

**attention to
<details>**

PRESENTED BY

Molly Engelberg

follow along with demos!

madebymollyy.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.

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!

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.

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.

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!

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!)

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!

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)`

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); }  
}
```

one more thing!

this is zach



remember him?

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

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.