.</p>
</details>
```

- ▼ This `<details>` element starts as open.
This information is NOT secret.
- ▶ This `<details>` element starts as closed.

By adding "open" to our details element, we can tell the browser that when it loads, we want that one to start as open.

Otherwise, by default, they'll all appear closed.

A good example of this might be important information, like at the top of a digital form, where it may say something along the lines of:

"How to fill out this form:

- Go get your passport
- Find the tiny ID numbers on the main page
- etc"

and then you can collapse it to see the rest of the form more easily.

exclusivity

DEMO 03

As of 2024, we can also use HTML to tell the browsers to only have one dropdown open at a time.

When "name = exclusive" is added, when a dropdown is opened, any other open dropdowns will close.

```
<details name="exclusive" open>
  <summary>This details element starts as open.</summary>
  <p>This information is NOT secret.</p>
</details>

<details name="exclusive">
  <summary>This details element starts as closed.</summary>
  <p>This information IS secret.</p>
</details>
```

▼ This `<details>` element starts as open.
This information is NOT secret.
▶ This `<details>` element starts as closed.



▶ This `<details>` element starts as open.
▼ This `<details>` element starts as closed.
This information IS secret.

Exclusivity is more useful for something like an extensive FAQ - you want the open dropdown to close when another dropdown opens. That way content stays collapsed and organized so it's easy to navigate between them.

styling <details>

adding css

simple css

DEMO 04

```
details {  
  background-color: lightblue;  
}  
  
details[open] {  
  background-color: lightgreen;  
}
```

▶ This is my summary!

▼

▼ This is my summary!

This is my secret hidden information!

We can use standard CSS on this element just like you would with other elements, like adding a background color, a border, limiting the max-width, et cetera.

We can also add extra css to apply only when the element is opened. Here's an example where it's blue while closed and green while open.

selecting parts of the details element

In order to select different parts of your details element in CSS without using a class or id, try these pseudo-classes:

details :: summary

summary :: marker

summary :: after

details :: details-content (NEW!)

Use `summary::after` when:

Adding toggle arrows

Adding icons

Adding background images

Use `details::details-content` when:

Styling the expanded section

Animating open/close

Adding padding/background/borders to content container

`::details-content` became standard just a few months ago in September 2025, might still have issues with Safari

changing text

DEMO 05

```
<details>
  <summary>Click here to</summary>
  <p>This is my secret information!</p>
</details>
```

```
summary::after {
  content: " open";
}

details[open] summary::after {
  content: " close";
}
```

▶ Click here to open

▼

▼ Click here to close

This is my secret information!

Here's how we'd use `::after` to make it say "click to open" when it's closed and "click to close" when it's open.

changing the marker

DEMO 06

```
summary::marker {  
  content: '-';  
}
```

note: safari doesn't support list-style-type and list-style-position

target safari markers with `:is(::-webkit-details-marker)`

Here's how we'd use `::marker` to customize the little arrow on the left.

Summary also can be manipulated like a list, by changing `list-style-type` and `list-style-position` to adjust the marker, but it's not supported across every browser yet.

To make sure it's compliant on Safari, add the webkit piece onto your CSS rule.

animating

no built-in transition support

```
details[open] summary ~ * {  
  animation: open 0.3s ease-in-out;  
}  
  
@keyframes open {  
  from { opacity: 0; transform: translateY(-10px); }  
  to   { opacity: 1; transform: translateY(0); }  
}
```

Animating `<details>` right now is tricky because there's no built-in consistent support across browsers to just say "transition: xyz" in CSS. Edge and Chrome-based browsers already support simply adding transition and animate line items to CSS for `<details>`, but Firefox and Safari have yet to implement. It still is worth including these in your CSS, though, because even though your content will still statically snap open and shut on Firefox and Safari right now, when the browser support is added, it will immediately start working.

Alternatively, an example of a workaround here is animating the content **INSIDE** the details element instead. This is some CSS that's saying when details gets the open attribute, every sibling after the summary animates in.

one more thing!

this is zach



remember him?

Zach Leatherman spoke at State of the Browser on Saturday. He has also created a very convenient open source tool for `<details>` that is available on his github.

bonus responsive control



He created `<details-utils>`

github: [zachleat](#)

Blog: zachleat.com/web/details-utils/

If you download his work and wrap your `<details>` element with `<details-utils>`, you can:

- Click outside to close
- Bind an optional close button
- Close with 'Escape' key
- Animate open and close more elegantly
- Toggle between multiple classes
- Force open/closed based on media queries

This is worth exploring if you plan to use a details element in your major project, like if you have a FAQ or something

If you're trying to use someone's github work you can link it into your work in your HTML, similar to how you'd link other external files like a font library

Could be fun to play with if you're excited to use the details element regularly

have fun with it!

hover effects

custom markers

animated markers

ombre backgrounds

make it look like tabs of a file folder



recap

summary + content = details

Details start closed by default. Adding specifics like `open` or `name = "exclusive"` changes what is open and closed at what time.

Nearly every part of `<details>` is customizable with CSS.

Ways to structure `<details>` are being streamlined **now**. Good to stay current and ensure everything you're doing is universally supported.