CSS, JavaScript, & Accessibility

Ire Aderinokun @

An Event Apart Spring Summit 2022

“**By default, HTML is accessible.**

Web accessibility involves ensuring that content remains accessible.”

https://developer.mozilla.org/en-US/docs/Learn/Accessibility

Web accessibility is the **inclusive practice** of ensuring there are no barriers that prevent access to websites

Physical disabilities

Situational disabilities

Socio-economic restrictions

Accessibility is about inclusion

W3C® Web Accessibility Initiative WAI

WCAG

W3C Accessibility Guidelines

1. Content must be **perceivable** 🧏‍♀️

2. Interface must be **operable** 🛠️

3. Content must be **understandable** 📖

4. Code must be **robust** ⚙️

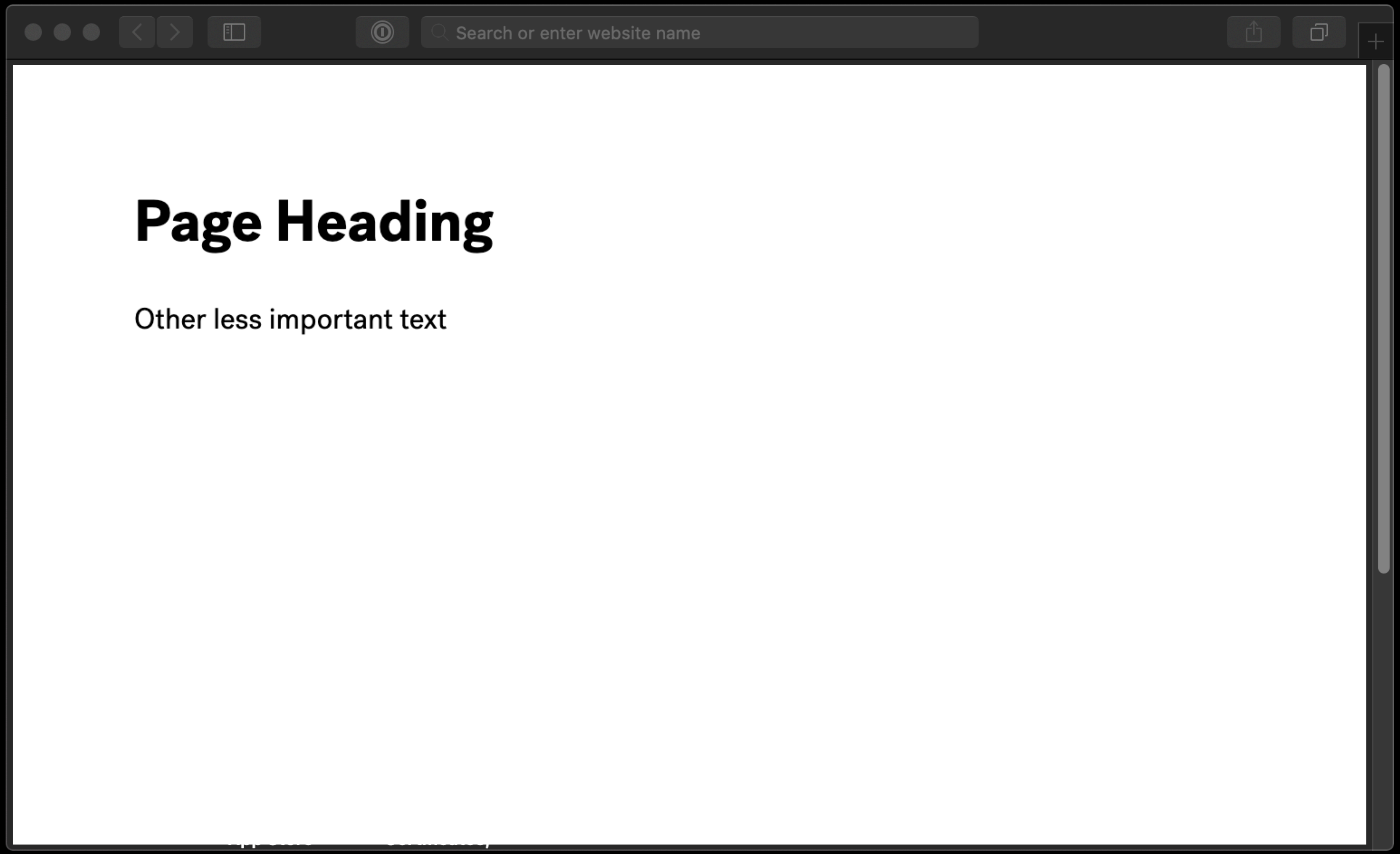
"By default, HTML is accessible, **if used correctly**.

Web accessibility involves ensuring that content remains accessible."

https://developer.mozilla.org/en-US/docs/Learn/Accessibility

With HTML, **WYSIWYG**

```
<main>

  <h1>Page Heading</h1>

  <p>Other less important text</p>

</main>
```

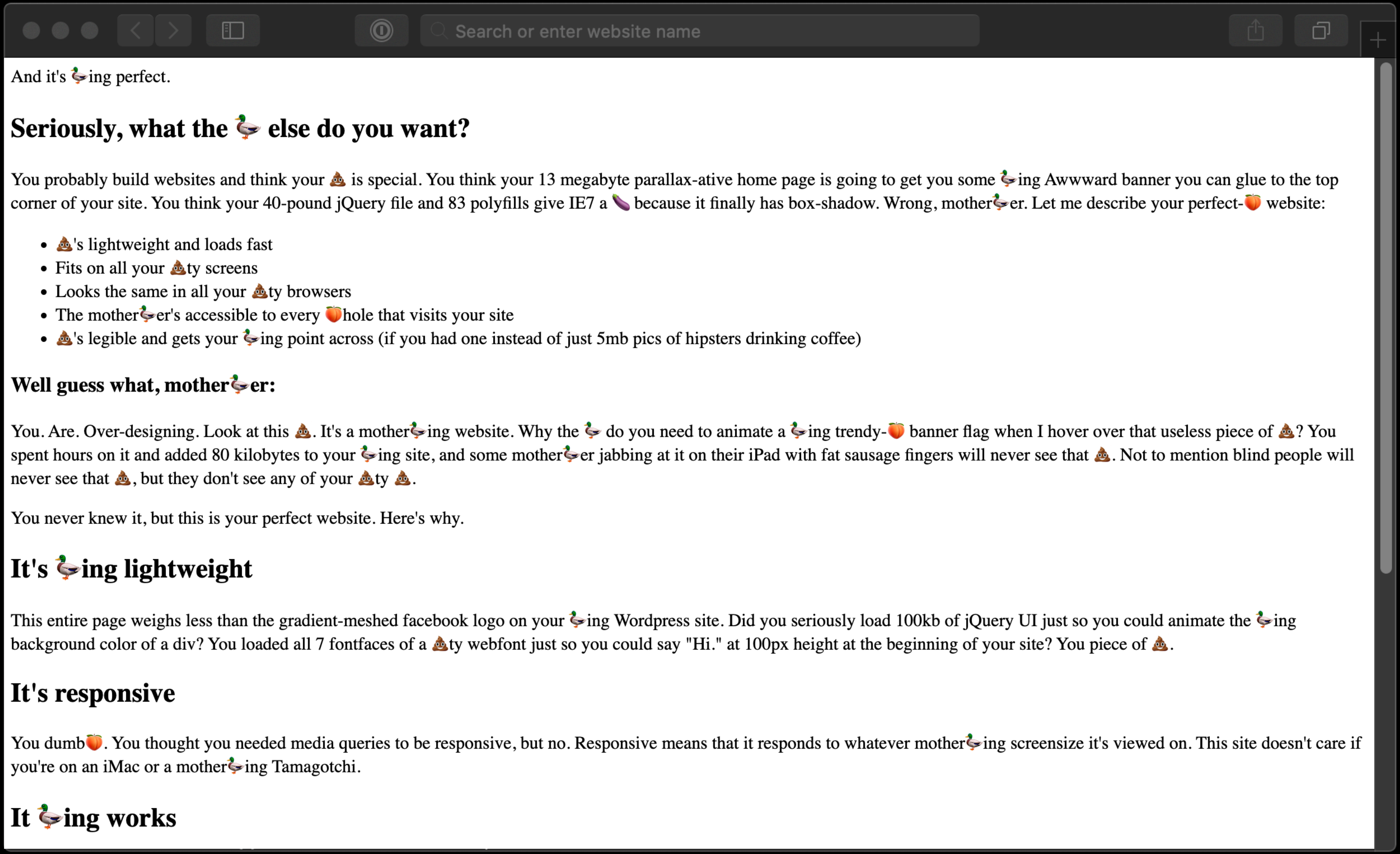


Page Heading

Other less important text

In the absence of other languages,

we *_have_* to use HTML correctly



And it's 🦆ing perfect.

Seriously, what the 🦆 else do you want?

You probably build websites and think your 💩 is special. You think your 13 megabyte parallax-ative home page is going to get you some 🦆ing Awwward banner you can glue to the top corner of your site. You think your 40-pound jQuery file and 83 polyfills give IE7 a 🍆 because it finally has box-shadow. Wrong, mother🦆er. Let me describe your perfect-🍑 website:

- 💩's lightweight and loads fast
- Fits on all your 💩ty screens
- Looks the same in all your 💩ty browsers
- The mother🦆er's accessible to every 🍑hole that visits your site
- 💩's legible and gets your 🦆ing point across (if you had one instead of just 5mb pics of hipsters drinking coffee)

Well guess what, mother🦆er:

You. Are. Over-designing. Look at this 💩. It's a mother🦆ing website. Why the 🦆 do you need to animate a 🦆ing trendy-🍑 banner flag when I hover over that useless piece of 💩? You spent hours on it and added 80 kilobytes to your 🦆ing site, and some mother🦆er jabbing at it on their iPad with fat sausage fingers will never see that 💩. Not to mention blind people will never see that 💩, but they don't see any of your 💩ty 💩.

You never knew it, but this is your perfect website. Here's why.

It's 🦆ing lightweight

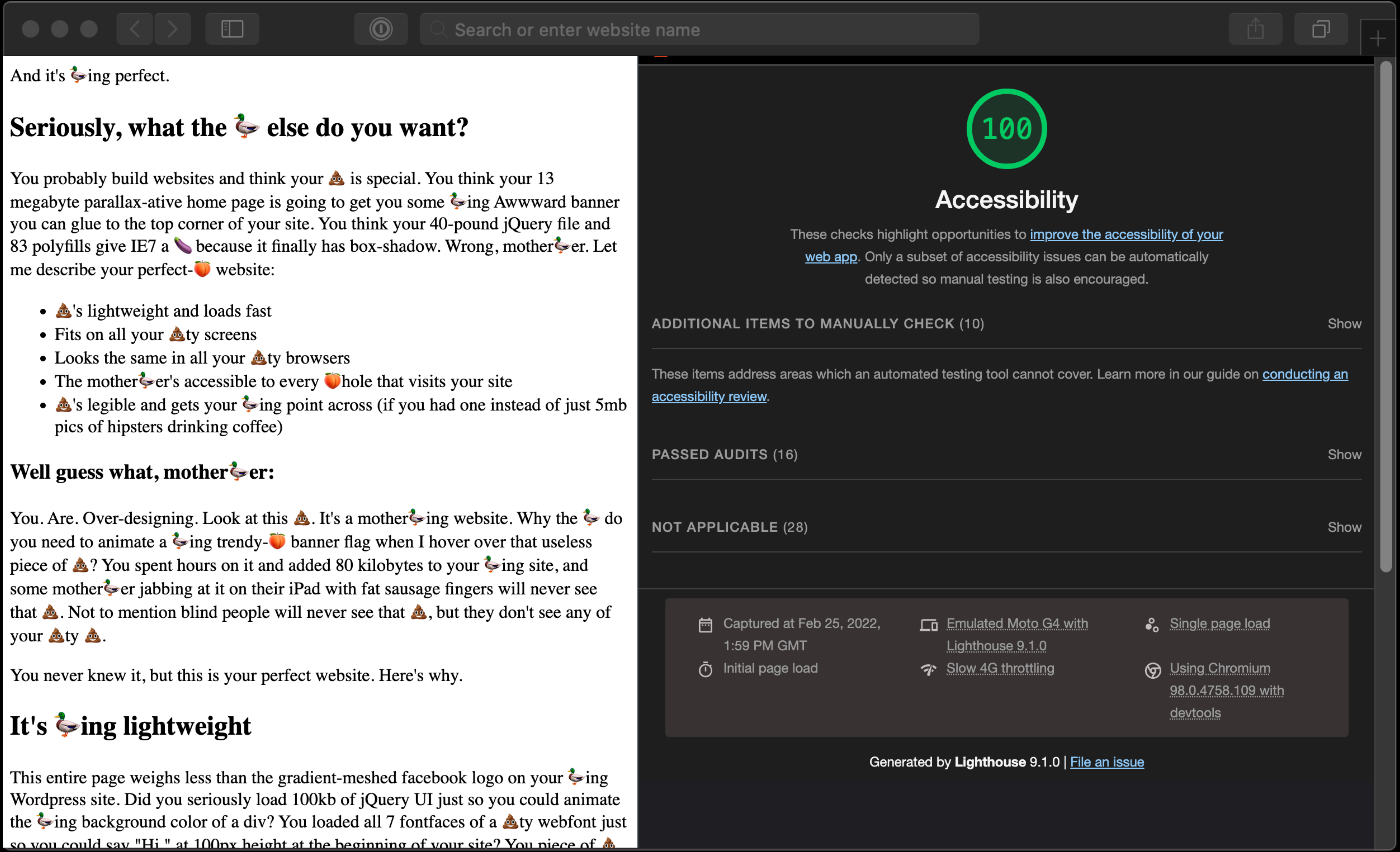
This entire page weighs less than the gradient-meshed facebook logo on your 🦆ing Wordpress site. Did you seriously load 100kb of jQuery UI just so you could animate the 🦆ing background color of a div? You loaded all 7 fontfaces of a 💩ty webfont just so you could say "Hi." at 100px height at the beginning of your site? You piece of 💩.

It's responsive

You dumb🍑. You thought you needed media queries to be responsive, but no. Responsive means that it responds to whatever mother🦆ing screensize it's viewed on. This site doesn't care if you're on an iMac or a mother🦆ing Tamagotchi.

It 🦆ing works

https://ireade.github.io/motherduckingwebsite/



And it's 🦆ing perfect.

Seriously, what the 🦆 else do you want?

You probably build websites and think your 💩 is special. You think your 13 megabyte parallax-ative home page is going to get you some 🦆ing Awwward banner you can glue to the top corner of your site. You think your 40-pound jQuery file and 83 polyfills give IE7 a 🍆 because it finally has box-shadow. Wrong, mother🦆er. Let me describe your perfect-🍑 website:

- 💩's lightweight and loads fast
- Fits on all your 💩ty screens
- Looks the same in all your 💩ty browsers
- The mother🦆er's accessible to every 🍑hole that visits your site
- 💩's legible and gets your 🦆ing point across (if you had one instead of just 5mb pics of hipsters drinking coffee)

Well guess what, mother🦆er:

You. Are. Over-designing. Look at this 💩. It's a mother🦆ing website. Why the 🦆 do you need to animate a 🦆ing trendy-🍑 banner flag when I hover over that useless piece of 💩? You spent hours on it and added 80 kilobytes to your 🦆ing site, and some mother🦆er jabbing at it on their iPad with fat sausage fingers will never see that 💩. Not to mention blind people will never see that 💩, but they don't see any of your 💩ty 💩.

You never knew it, but this is your perfect website. Here's why.

It's 🦆ing lightweight

This entire page weighs less than the gradient-meshed facebook logo on your 🦆ing Wordpress site. Did you seriously load 100kb of jQuery UI just so you could animate the 🦆ing background color of a div? You loaded all 7 fontfaces of a 💩ty webfont just so you could say "Hi." at 100px height at the beginning of your site? You piece of 💩

100

Accessibility

These checks highlight opportunities to improve the accessibility of your web app. Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

ADDITIONAL ITEMS TO MANUALLY CHECK (10) Show

These items address areas which an automated testing tool cannot cover. Learn more in our guide on conducting an accessibility review.

PASSED AUDITS (16) Show

NOT APPLICABLE (28) Show

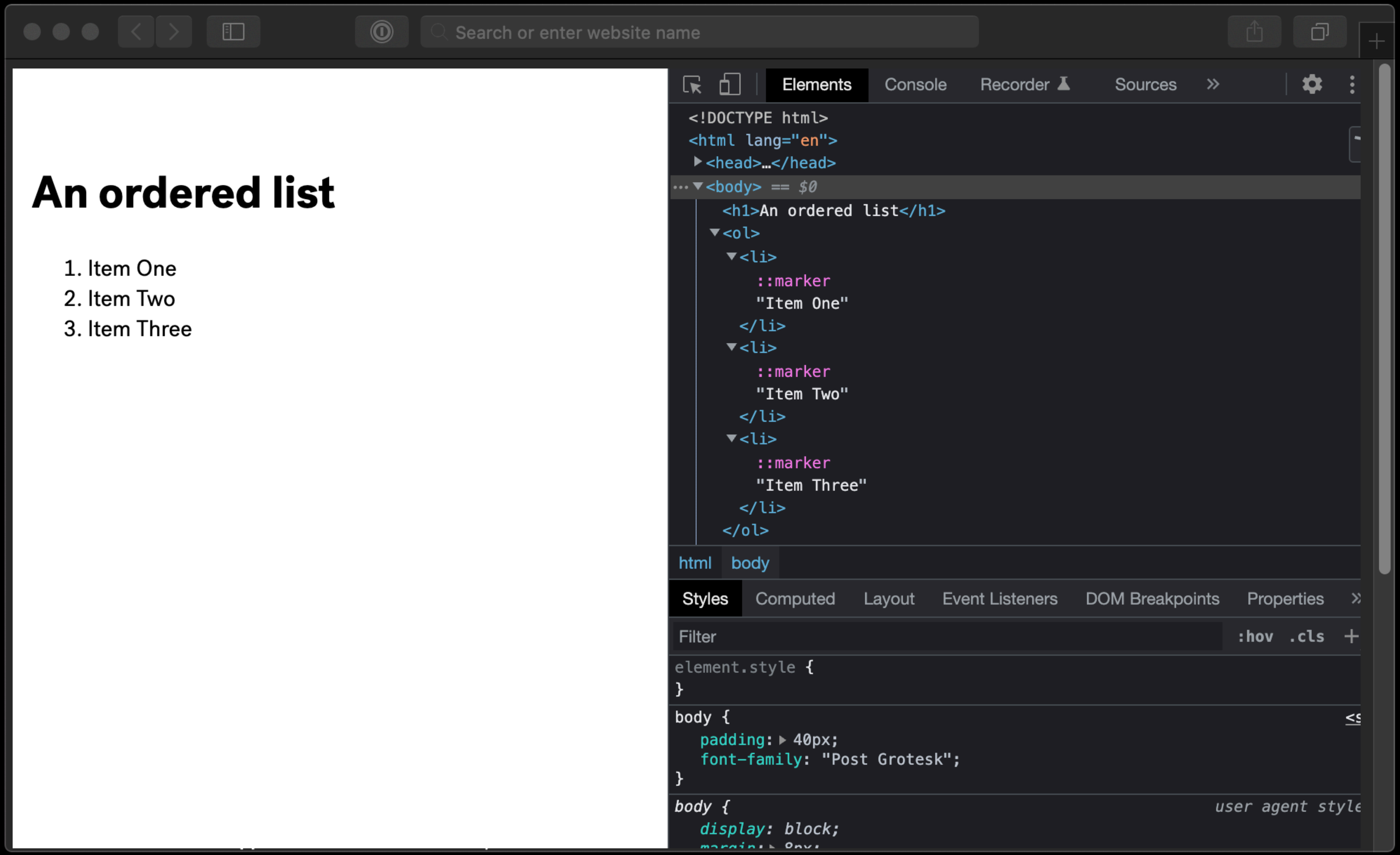
Captured at Feb 25, 2022, 1:59 PM GMT
Initial page load
Emulated Moto G4 with Lighthouse 9.1.0
Slow 4G throttling
Single page load
Using Chromium 98.0.4758.109 with devtools

Generated by **Lighthouse** 9.1.0 | File an issue

https://ireade.github.io/motherduckingwebsite/

By default, HTML is **perceivable** 🧏‍♀️

Guideline 1.3 — Create content that can be presented in different ways ✅



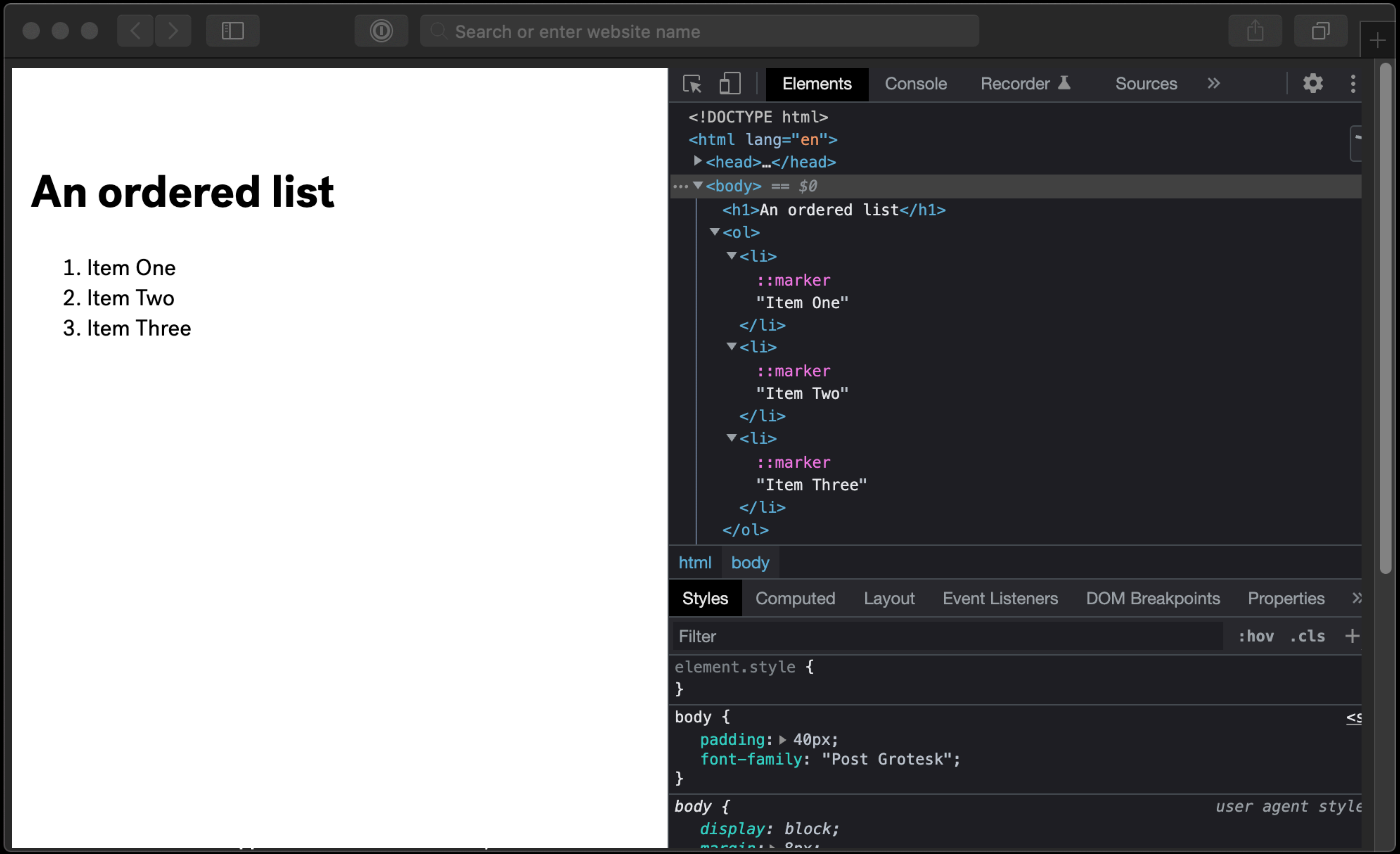
An ordered list

1. Item One
2. Item Two
3. Item Three

```
<!DOCTYPE html>
<html lang="en">
 ▶<head>…</head>
…▼<body> == $0
    <h1>An ordered list</h1>
   ▼<ol>
     ▼<li>
        ::marker
        "Item One"
      </li>
     ▼<li>
        ::marker
        "Item Two"
      </li>
     ▼<li>
        ::marker
        "Item Three"
      </li>
    </ol>
```

```
element.style {
}
body {
   padding: ▶ 40px;
   font-family: "Post Grotesk";
}
body {                                          user agent style
   display: block;
```

Guideline 1.4 — Make it easier for users to see and hear content including separating foreground from background ✅



An ordered list

1. Item One
2. Item Two
3. Item Three

```
<!DOCTYPE html>
<html lang="en">
▶<head>…</head>
▼<body> == $0
    <h1>An ordered list</h1>
  ▼<ol>
    ▼<li>
        ::marker
        "Item One"
      </li>
    ▼<li>
        ::marker
        "Item Two"
      </li>
    ▼<li>
        ::marker
        "Item Three"
      </li>
    </ol>
```

```
element.style {
}
body {
   padding: ▶ 40px;
   font-family: "Post Grotesk";
}
body {                                    user agent style
   display: block;
```

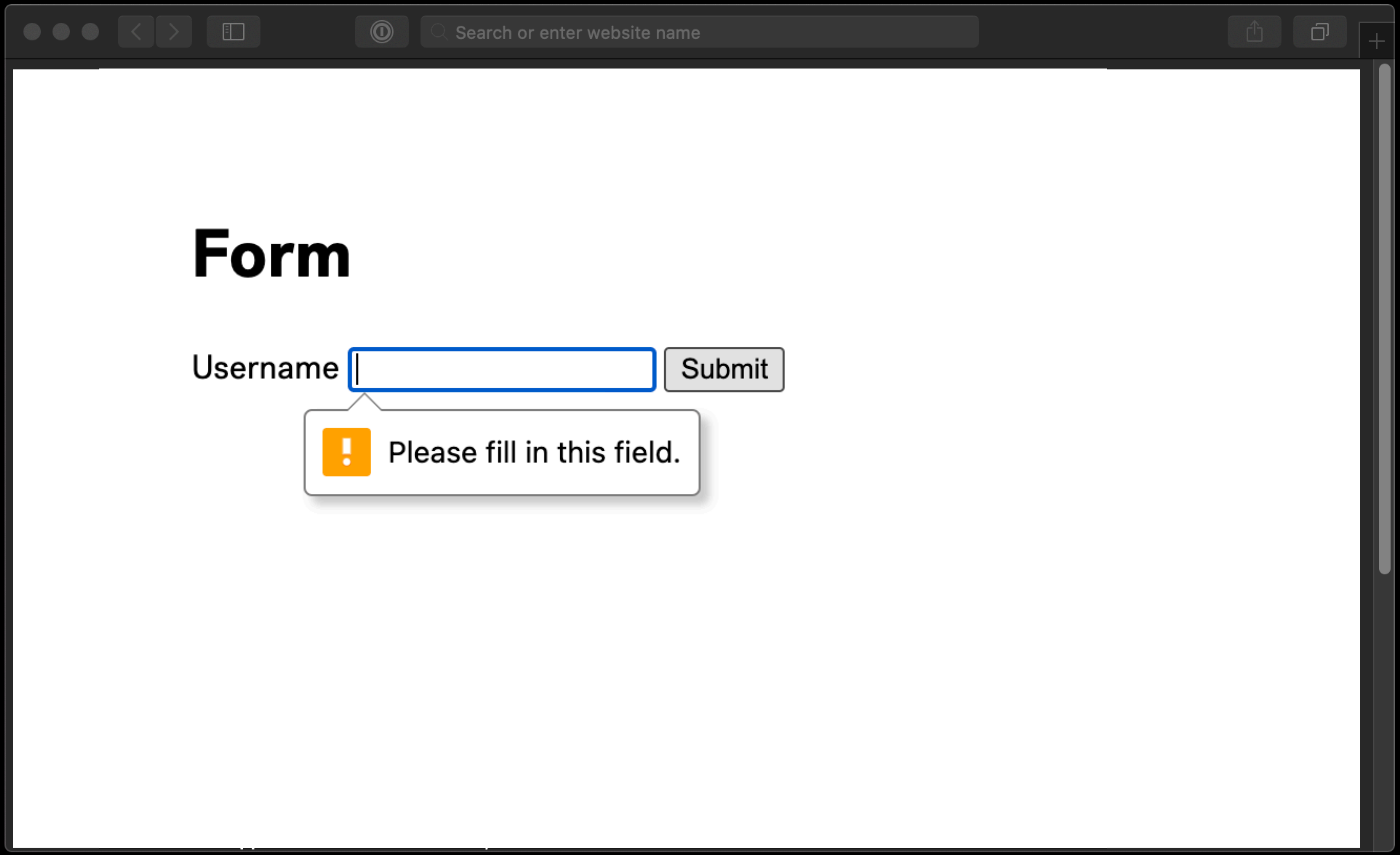
By default, HTML is **operable** 🛠️

Guideline 2.1 — Make all functionality available from a keyboard ✅

Guideline 2.5 — Make it easier for users to operate functionality through various inputs beyond keyboard ✅

By default, HTML is **understandable** 📖

Guideline 3.2 — Make Web pages appear and operate in predictable ways ✅



Form

Username

Please fill in this field.

Guideline 3.3 — Help users avoid and correct mistakes ✅

By default, HTML is **robust** ⚙️

By default, HTML is robust,
if used correctly ✅

HTML ***can*** be inaccessible

Missing text alternatives ☹️

```
<img src=“link/to/really-important-image.png”    >
```

Using the wrong elements ☹️

```
<p>Enter your username:</p>

<input type="text">
```

Weird tab order ☹️

```
<ul>

    <li><a href="/one" tabindex="2">Link One</a></li>

    <li><a href="/two" tabindex="3">Link Two</a></li>

    <li><a href="/three" tabindex="1">Link Three</a></li>

</ul>
```

HTML

CSS
JavaScript

Adding content with CSS ☹️

```
<div id="important"></div>

#important::after {
    content: "Really important information that everyone should know"
}
```

Altering element behaviour with JavaScript ☹️

```
<div onClick=“goToPage(‘/about’)”>
    About Us
</div>
```

90% of accessibility is about **using HTML correctly**.

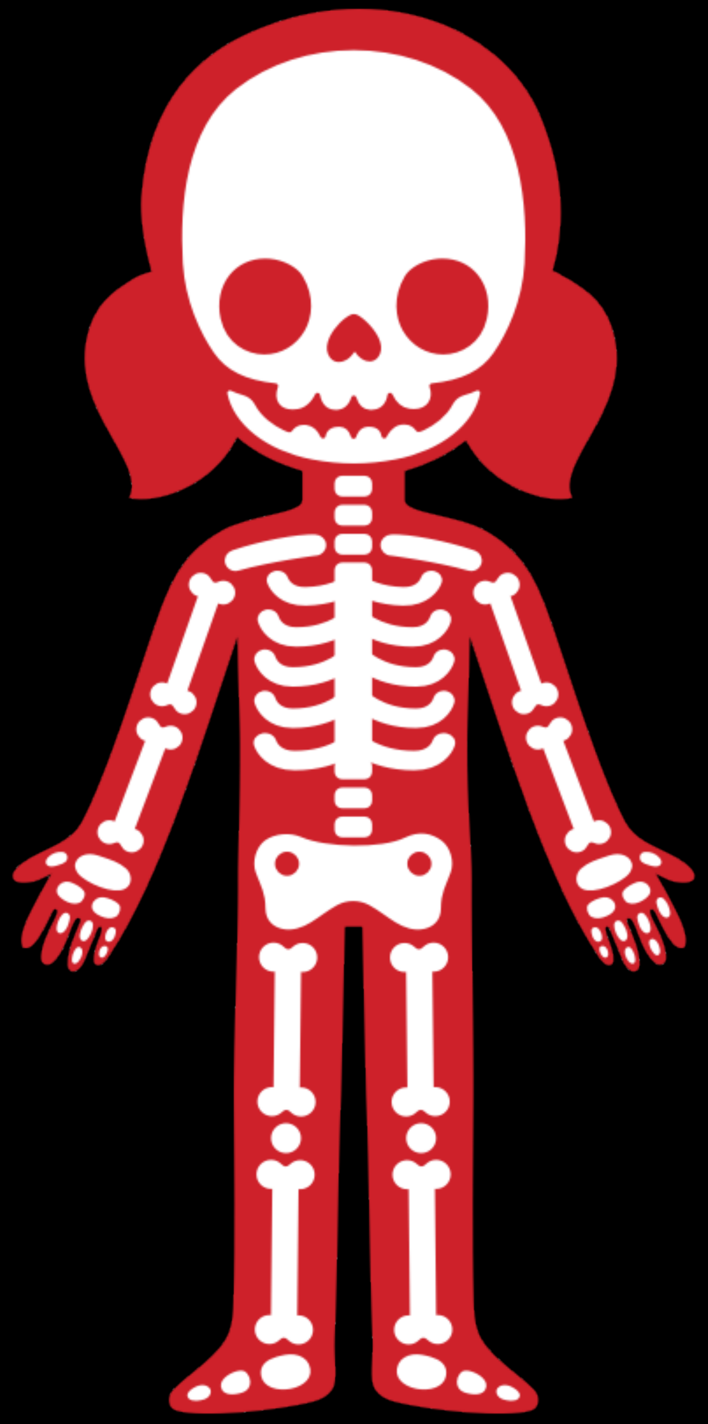
The other 10% is about **not using CSS/JS incorrectly**.

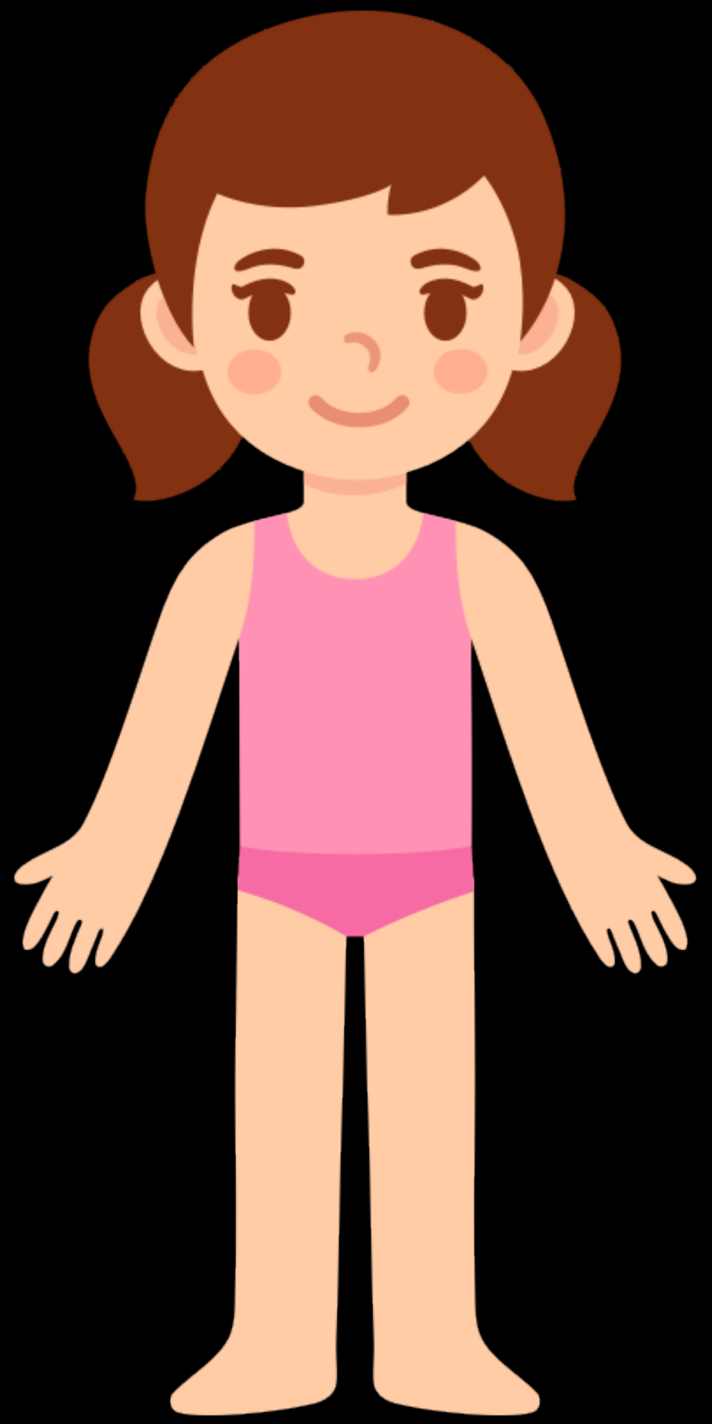
CSS, JavaScript, & Accessibility

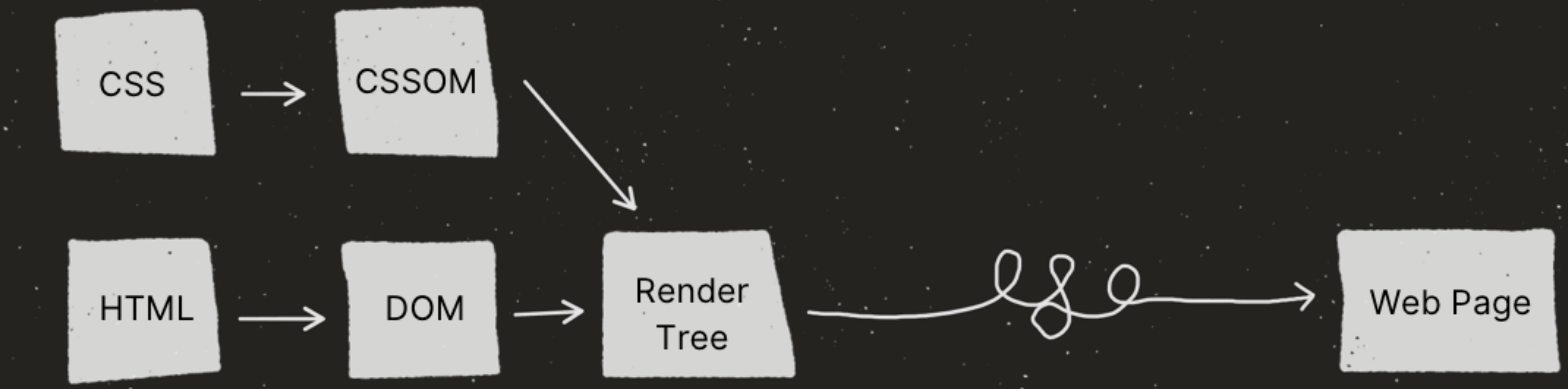
Part One

CSS & Accessibility

CSS is used to **describe the presentation** of an HTML document

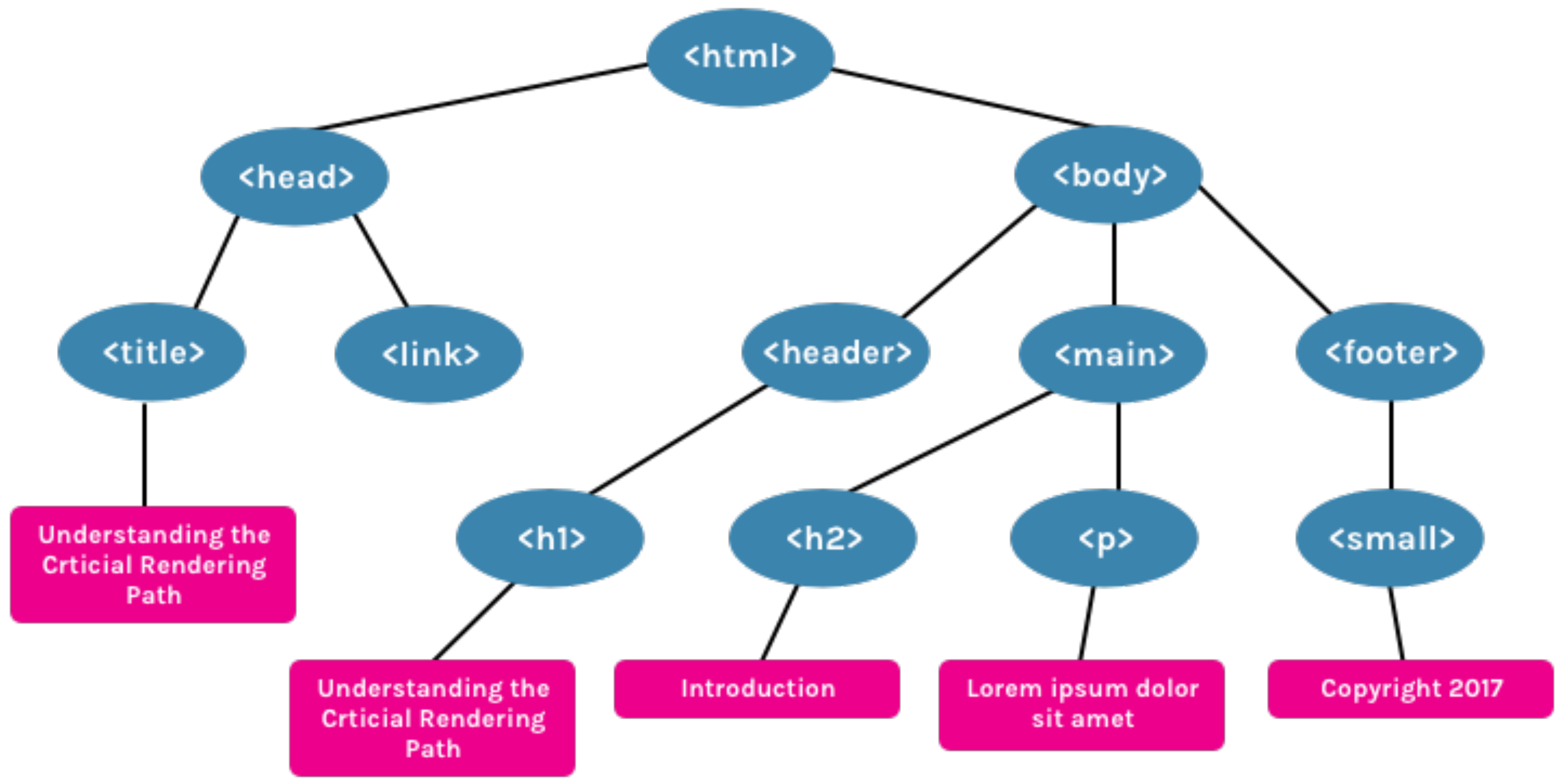

HTML


CSS


CSS
CSSOM
HTML
DOM
Render Tree
Web Page

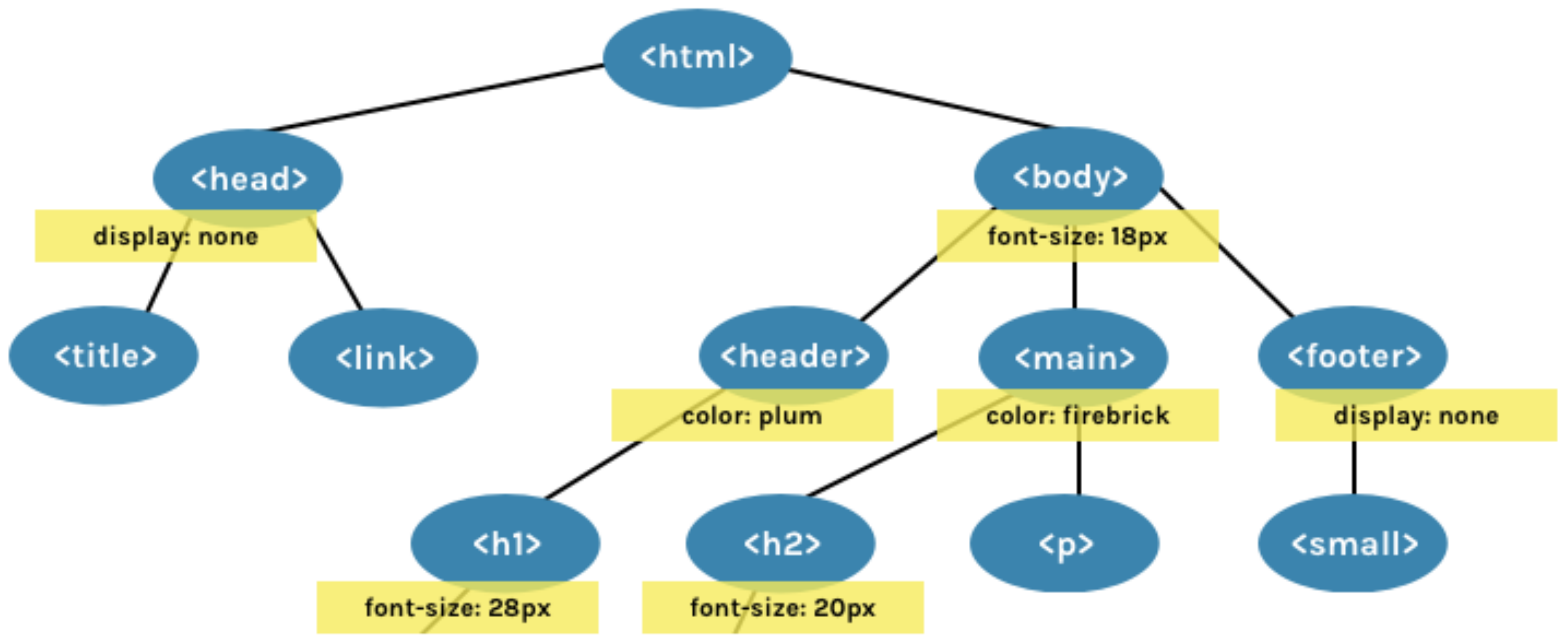
DOM

```
<head>
    <title>Understanding the Critical Rendering Path</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <header>
        <h1>Understanding the Critical Rendering Path</h1>
    </header>
    <main>
        <h2>Introduction</h2>
        <p>Lorem ipsum dolor sit amet</p>
    </main>
    <footer><small>Copyright 2017</small></footer>
</body>
```


<html>
<head>
<body>
<title>
<link>
<header>
<main>
<footer>
Understanding the Crticial Rendering Path
<h1>
<h2>
<p>
<small>
Understanding the Crticial Rendering Path
Introduction
Lorem ipsum dolor sit amet
Copyright 2017

CSSOM

```
body { font-size: 18px; }


header { color: plum; }

h1 { font-size: 28px; }


main { color: firebrick; }

h2 { font-size: 20px; }


footer { display: none; }
```

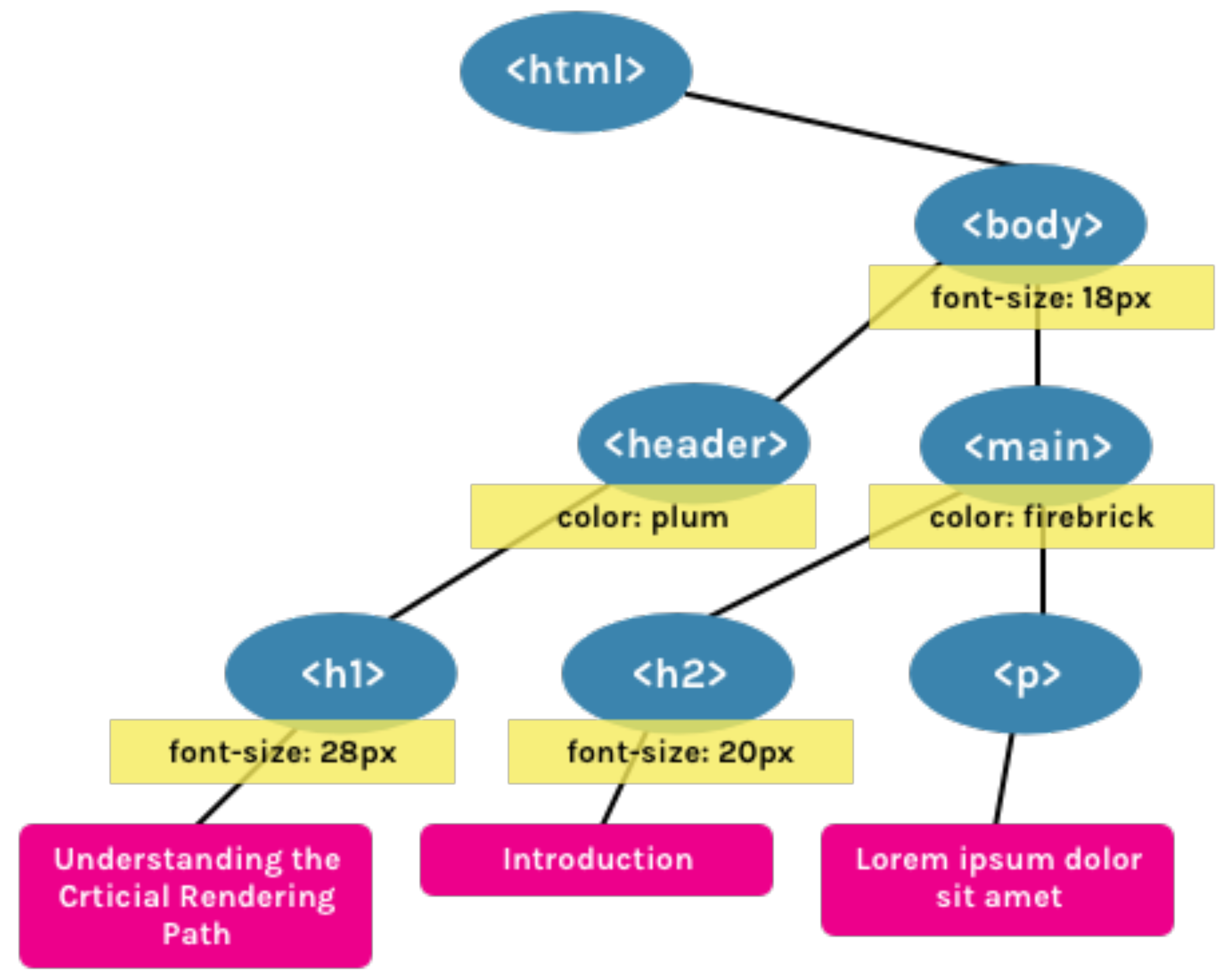


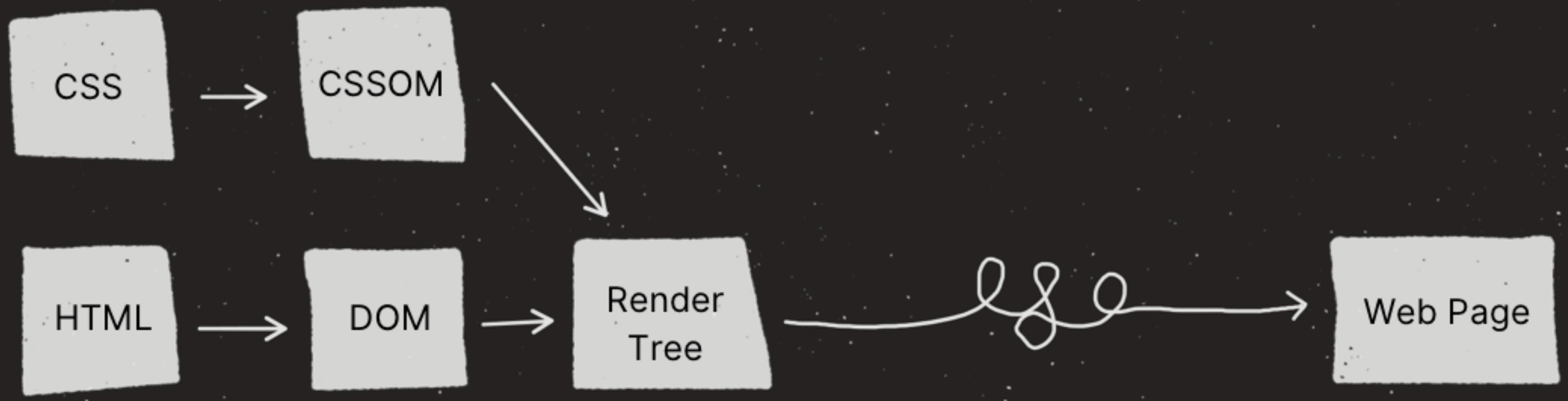
DOM

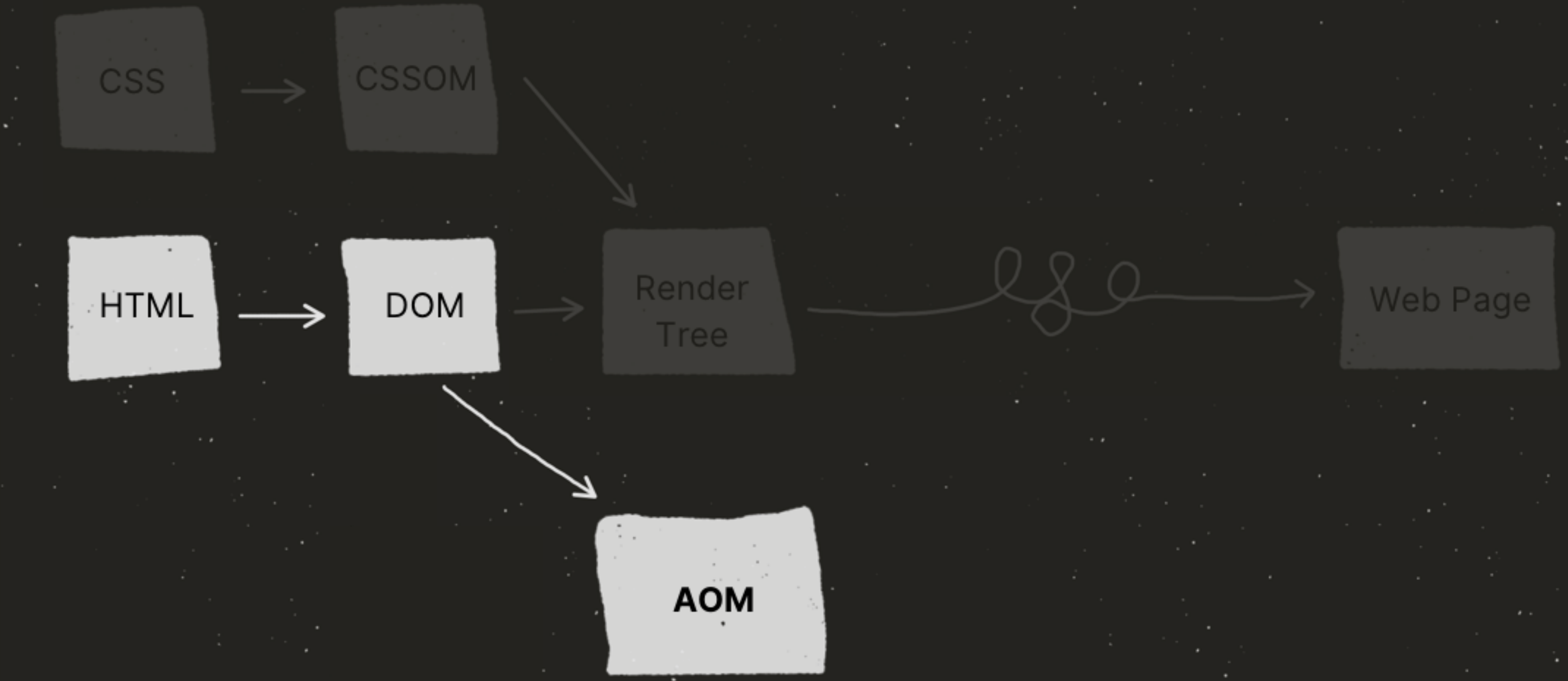
+ CSSOM

- Non-visible elements

= Render tree

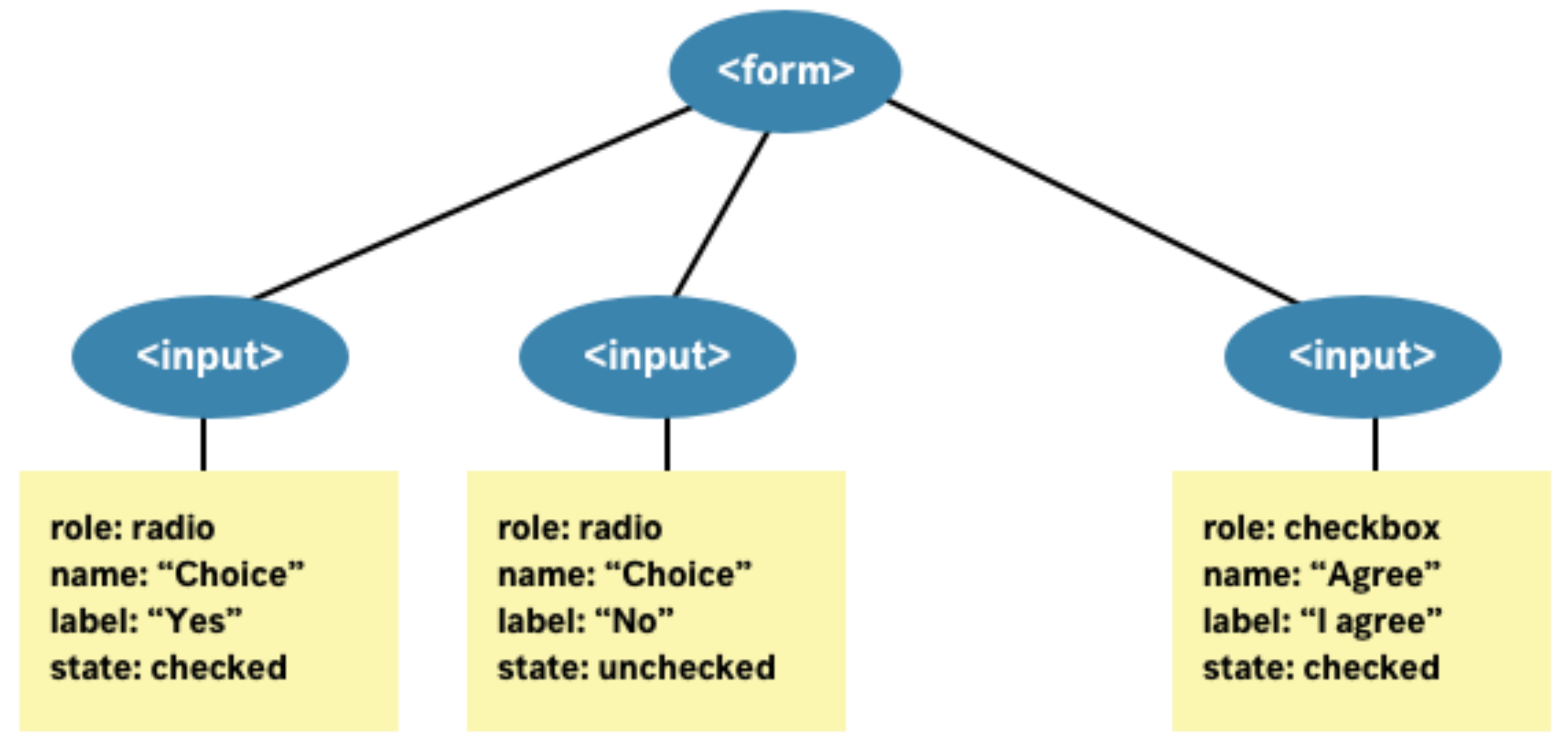


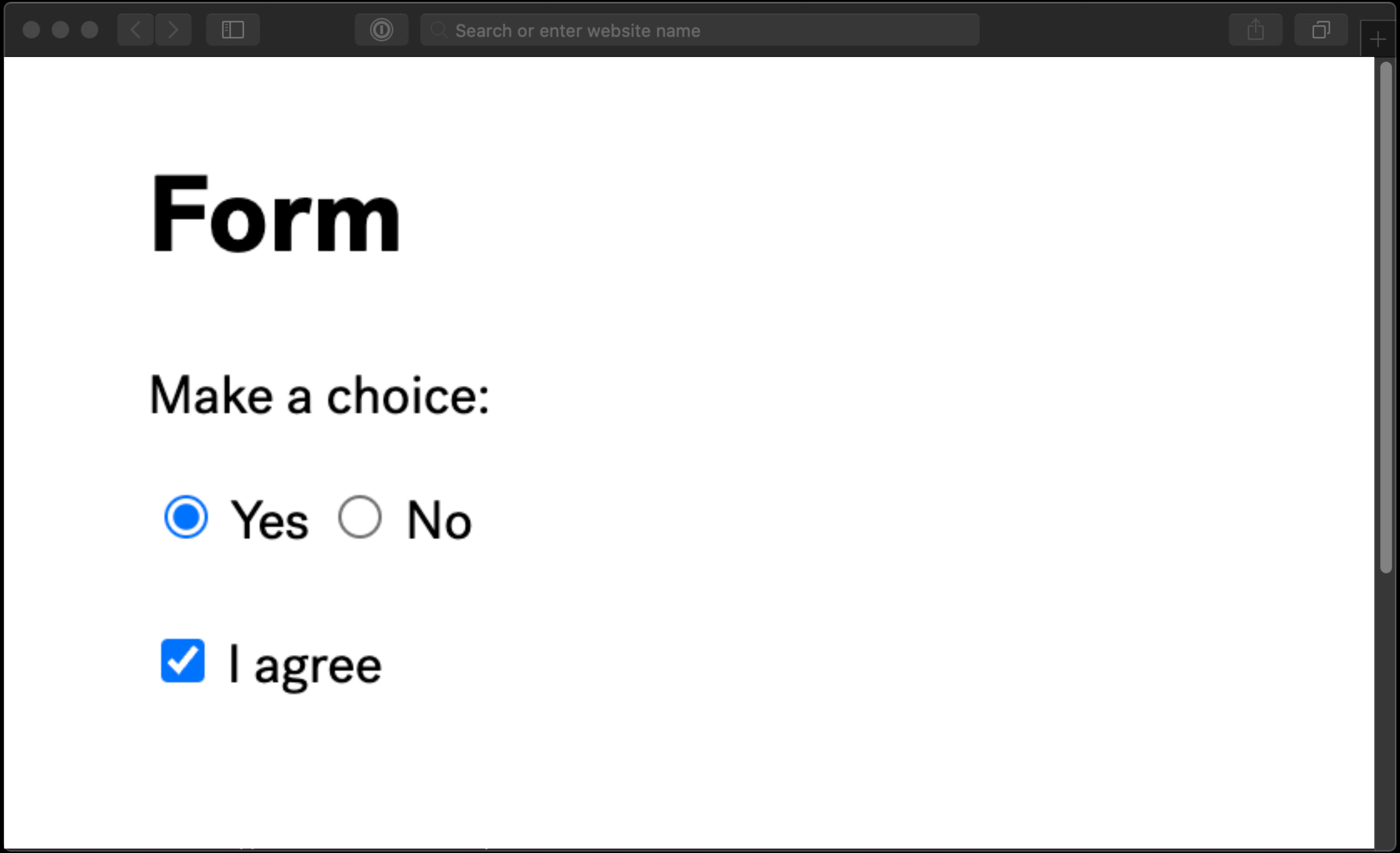

CSS
CSSOM
HTML
DOM
Render Tree
Web Page


CSS
CSSOM
HTML
DOM
Render Tree
Web Page
AOM

AOM

```
<form>
    <p>Make a choice:</p>
    <input type="radio" name="choice" id="yes" />
    <label>Yes</label>
    <input type="radio" name="choice" id="no" />
    <label>No</label>

    <input type="checkbox" name="agree" id="agree" />
    <label>I agree</label>
</form>
```





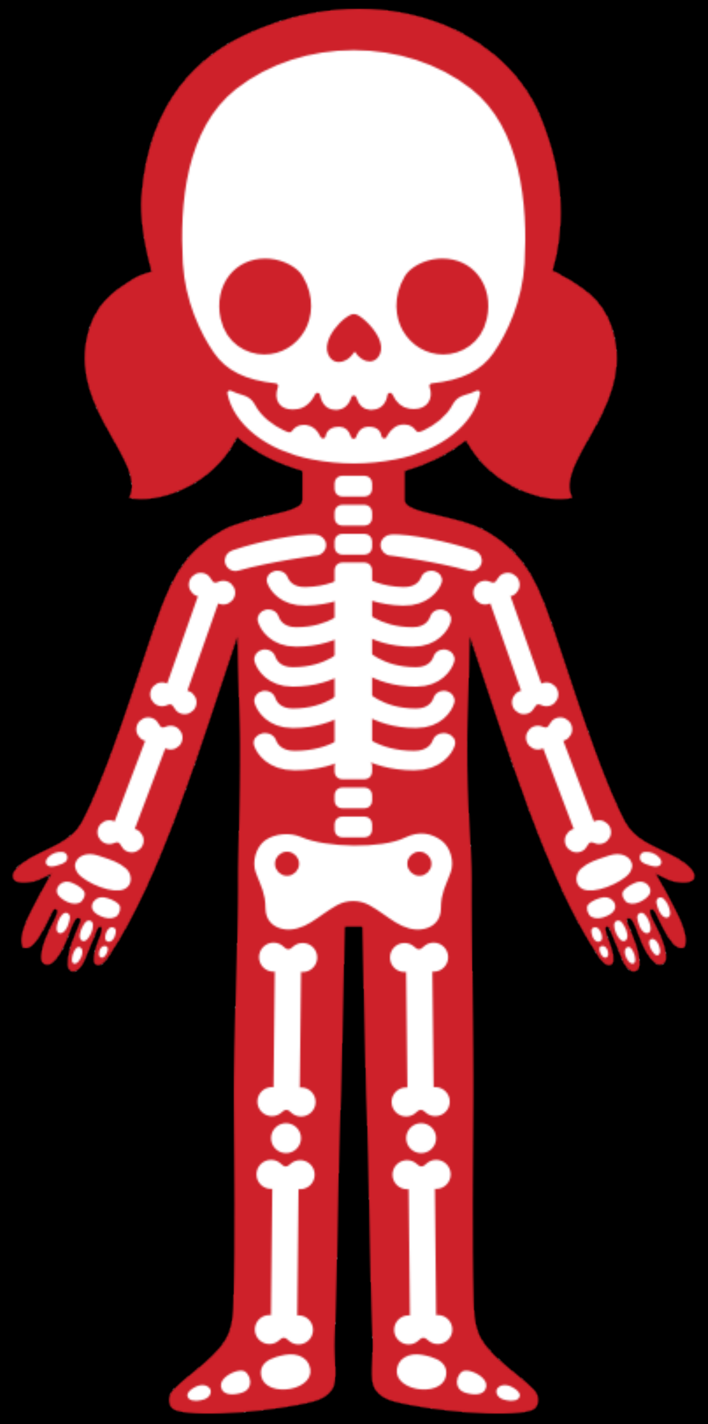
Search or enter website name

Form

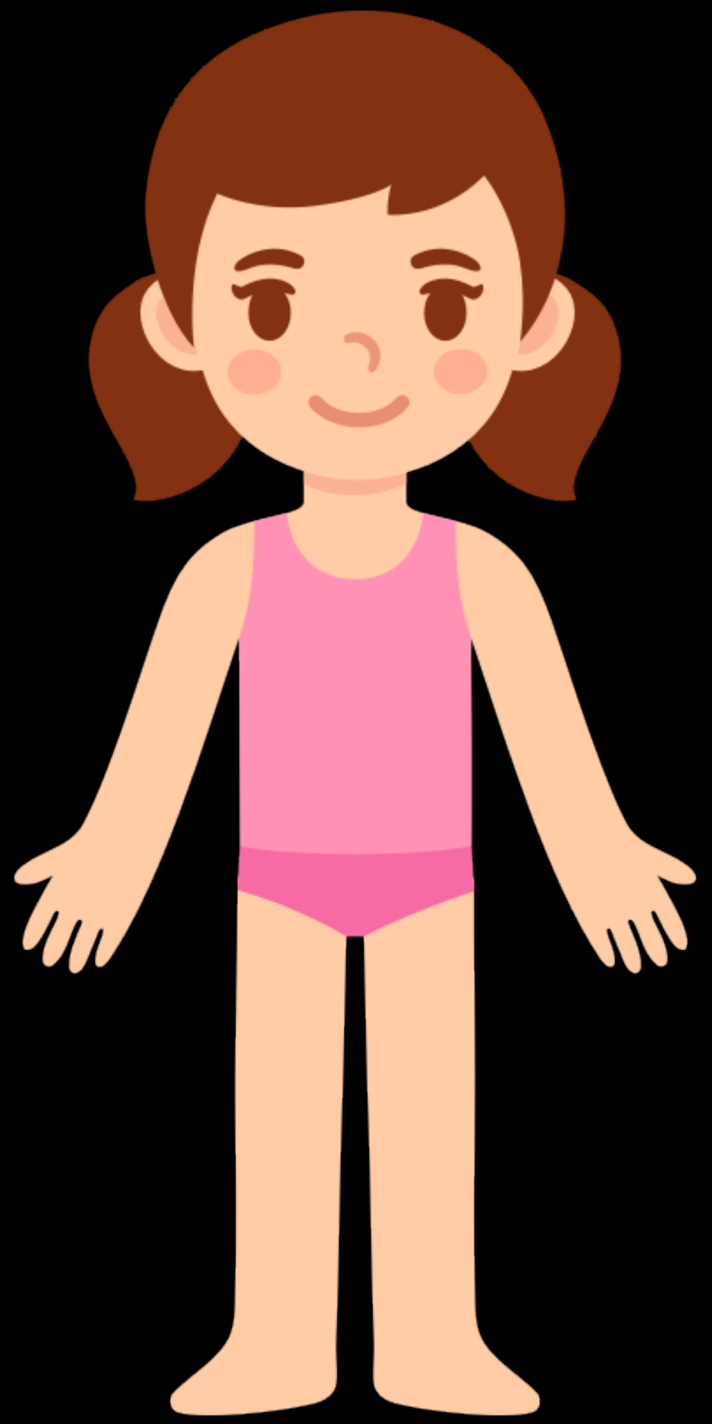
Make a choice:

[✓ Yes] No

✓ I agree



HTML

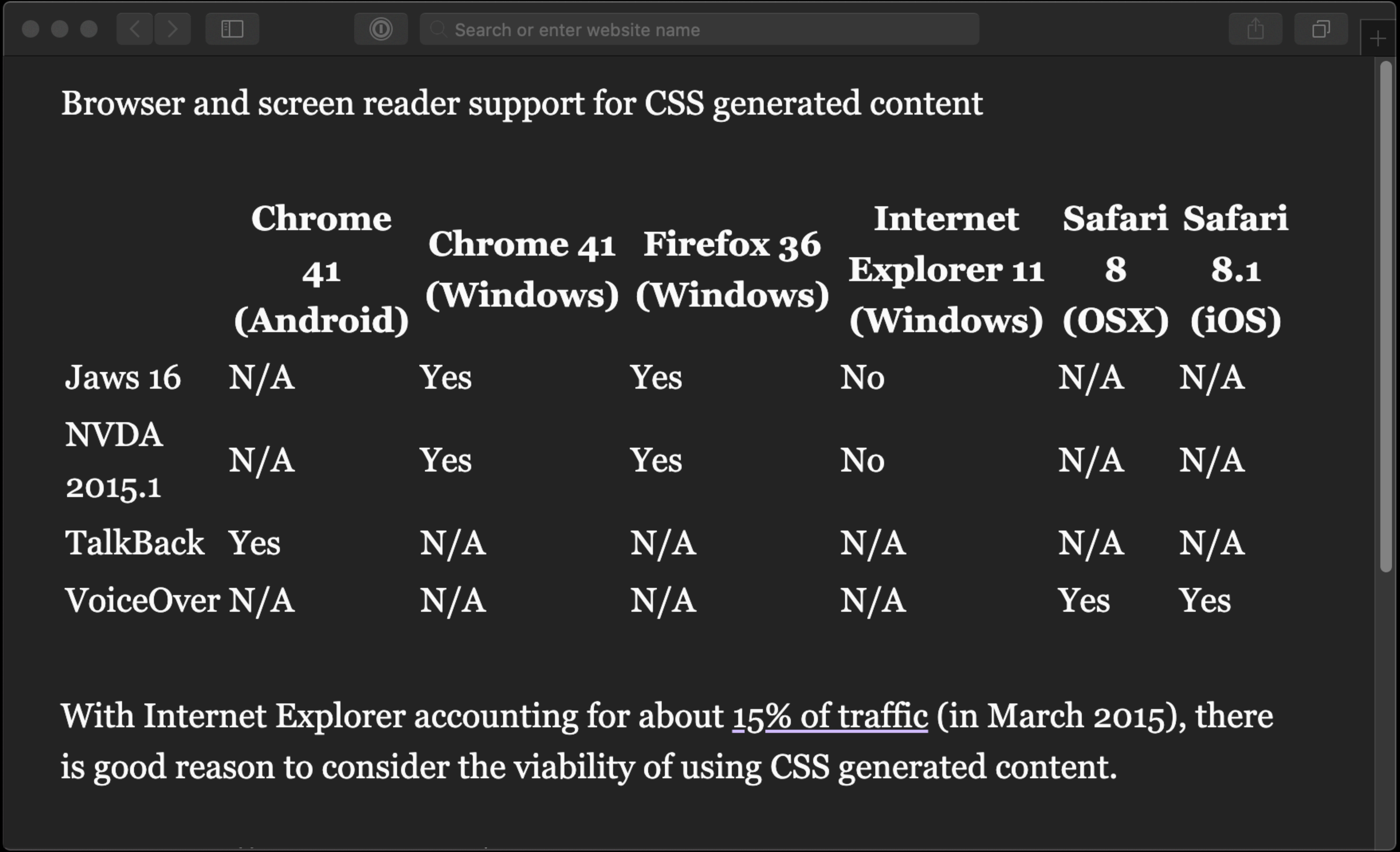


CSS

#1

Don’t use CSS to convey meaning or content

```
h1::after {
    content: “My Page Title”
}
```


Search or enter website name

Browser and screen reader support for CSS generated content

	Chrome 41 (Android)	Chrome 41 (Windows)	Firefox 36 (Windows)	Internet Explorer 11 (Windows)	Safari 8 (OSX)	Safari 8.1 (iOS)
Jaws 16	N/A	Yes	Yes	No	N/A	N/A
NVDA 2015.1	N/A	Yes	Yes	No	N/A	N/A
TalkBack	Yes	N/A	N/A	N/A	N/A	N/A
VoiceOver	N/A	N/A	N/A	N/A	Yes	Yes

With Internet Explorer accounting for about 15% of traffic (in March 2015), there is good reason to consider the viability of using CSS generated content.

https://tink.uk/accessibility-support-for-css-generated-content/

“In other words, use CSS generated content to change or supplement the design, **but not to create or alter important content** on the page.”

— Léonie Watson

https://tink.uk/accessibility-support-for-css-generated-content/

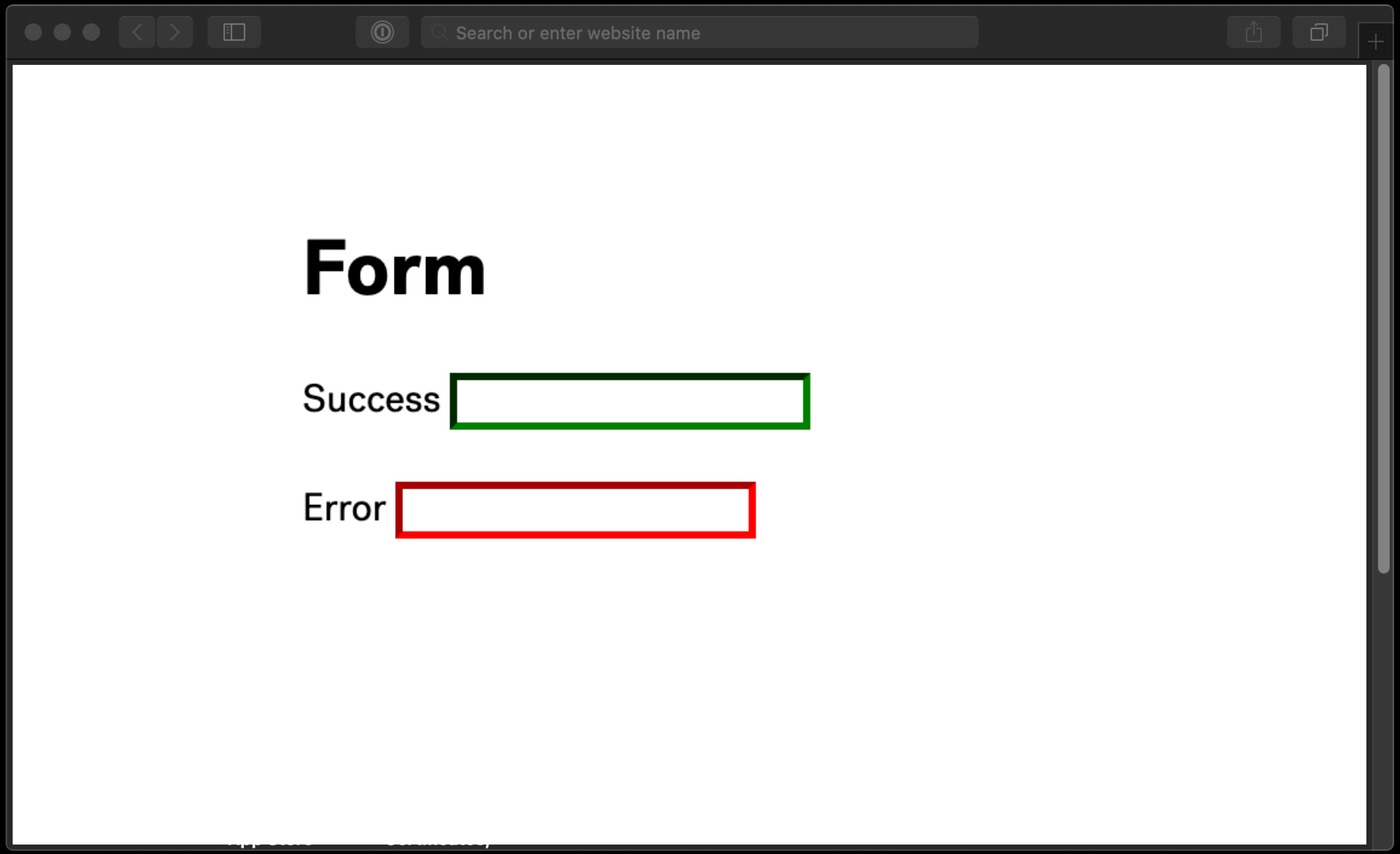
❌

```
.element::after {
    content: /* any content */
}
```

✅

```
.element::after {
    content: “”
}
```

```
<input class="error" >

<input class="success" >


.error { border-color: red; }

.success { border-color: green; }
```

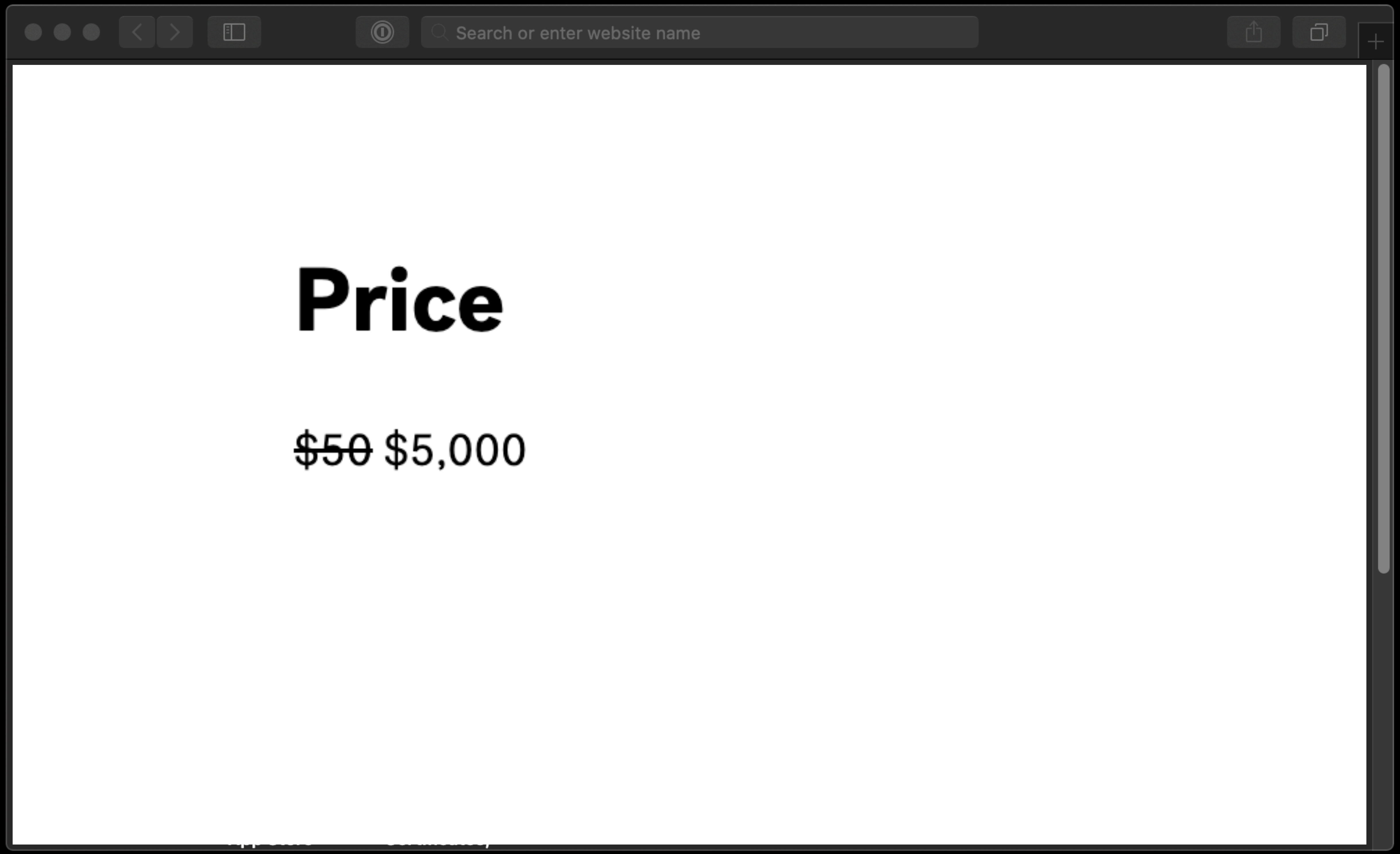


Form

Success

Error

```
<p>Price:</p>

<span class="strike">$50</span>

<span>$5,000</span>



.strike {

    text-decoration: line-through;

}
```



Price

~~$50~~ $5,000

#2

Don’t use CSS to change the semantics of HTML

❌

```
This is

<span class="emphasise">really</span>

important


.emphasise { font-style: italic; }
```

✅

```
This is <em>really</em> important
```

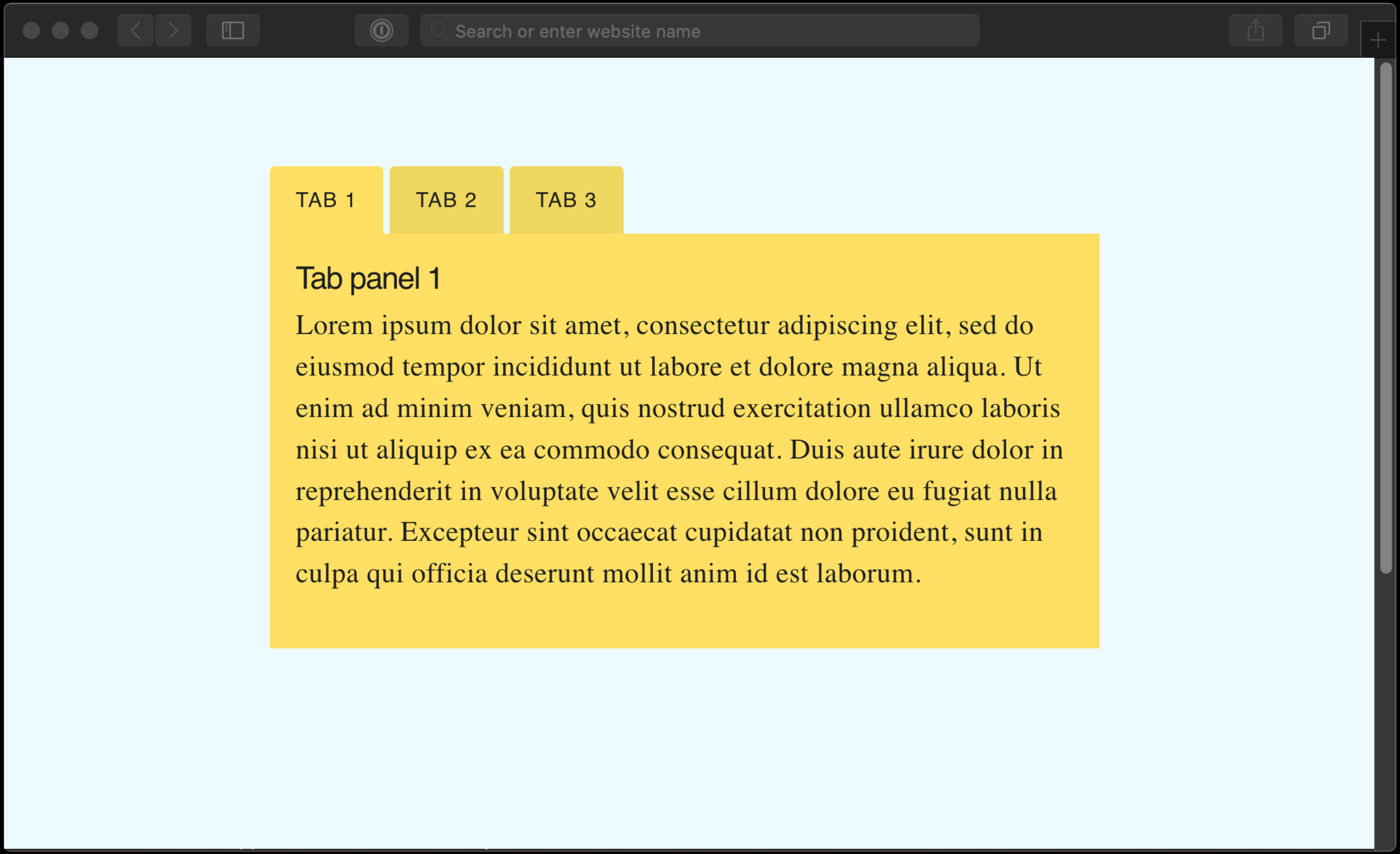
	`display: none;`	`visibility: hidden;`	`opacity: 0;`	`position: absolute;`
				`top: -9999px;`
				`left: -9999px;`

Visually hidden?	✅	✅	✅	✅

	`display: none;`	`visibility: hidden;`	`opacity: 0;`	`position: absolute;`
				`top: -9999px;`
				`left: -9999px;`

Visually hidden?	✅	✅	✅	✅
Box model generated?	❌	✅	✅	✅
Affects layout?	❌	✅	✅	❌
Read by assistive technologies?	❌	❌	✅	✅

	`display: none;`	`visibility: hidden;`	`opacity: 0;`	`position: absolute; top: -9999px; left: -9999px;`
Visually hidden?	✅	✅	✅	✅
Box model generated?	❌	✅	✅	✅
Affects layout?	❌	✅	✅	❌
Read by assistive technologies?	❌	❌	✅	✅



Tab panel 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

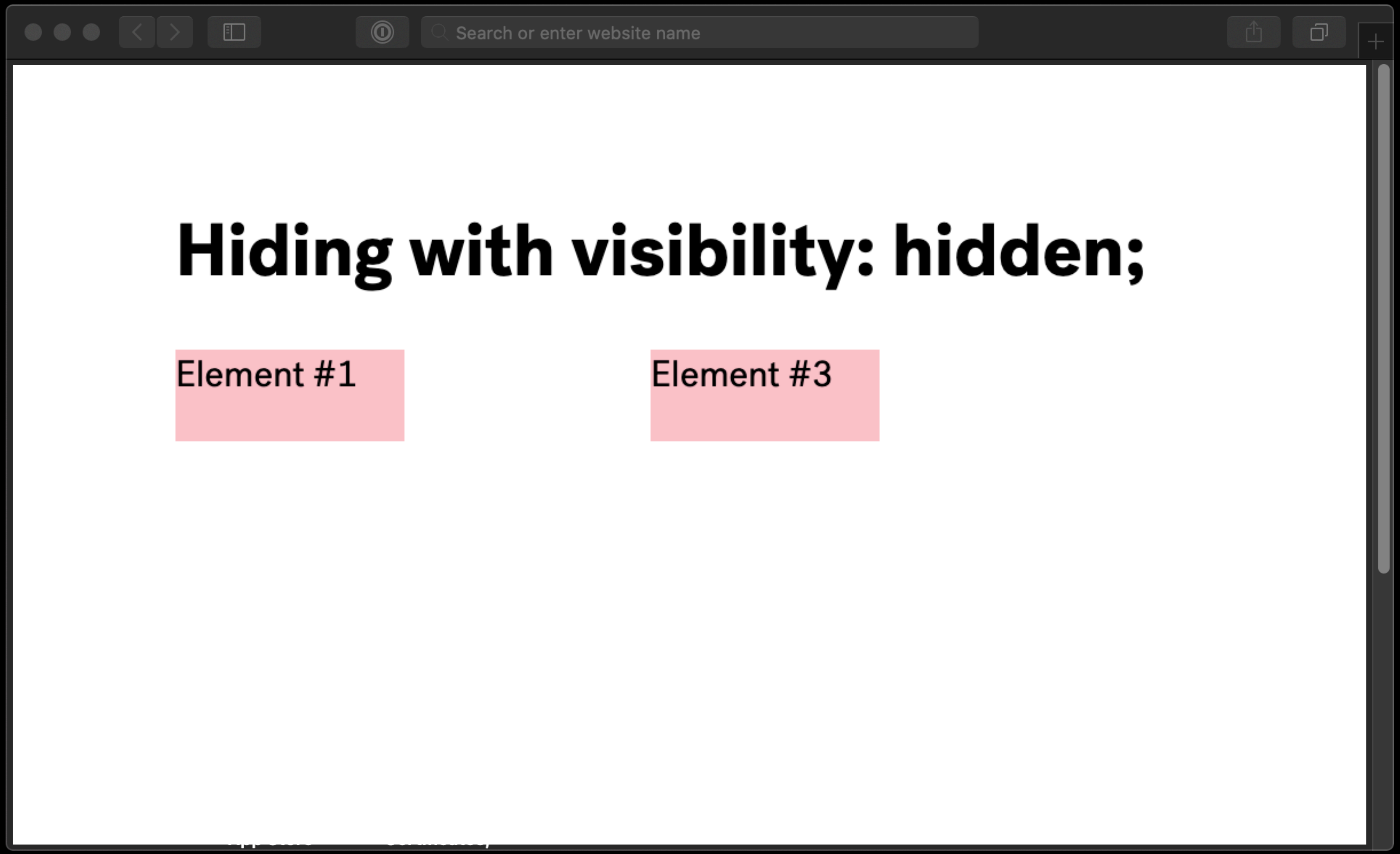
https://frend.co/components/tabs/

	`display: none;`	`visibility: hidden;`	`opacity: 0;`	`position: absolute;`
				`top: -9999px;`
				`left: -9999px;`

Visually hidden?	✅	✅	✅	✅
Box model generated?	❌	✅	✅	✅
Affects layout?	❌	✅	✅	❌
Read by assistive technologies?	❌	❌	✅	✅

	`display: none;`	`visibility: hidden;`	`opacity: 0;`	`position: absolute; top: -9999px; left: -9999px;`
Visually hidden?	✅	✅	✅	✅
Box model generated?	❌	✅	✅	✅
Affects layout?	❌	✅	✅	❌
Read by assistive technologies?	❌	❌	✅	✅

```
<div>Element #1</div>



<div style="visibility:hidden;">

  Element #2

</div>



<div>Element #3</div>
```



Hiding with visibility: hidden;

Element #1

Element #3

Don’t use `visibility: hidden;`

	`display: none;`	`visibility: hidden;`	`opacity: 0;`	`position: absolute; top: -9999px; left: -9999px;`
Visually hidden?	✅	✅	✅	✅
Box model generated?	❌	✅	✅	✅
Affects layout?	❌	✅	✅	❌
Read by assistive technologies?	❌	❌	✅	✅

	`display: none;`	`visibility: hidden;`	`opacity: 0;`	`position: absolute;` `top: -9999px;` `left: -9999px;`
Visually hidden?	✅	✅	✅	✅
Box model generated?	❌	✅	✅	✅
Affects layout?	❌	✅	✅	❌
Read by assistive technologies?	❌	❌	✅	✅

Visual Order

Source Order

Source order

```
<body>

    <footer>...</footer>

    <nav>...</nav>

    <header>...</header>

    <main>...</main>

</body>
```

Visual order

```
body { display: flex }

header { order: 1 }

nav { order: 2 }

main { order: 3 }

footer { order: 4 }
```

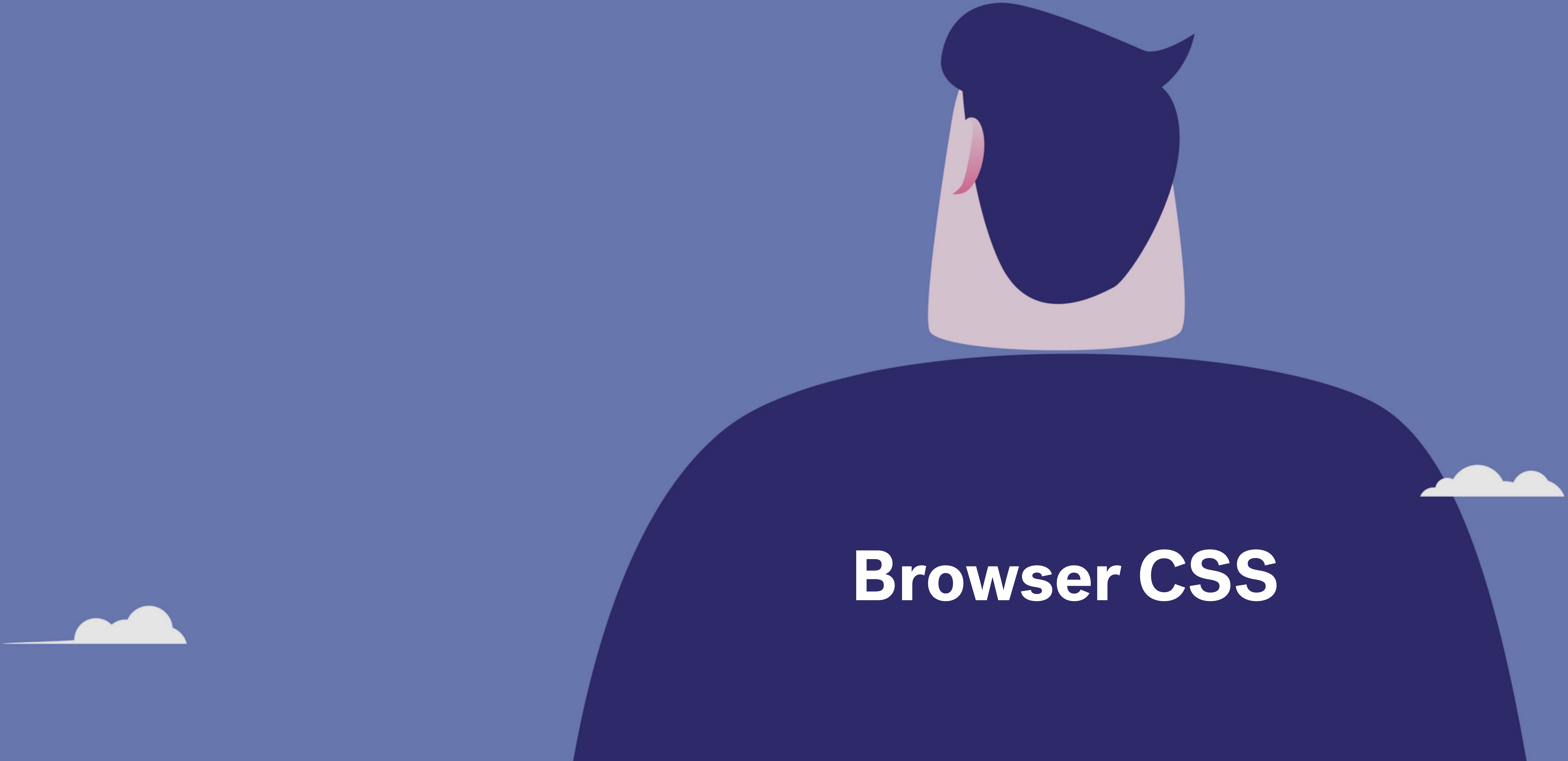
“With this power comes **great responsibility**”

— Rachel Andrew

https://rachelandrew.co.uk/archives/2015/07/28/modern-css-layout-power-and-responsibility/

#3

Don’t write CSS that undoes the default accessible styles

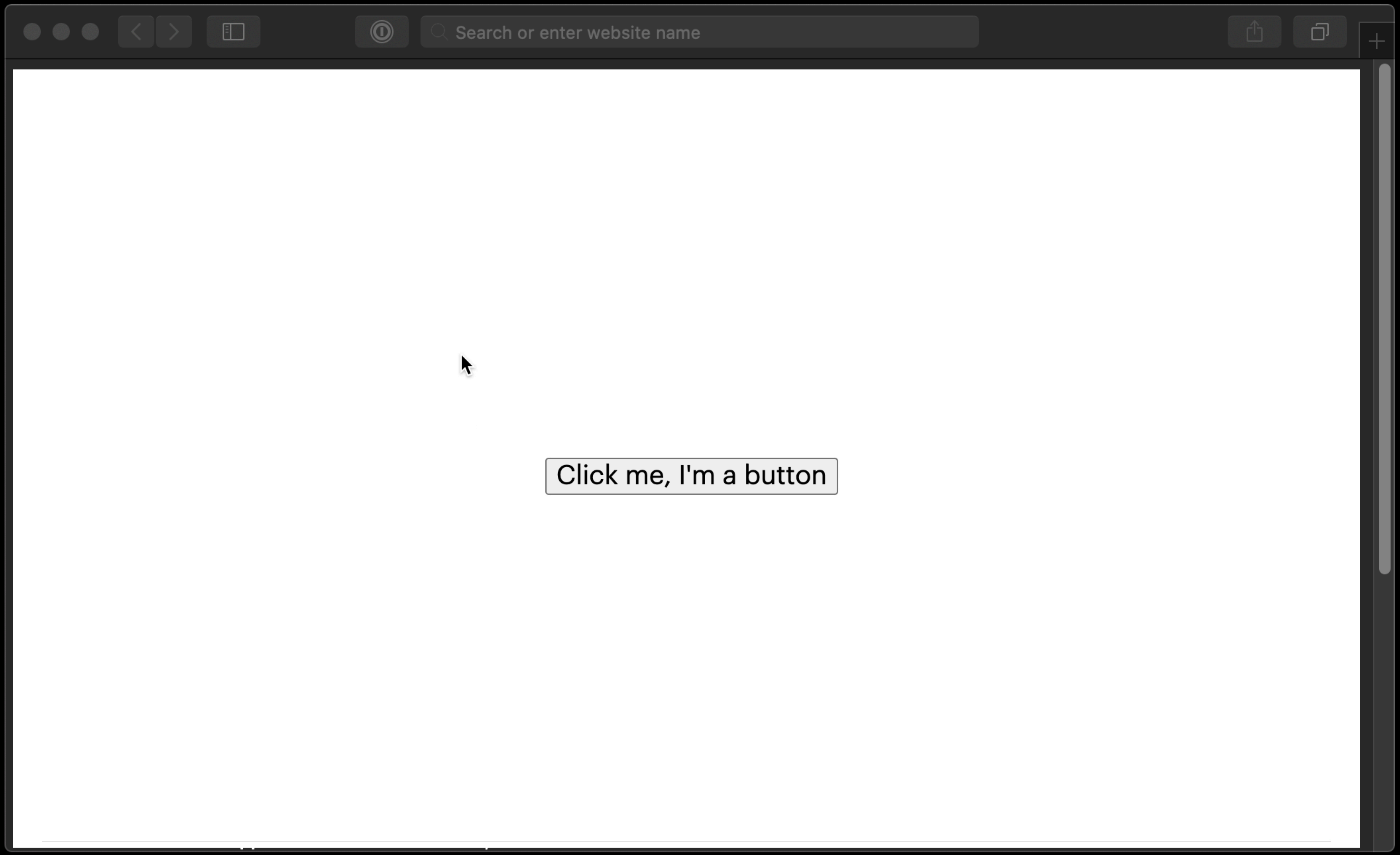


Browser CSS

✅ **Text & element sizes**

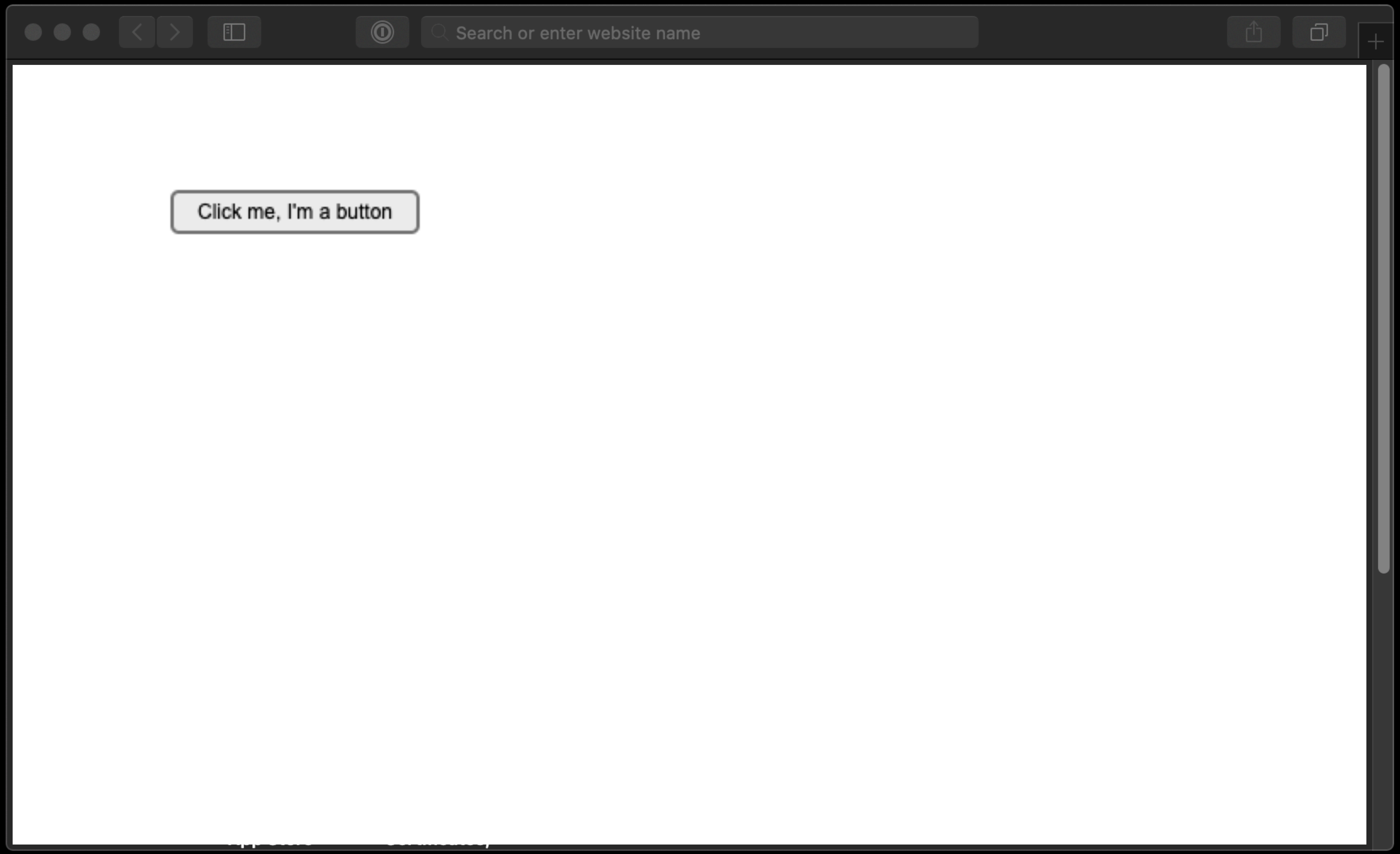
✅ **Colours & contrast**

✅ **Hover, focus, & active states**

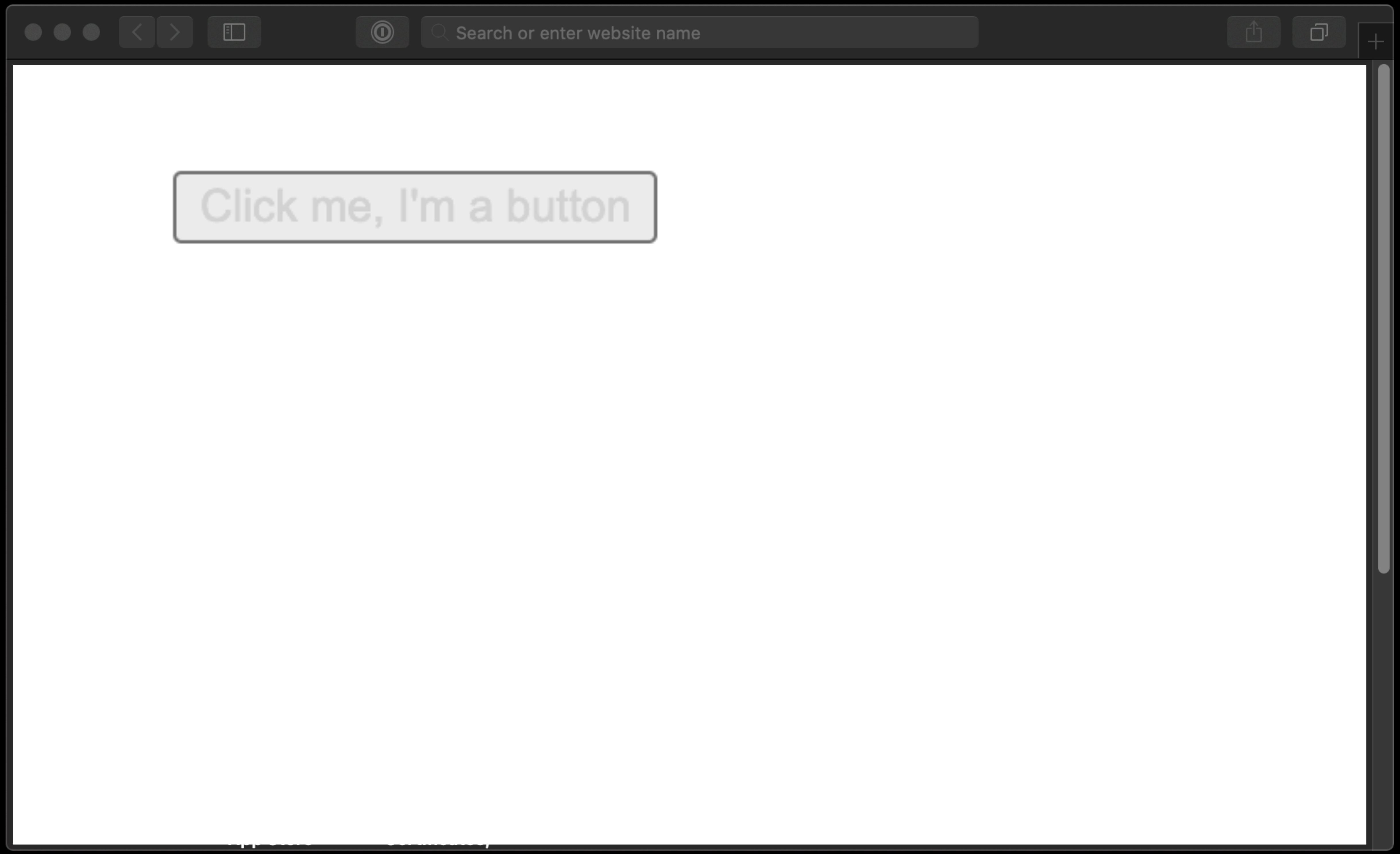


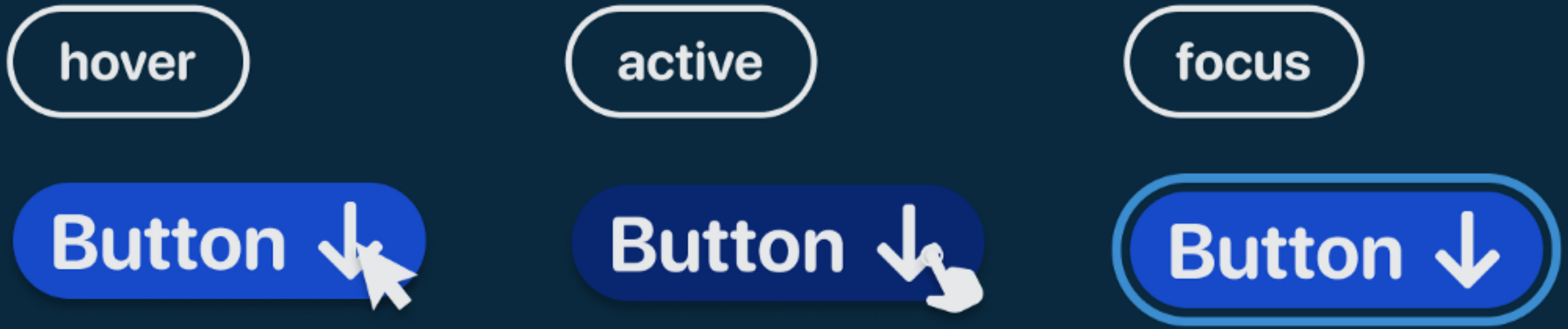
My CSS
Browser CSS

```
button {

    font-size: 6px;

}
```

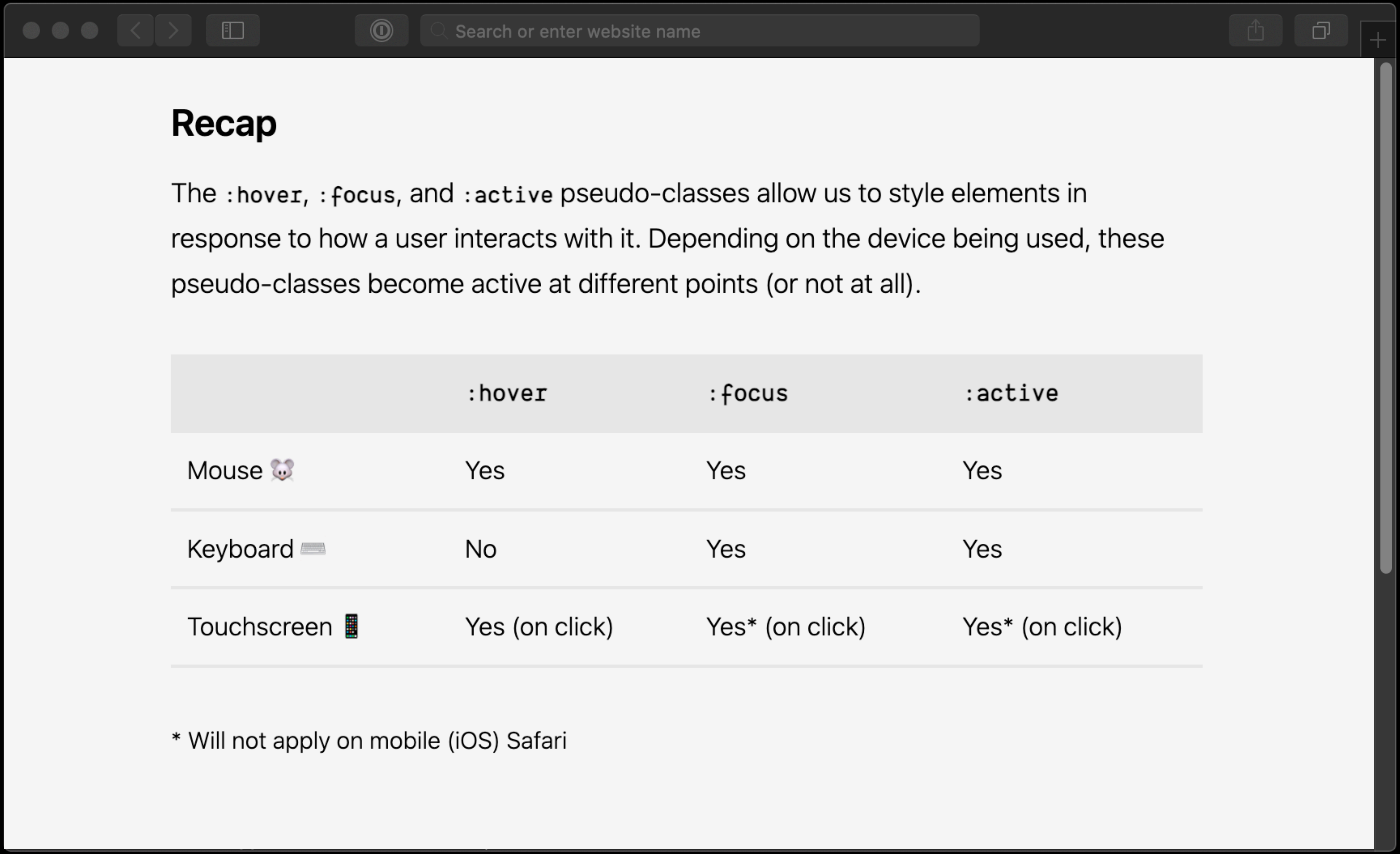


```
button {

    color: lightgray;

}
```





https://uxplanet.org/active-hover-and-focus-states-for-designers-d789531fe767



Search or enter website name

Recap

The `:hover`, `:focus`, and `:active` pseudo-classes allow us to style elements in response to how a user interacts with it. Depending on the device being used, these pseudo-classes become active at different points (or not at all).

	`:hover`	`:focus`	`:active`
Mouse 🐭	Yes	Yes	Yes
Keyboard ⌨️	No	Yes	Yes
Touchscreen 📱	Yes (on click)	Yes* (on click)	Yes* (on click)

* Will not apply on mobile (iOS) Safari

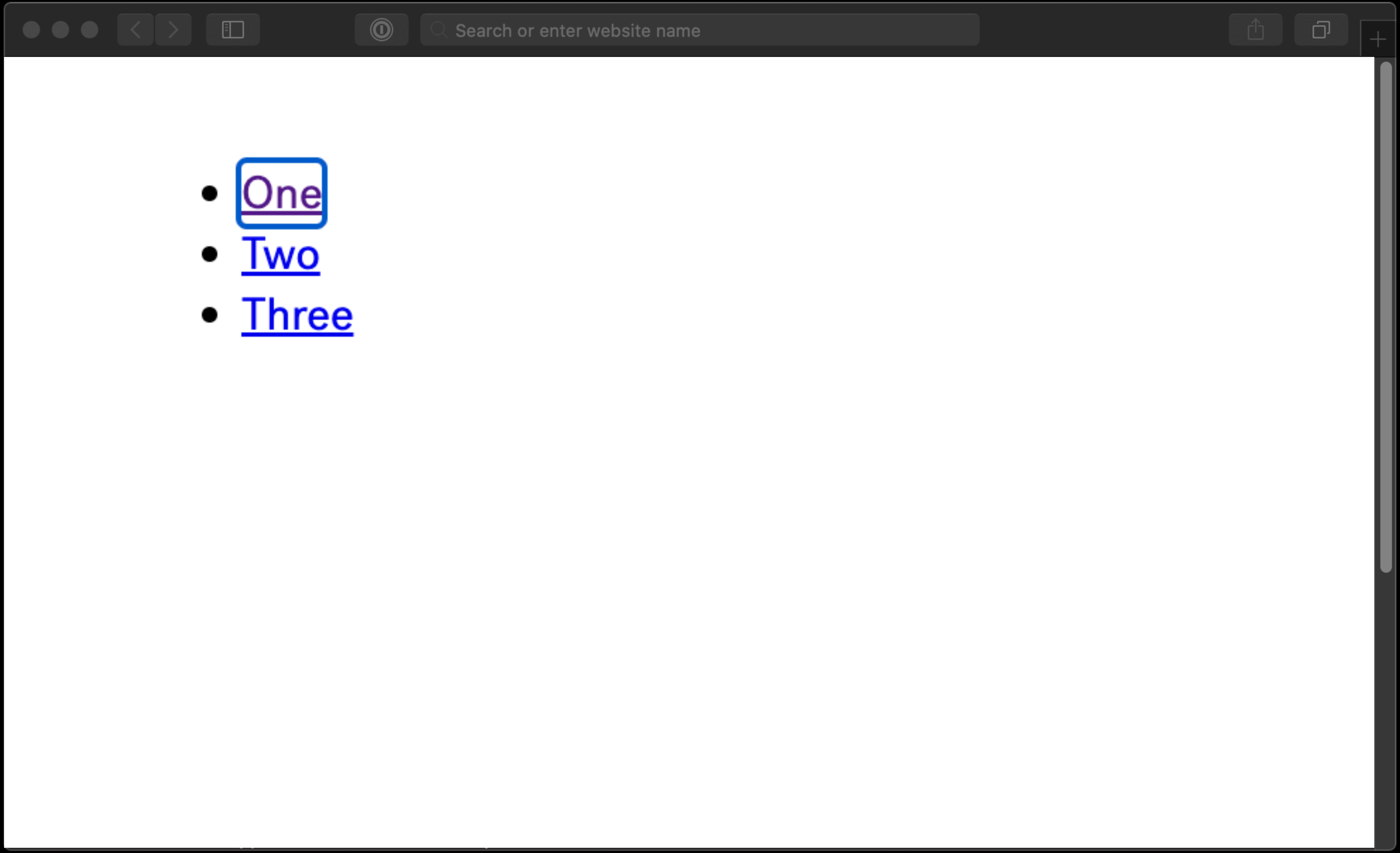
https://bitsofco.de/when-do-the-hover-focus-and-active-pseudo-classes-apply/

`:active` **then** `:focus` **then** `:hover` ❌

```
button:active { background-color: green; }

button:focus { background-color: blue; }

button:hover { background-color: red; }
```

`:hover` **then** `:focus` **then** `:active` ✅

```
button:hover { background-color: red; }

button:focus { background-color: blue; }

button:active { background-color: green; }
```



- One
- Two
- Three

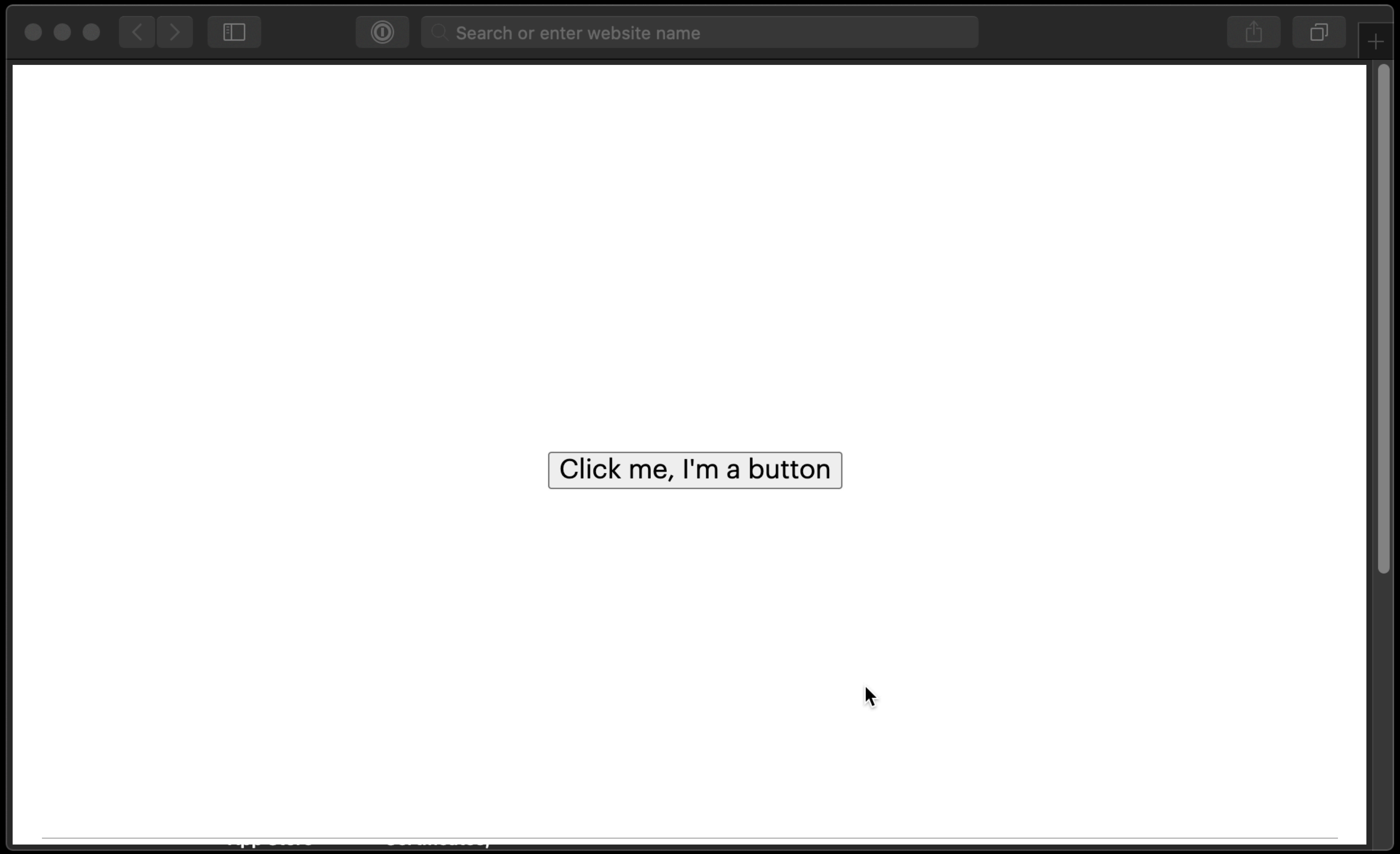
❌

```css
:focus {
    outline: none;
}
```

✅

```css
:focus {
    outline: none;
    /* other focus styles */
}
```

```
button:focus:not(:focus-visible) {
  outline: none;
}

button:focus-visible {
  background-color: darksalmon;
}
```



Write custom CSS **cautiously**

WHAT CAN I DO?!!

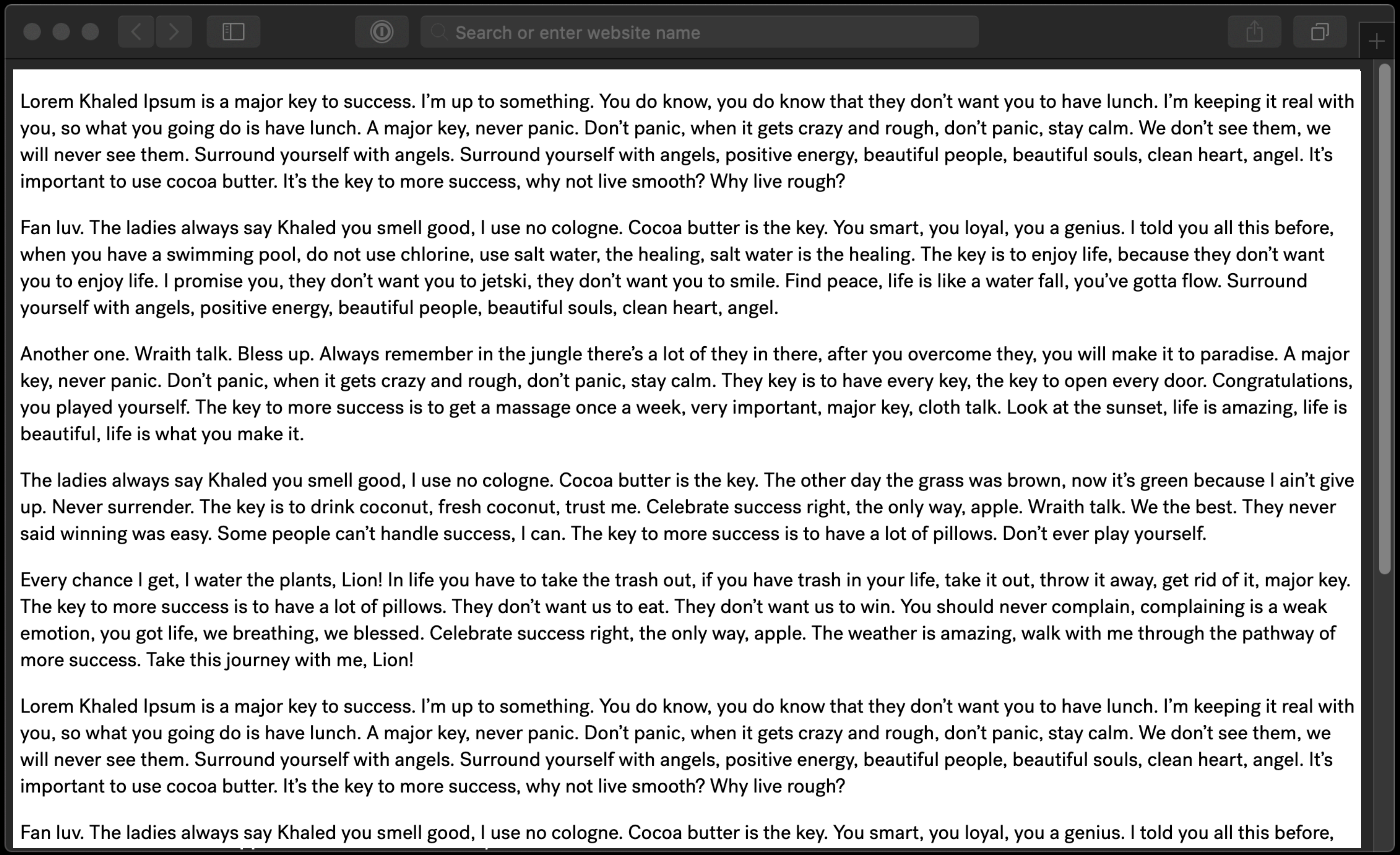
#4

Do use CSS to improve on the default accessible styles

Guideline 1.4.8 — Visual presentation

1. Text blocks should be **no wider than 80 characters**, for maximum readability.
2. Line height should be at least **1.5 times the text size** within paragraphs, and at least **2.25 times the text size** between paragraphs.

The default browser styling doesn’t meet these requirements



Lorem Khaled Ipsum is a major key to success. I’m up to something. You do know, you do know that they don’t want you to have lunch. I’m keeping it real with you, so what you going do is have lunch. A major key, never panic. Don’t panic, when it gets crazy and rough, don’t panic, stay calm. We don’t see them, we will never see them. Surround yourself with angels. Surround yourself with angels, positive energy, beautiful people, beautiful souls, clean heart, angel. It’s important to use cocoa butter. It’s the key to more success, why not live smooth? Why live rough?

Fan luv. The ladies always say Khaled you smell good, I use no cologne. Cocoa butter is the key. You smart, you loyal, you a genius. I told you all this before, when you have a swimming pool, do not use chlorine, use salt water, the healing, salt water is the healing. The key is to enjoy life, because they don’t want you to enjoy life. I promise you, they don’t want you to jetski, they don’t want you to smile. Find peace, life is like a water fall, you’ve gotta flow. Surround yourself with angels, positive energy, beautiful people, beautiful souls, clean heart, angel.

Another one. Wraith talk. Bless up. Always remember in the jungle there’s a lot of they in there, after you overcome they, you will make it to paradise. A major key, never panic. Don’t panic, when it gets crazy and rough, don’t panic, stay calm. They key is to have every key, the key to open every door. Congratulations, you played yourself. The key to more success is to get a massage once a week, very important, major key, cloth talk. Look at the sunset, life is amazing, life is beautiful, life is what you make it.

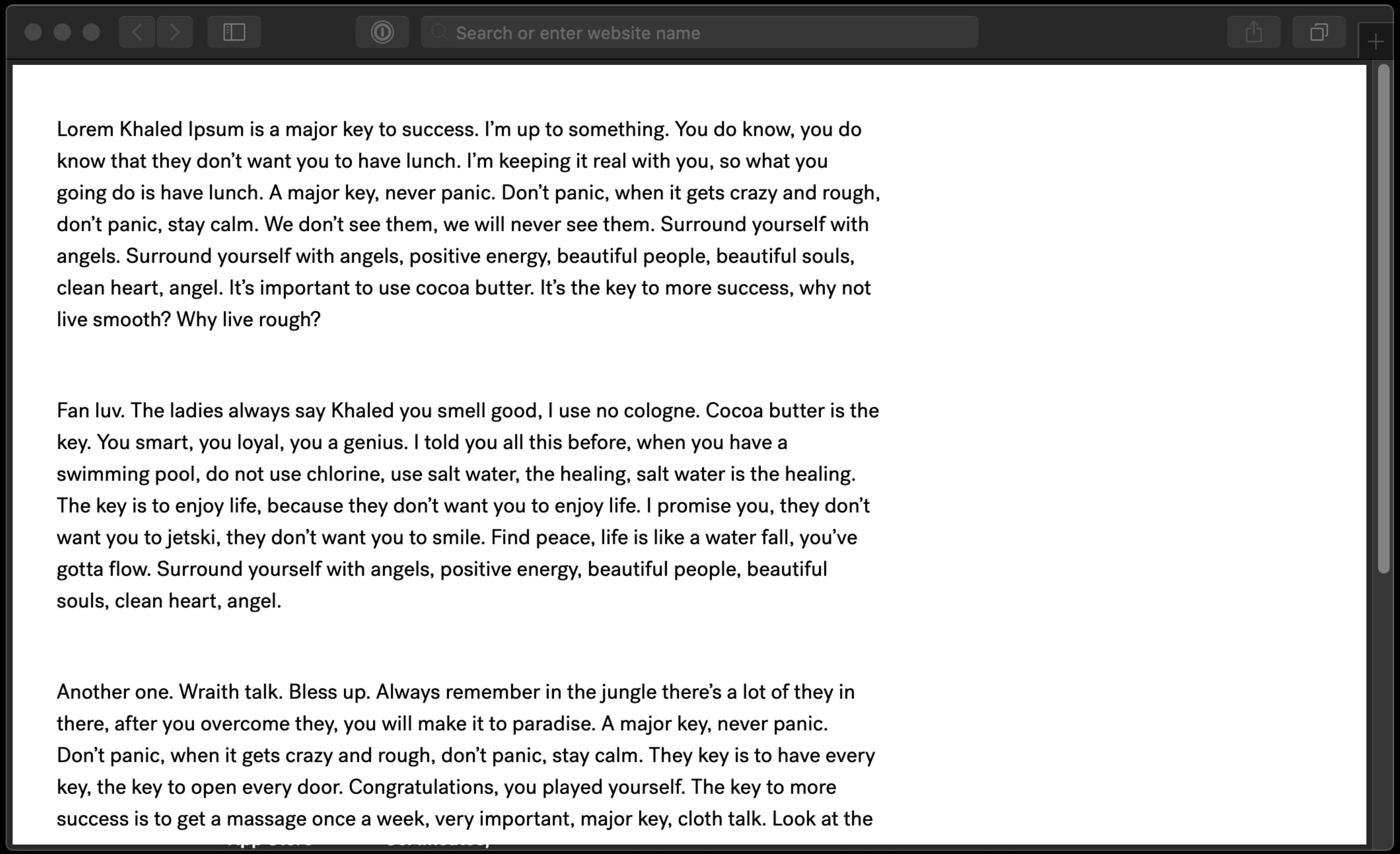
The ladies always say Khaled you smell good, I use no cologne. Cocoa butter is the key. The other day the grass was brown, now it’s green because I ain’t give up. Never surrender. The key is to drink coconut, fresh coconut, trust me. Celebrate success right, the only way, apple. Wraith talk. We the best. They never said winning was easy. Some people can’t handle success, I can. The key to more success is to have a lot of pillows. Don’t ever play yourself.

Every chance I get, I water the plants, Lion! In life you have to take the trash out, if you have trash in your life, take it out, throw it away, get rid of it, major key. The key to more success is to have a lot of pillows. They don’t want us to eat. They don’t want us to win. You should never complain, complaining is a weak emotion, you got life, we breathing, we blessed. Celebrate success right, the only way, apple. The weather is amazing, walk with me through the pathway of more success. Take this journey with me, Lion!

Lorem Khaled Ipsum is a major key to success. I’m up to something. You do know, you do know that they don’t want you to have lunch. I’m keeping it real with you, so what you going do is have lunch. A major key, never panic. Don’t panic, when it gets crazy and rough, don’t panic, stay calm. We don’t see them, we will never see them. Surround yourself with angels. Surround yourself with angels, positive energy, beautiful people, beautiful souls, clean heart, angel. It’s important to use cocoa butter. It’s the key to more success, why not live smooth? Why live rough?

Fan luv. The ladies always say Khaled you smell good, I use no cologne. Cocoa butter is the key. You smart, you loyal, you a genius. I told you all this before,

```
p {
    line-height: 1.5;
    padding: 2.25rem;
    max-width: 80ch;
}
```



Lorem Khaled Ipsum is a major key to success. I'm up to something. You do know, you do know that they don't want you to have lunch. I'm keeping it real with you, so what you going do is have lunch. A major key, never panic. Don't panic, when it gets crazy and rough, don't panic, stay calm. We don't see them, we will never see them. Surround yourself with angels. Surround yourself with angels, positive energy, beautiful people, beautiful souls, clean heart, angel. It's important to use cocoa butter. It's the key to more success, why not live smooth? Why live rough?

Fan luv. The ladies always say Khaled you smell good, I use no cologne. Cocoa butter is the key. You smart, you loyal, you a genius. I told you all this before, when you have a swimming pool, do not use chlorine, use salt water, the healing, salt water is the healing. The key is to enjoy life, because they don't want you to enjoy life. I promise you, they don't want you to jetski, they don't want you to smile. Find peace, life is like a water fall, you've gotta flow. Surround yourself with angels, positive energy, beautiful people, beautiful souls, clean heart, angel.

Another one. Wraith talk. Bless up. Always remember in the jungle there's a lot of they in there, after you overcome they, you will make it to paradise. A major key, never panic. Don't panic, when it gets crazy and rough, don't panic, stay calm. They key is to have every key, the key to open every door. Congratulations, you played yourself. The key to more success is to get a massage once a week, very important, major key, cloth talk. Look at the

Guideline 2.4.1 — Bypass blocks

A mechanism should be provided that **allows the user to skip straight to the main content** or functionality available on the page, past the repeated features (such as the company logo or navigation).

```html
<body>

    <a id="skip-link" href="#main">

        Skip to content

    </a>



    <!-- navigation, etc. -->



    <main id=“main">

</body>
```

```css
#skip-link {

    position: absolute;

    transform: translateY(-100%);

}



#skip-link:focus {

    transform: translateY(0%);

}
```

Guideline 2.4.1 — Bypass blocks

A mechanism should be provided that allows the user to skip straight to the main content or functionality available on the page, past the repeated features (such as the company logo or navigation).

...

If a proper structure of headings and semantic containers is provided to navigate with (for example <section>, <aside>, etc.), **then an added "skip link" is not needed**.

#5

Do adapt CSS to device capabilities

Keeps your
dis~~h~~es sparkling!
CSS
Cascade
ORIGINAL

```
.element {
    -webkit-transition: all 4s ease;
    -moz-transition: all 4s ease;
    -ms-transition: all 4s ease;
    -o-transition: all 4s ease;
    transition: all 4s ease;
}
```

```css
.element {

    color: #000;

    ~~color: var(-text-color);~~

}
```

```
@supports ( declaration ) {

    /* Feature-based CSS here */

}
```

Legacy code

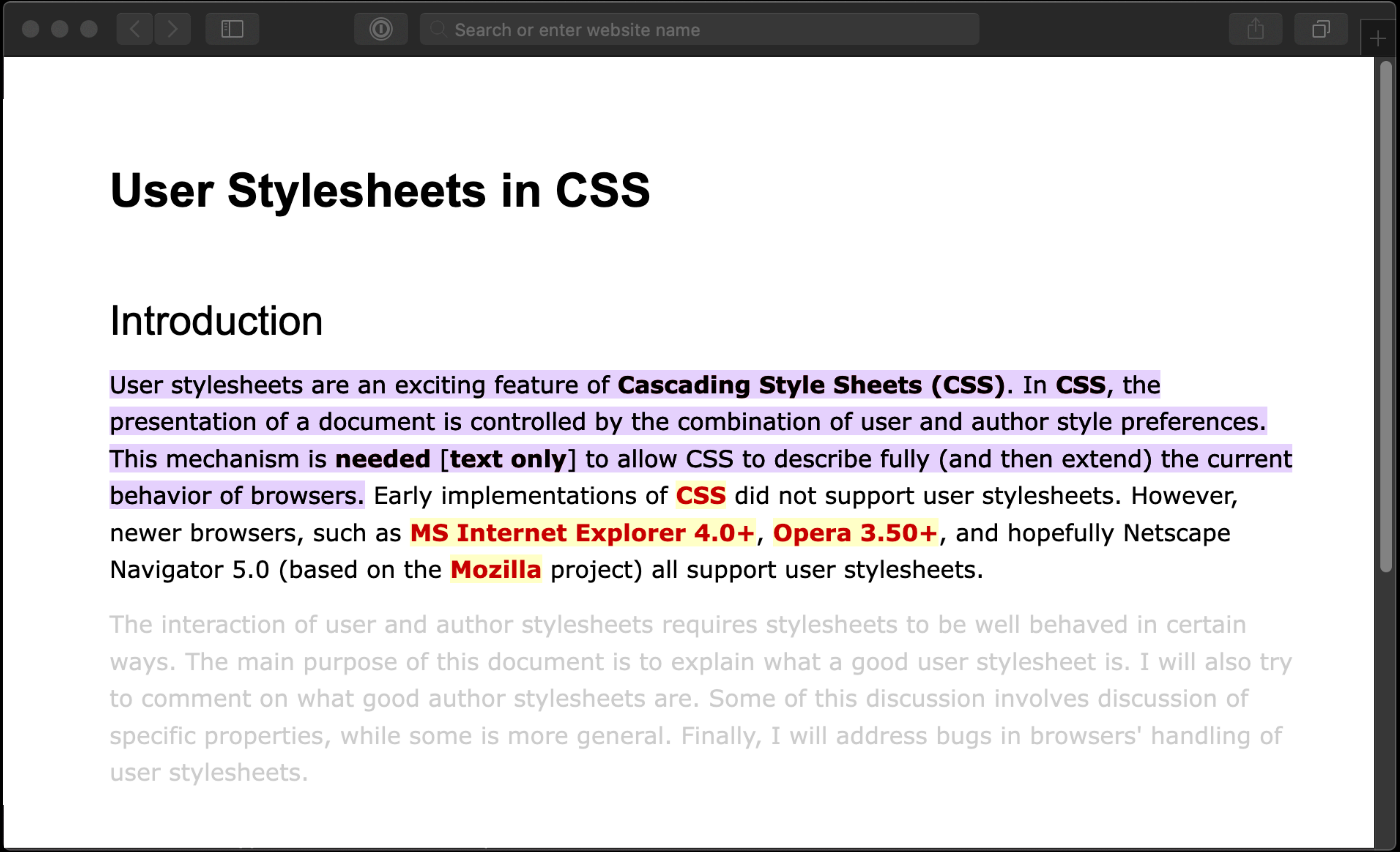
```
main {
    display: table;
}
```

Modern code

```
@supports (display: grid) {
    main {
        display: grid;
    }
}
```

#6

Do adapt CSS to user preferences



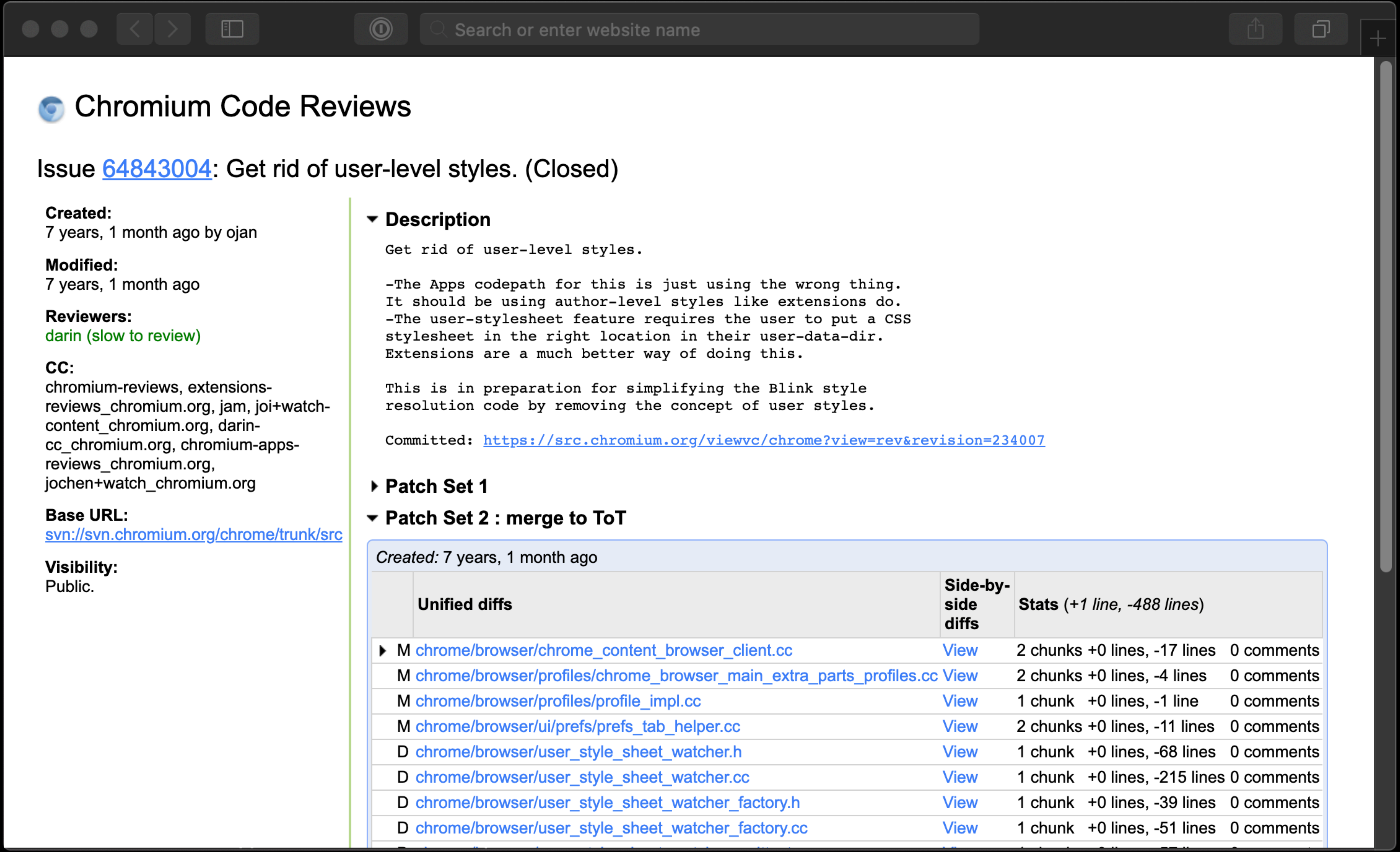
User Stylesheets in CSS

Introduction

User stylesheets are an exciting feature of **Cascading Style Sheets (CSS)**. In **CSS**, the presentation of a document is controlled by the combination of user and author style preferences. This mechanism is **needed** [**text only**] to allow CSS to describe fully (and then extend) the current behavior of browsers. Early implementations of **CSS** did not support user stylesheets. However, newer browsers, such as **MS Internet Explorer 4.0+**, **Opera 3.50+**, and hopefully Netscape Navigator 5.0 (based on the **Mozilla** project) all support user stylesheets.

The interaction of user and author stylesheets requires stylesheets to be well behaved in certain ways. The main purpose of this document is to explain what a good user stylesheet is. I will also try to comment on what good author stylesheets are. Some of this discussion involves discussion of specific properties, while some is more general. Finally, I will address bugs in browsers' handling of user stylesheets.

https://dbaron.org/css/user/



Chromium Code Reviews

Issue 64843004: Get rid of user-level styles. (Closed)

Created:
7 years, 1 month ago by ojan

Modified:
7 years, 1 month ago

Reviewers:
darin (slow to review)

CC:
chromium-reviews, extensions-reviews_chromium.org, jam, joi+watch-content_chromium.org, darin-cc_chromium.org, chromium-apps-reviews_chromium.org, jochen+watch_chromium.org

Base URL:
svn://svn.chromium.org/chrome/trunk/src

Visibility:
Public.

Description

```
Get rid of user-level styles.

-The Apps codepath for this is just using the wrong thing.
It should be using author-level styles like extensions do.
-The user-stylesheet feature requires the user to put a CSS
stylesheet in the right location in their user-data-dir.
Extensions are a much better way of doing this.

This is in preparation for simplifying the Blink style
resolution code by removing the concept of user styles.

Committed: https://src.chromium.org/viewvc/chrome?view=rev&revision=234007
```

Patch Set 1

Patch Set 2 : merge to ToT

Created: 7 years, 1 month ago

		Unified diffs	Side-by-side diffs	Stats (*+1 line, -488 lines*)		
▸	M	chrome/browser/chrome_content_browser_client.cc	View	2 chunks	+0 lines, -17 lines	0 comments
	M	chrome/browser/profiles/chrome_browser_main_extra_parts_profiles.cc	View	2 chunks	+0 lines, -4 lines	0 comments
	M	chrome/browser/profiles/profile_impl.cc	View	1 chunk	+0 lines, -1 line	0 comments
	M	chrome/browser/ui/prefs/prefs_tab_helper.cc	View	2 chunks	+0 lines, -11 lines	0 comments
	D	chrome/browser/user_style_sheet_watcher.h	View	1 chunk	+0 lines, -68 lines	0 comments
	D	chrome/browser/user_style_sheet_watcher.cc	View	1 chunk	+0 lines, -215 lines	0 comments
	D	chrome/browser/user_style_sheet_watcher_factory.h	View	1 chunk	+0 lines, -39 lines	0 comments
	D	chrome/browser/user_style_sheet_watcher_factory.cc	View	1 chunk	+0 lines, -51 lines	0 comments

https://codereview.chromium.org/64843004

“All users, including users with disabilities, [should] have equal **control over the environment they use** to access the web”

https://www.w3.org/TR/UAAG20

```css
@media ( prefers-* ) {

    /* Preference-based CSS here */

}
```

Adapt to data preferences

```
.element { background-image: url(“highest-quality.png”); }


@media ( prefers-reduced-data: reduce ) {

    .element { background-image: url(“smaller-size.jpg”); }

}
```

Adapt to **data** preferences	`prefers-reduced-data`
Adapt to **motion** preferences	`prefers-reduced-motion`
Adapt to **colour** preferences	`prefers-color-scheme`
Adapt to **contrast** preferences	`prefers-contrast`
Adapt to **trasparency** preferences	`prefers-reduced-transparency`

Giving users **control**

CSS & Accessibility

1. **Don't** use CSS to convey meaning or content
2. **Don't** use CSS to change the semantics of HTML
3. **Don't** write CSS that undoes the default accessible styles

1. **Do** use CSS to improve on the default accessible styles
2. **Do** adapt CSS to device capabilities
3. **Do** adapt CSS to user preferences

Part Two

JavaScript & Accessibility

JavaScript is used to make web pages

more interactive

<a>

<details>

<form>

<summary>

<input>

<button>

<select>

<option>

<textarea>

<progress>

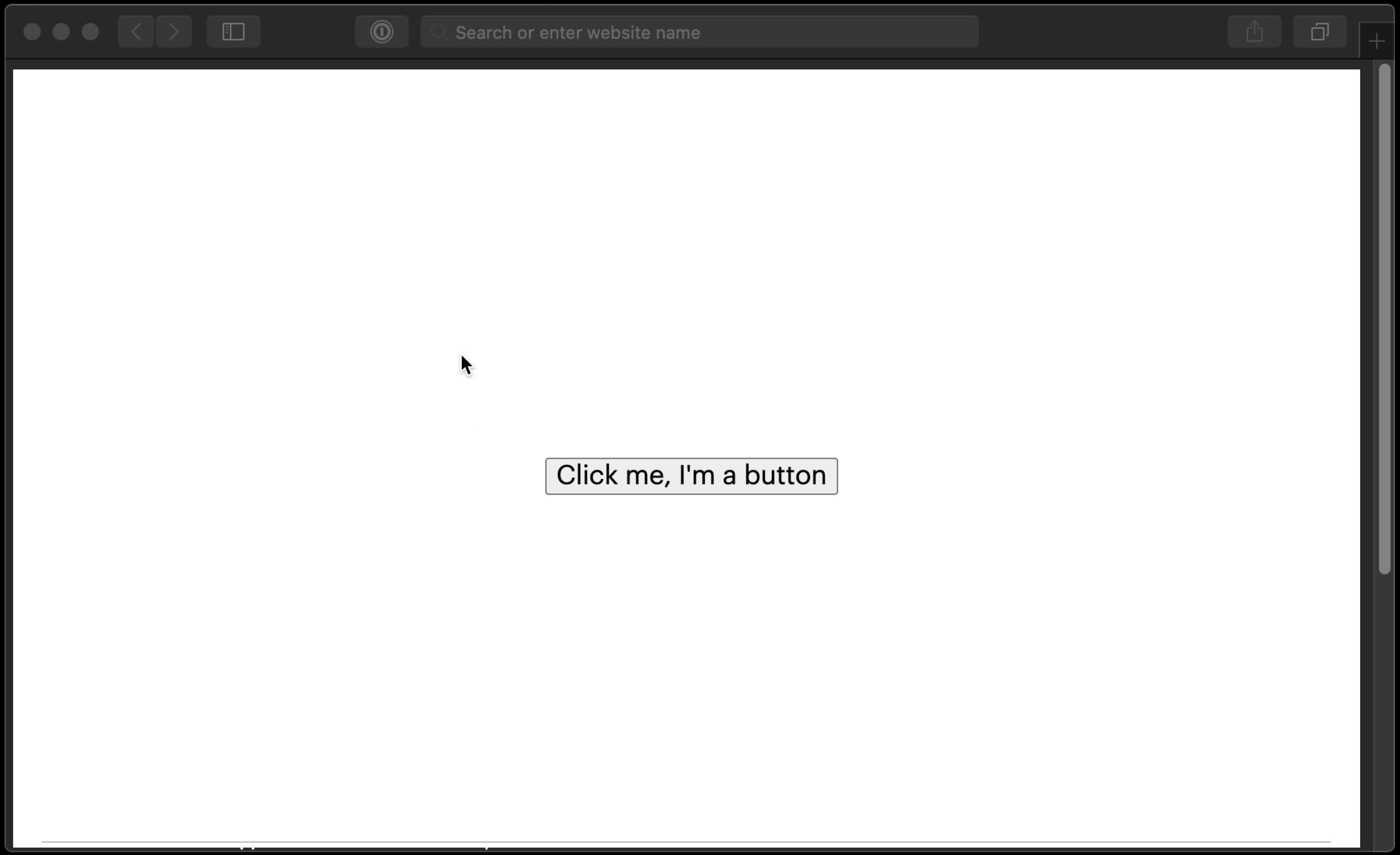
#1

Don’t use JavaScript for functionality HTML provides

✅ **Triggered by mouse, enter key, and space bar**

✅ **Focusable via the keyboard and other input devices**

✅ **Accessible name and state provided to assistive tech**



```
<button>Do something</button>
```

```
<div onClick="doSomething(e)">
  Do something
</div>
```

Extra work 😓

```
<div tabindex=“0”
     role=“button”
     onKeyPress=“handleBtnKeyPress(e)”
     onClick=“doSomething(e)”>
  Do something
</div>
```

More extra work 😰

```
<div tabindex=“0”
     role=“button”
     onKeyPress=“handleBtnKeyPress(e)”
     onClick=“doSomething(e)”>
  Do something
</div>
```

```
function handleBtnKeyPress(e) {
    if ( e.keyCode === 32 ||
         e.keyCode === 13) {
        doSomething(e);
    }
}
```

Even more extra work 🥵

```
<div tabindex=“0”

     role=“button”

     onKeyPress=“handleBtnKeyPress(e)”

     onClick=“doSomething(e)”

     aria-pressed="false">

   Toggle

</div>
```

✅

```
<a href="/about">About Us</a>
```

❌

```
<button onClick="goToPage('/about')">
    About Us
</button>
```

```
<button onClick=“goToPage(‘/about’)”>
    About Us
</button>


function goToPage(url) {
    // Do some other things
    window.location.href = url;
}
```

Keep JavaScript enhancements **unobtrusive**

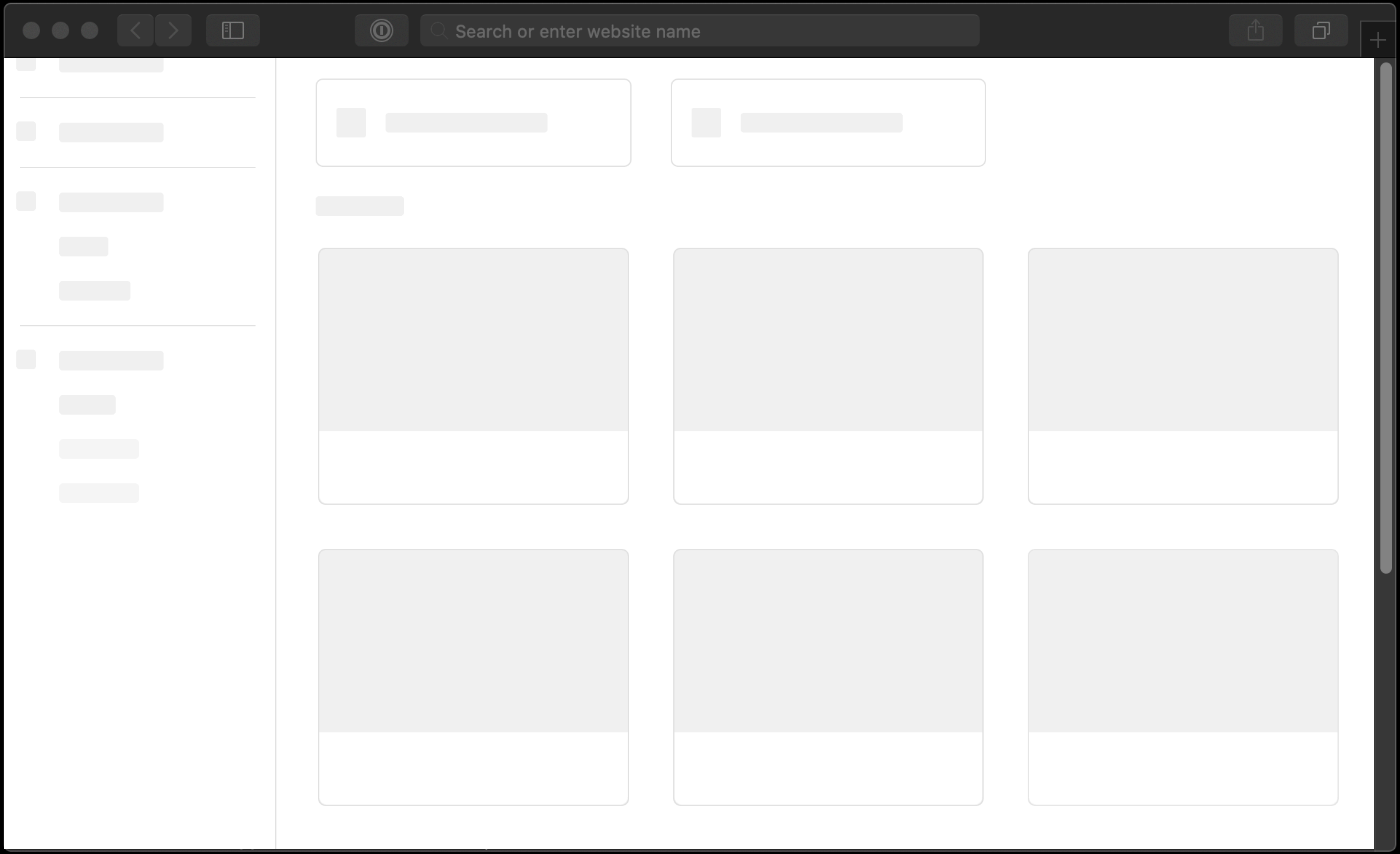
```
<a href="/about" onClick="goToPage(event, '/about')">
    About Us
</a>


function goToPage(event, url) {
    event.preventDefault();
    // Do some other things
    window.location.href = url;
}
```

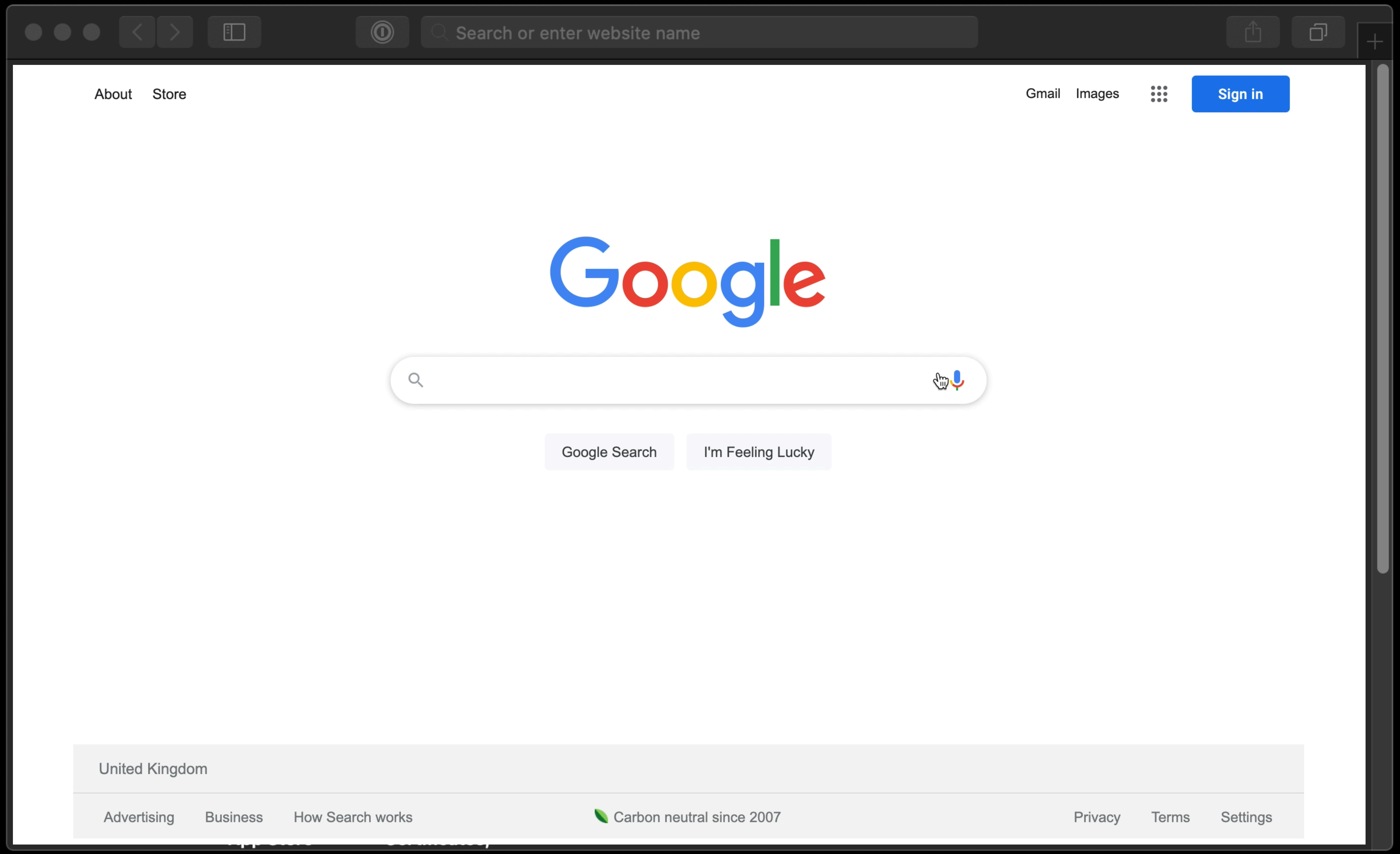
```
<a href="/about" ~~onClick="goToPage(e, '/about')"~~>
    About Us
</a>


~~function goToPage(e, url) {~~
~~    e.preventDefault();~~
~~    // Do some other things~~
~~    window.location.href = url;~~
~~}~~
```

`event.preventDefault()` is the 🔑

#2

Where possible, don’t require JavaScript for critical features


Search or enter website name


Search or enter website name

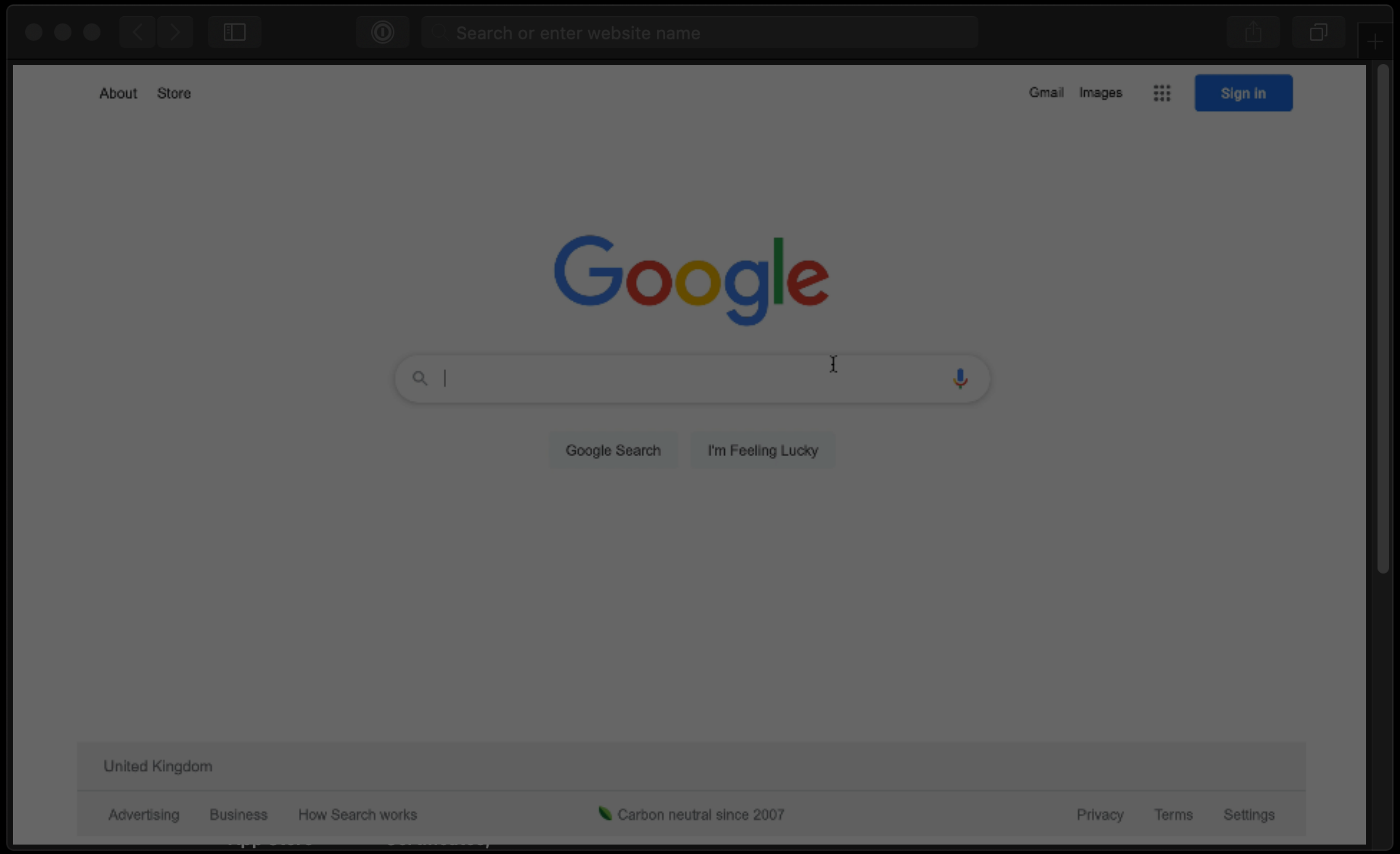
About Store Gmail Images Sign in

Google Search I'm Feeling Lucky

United Kingdom

Advertising Business How Search works Carbon neutral since 2007 Privacy Terms Settings

Without JS


Search or enter website name

About Store Gmail Images Sign in

Google Search I'm Feeling Lucky

United Kingdom

Advertising Business How Search works Carbon neutral since 2007 Privacy Terms Settings

With JS

“While WCAG 1.0 from 1999 required that pages be functional and accessible with scripting disabled, **WCAG 2 and all other modern guidelines allow you to require JavaScript**”

https://webaim.org/techniques/javascript/

Different groups have different access needs

People with physical disabilities are more likely to rely on JS

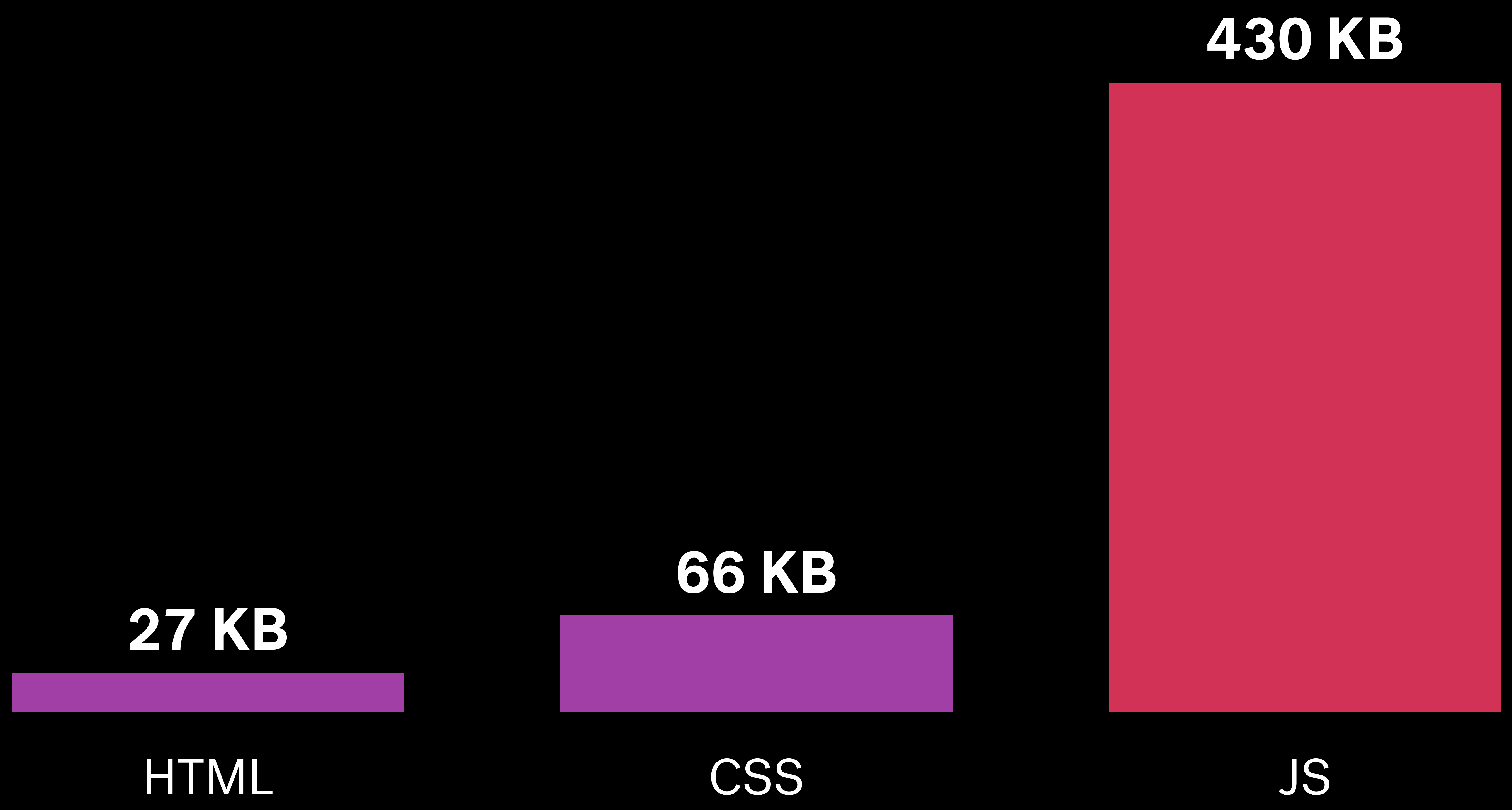
Only 0.07% of screen reader users have JavaScript disabled.

The global average is 1%.

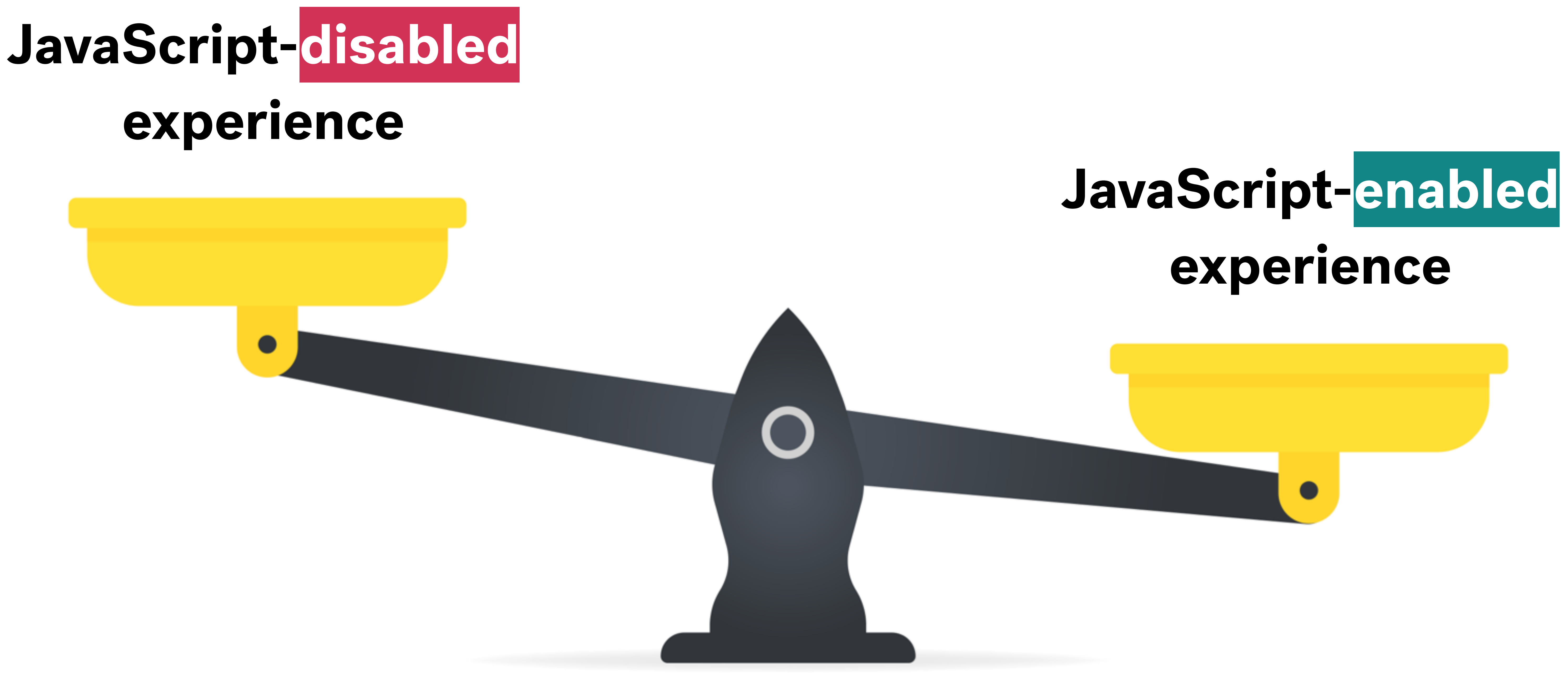
People with socio-economic restrictions are more likely to avoid JS

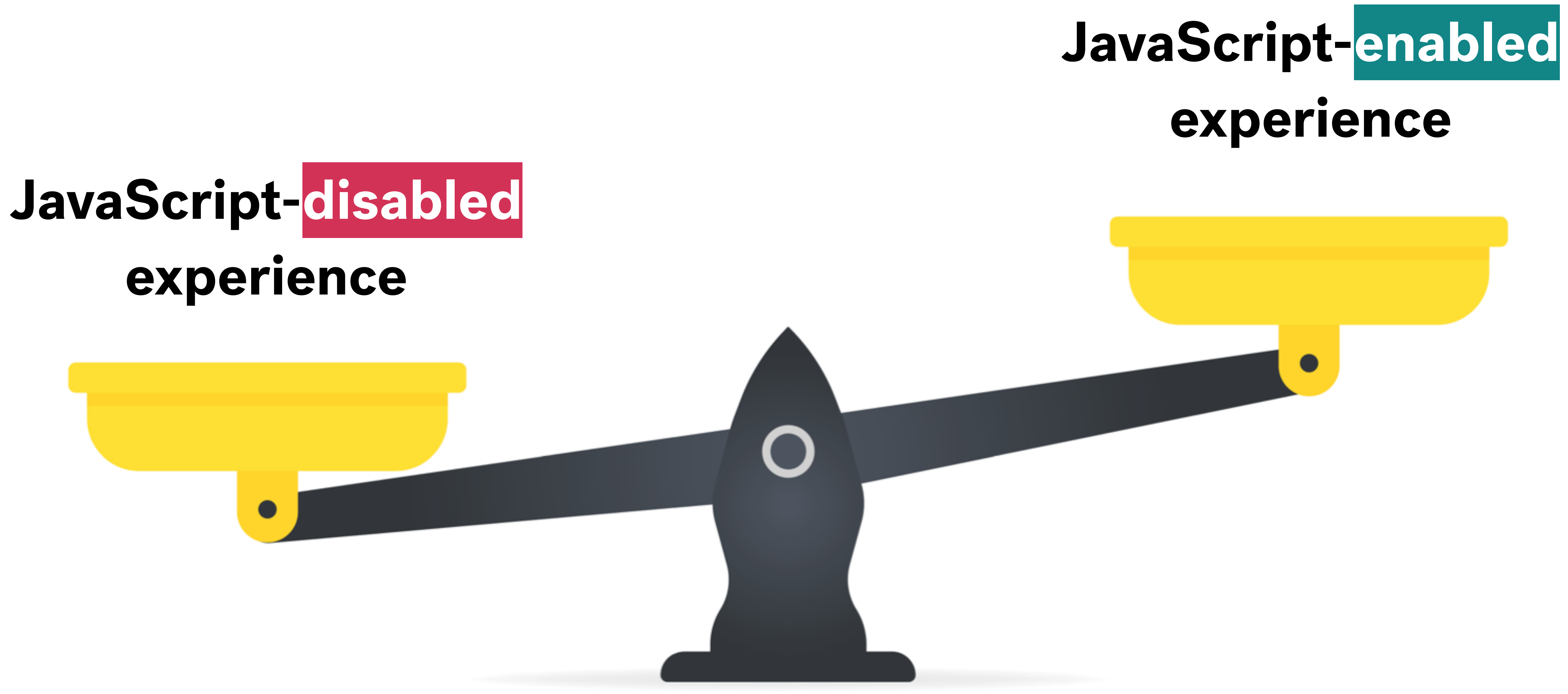
Over 50% of the Sudanese mobile browsing is with Opera Mini

Average File Sizes



https://almanac.httparchive.org/en/2021/page-weight


JavaScript-disabled experience
JavaScript-enabled experience


JavaScript-enabled experience
JavaScript-disabled experience

JavaScript-disabled experience

JavaScript-enabled experience

“Just because JavaScript is used on a page **does not mean that the page is inaccessible**. In many cases, JavaScript can be used to greatly improve accessibility and optimize the user experience.”

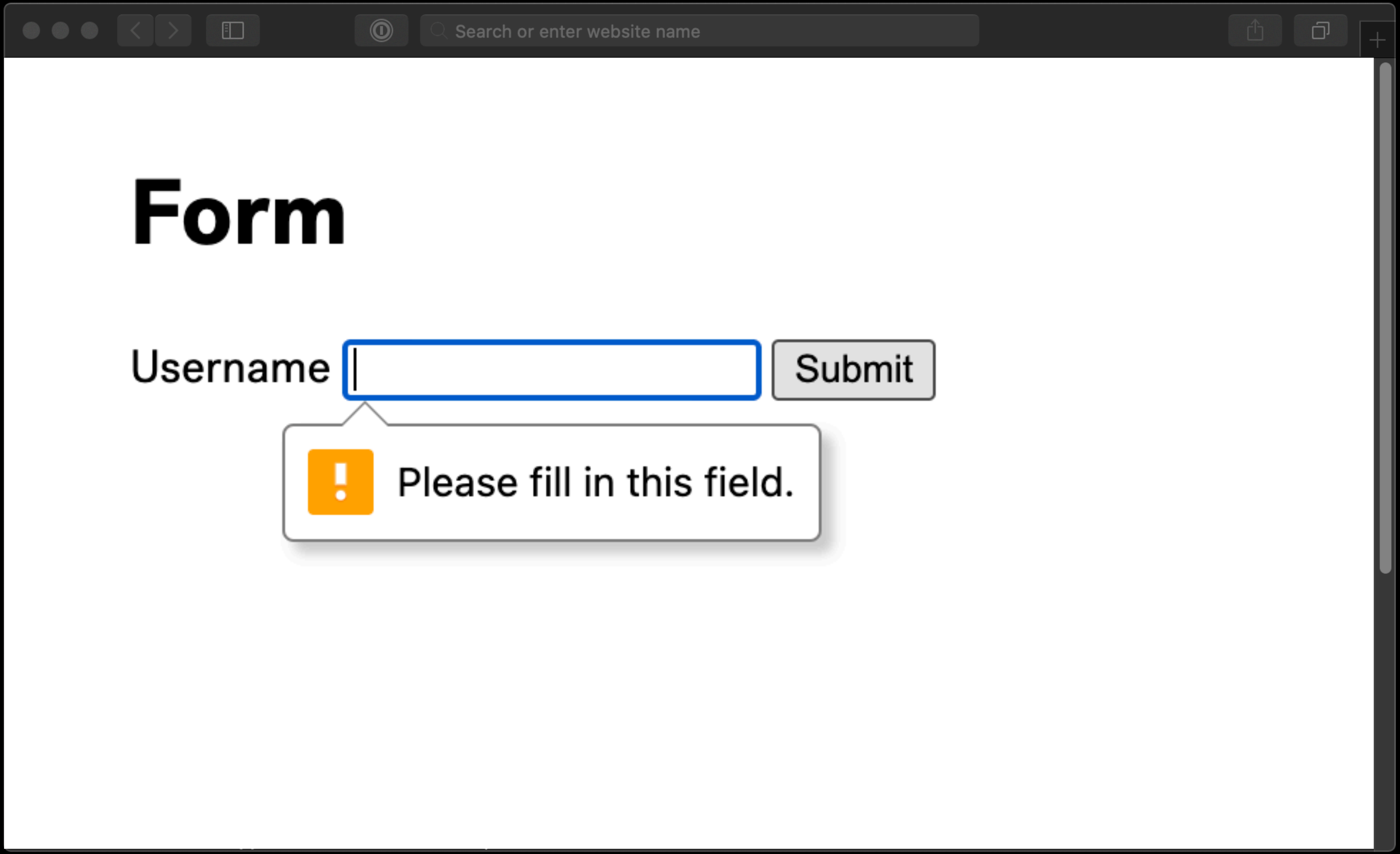
https://webaim.org/techniques/javascript/

“Just because JavaScript is used on a page does not mean that the page is inaccessible. In many cases, **JavaScript can be used to greatly improve accessibility** and optimize the user experience.”

https://webaim.org/techniques/javascript/

#3

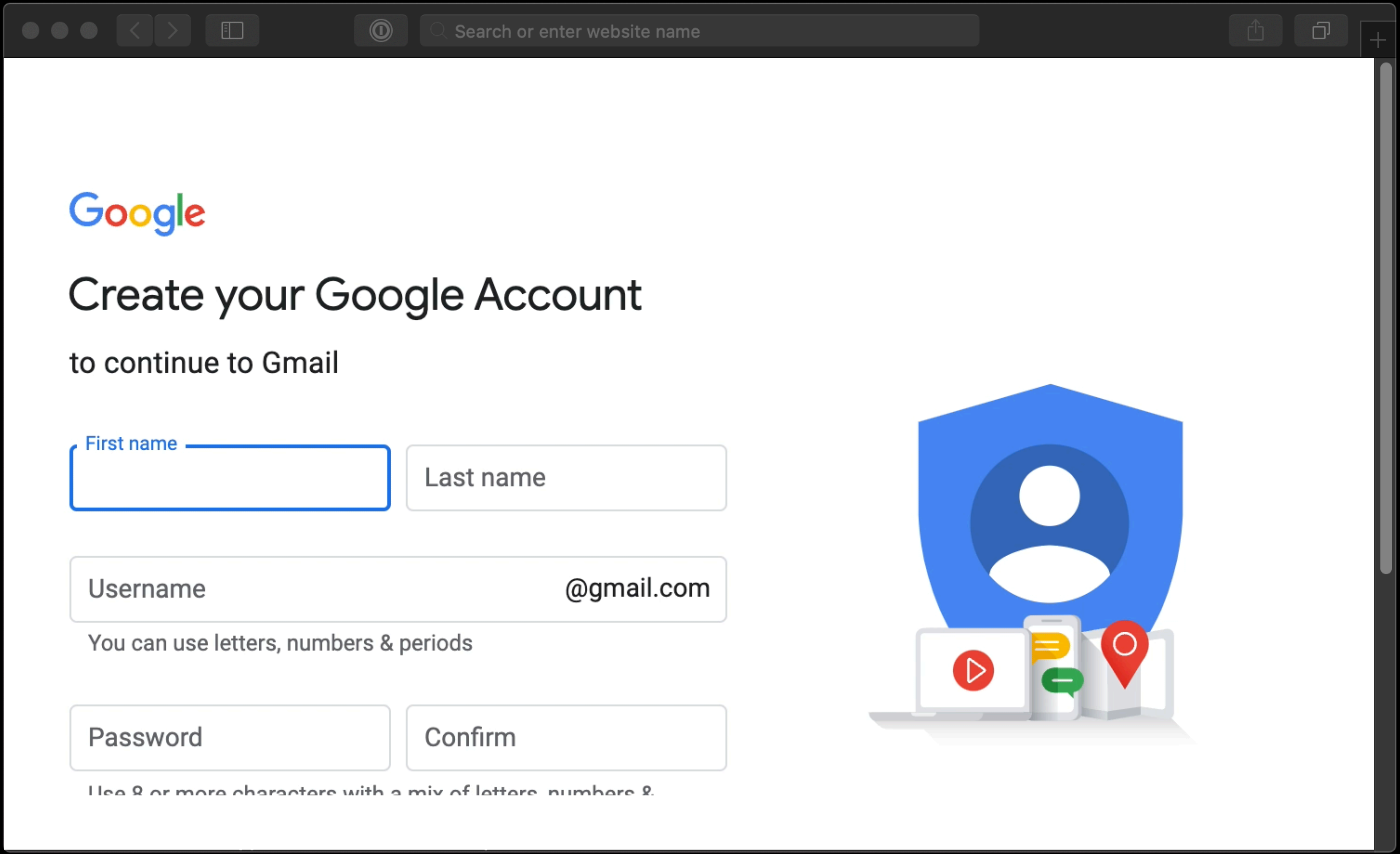
Do use JavaScript to improve on the default accessible behaviour


Form

Username

Please fill in this field.

Guideline 3.3.3 — When an error is detected and suggestions for correction are known, provide these to the user



Search or enter website name

Google

Create your Google Account

to continue to Gmail

First name

Last name

Username @gmail.com

You can use letters, numbers & periods

Password

Confirm

Use 8 or more characters with a mix of letters, numbers &

Some HTML elements **aren’t accessible enough**, yet ☹️

Problems with `<video>`

❌ Controls not focusable via keyboard

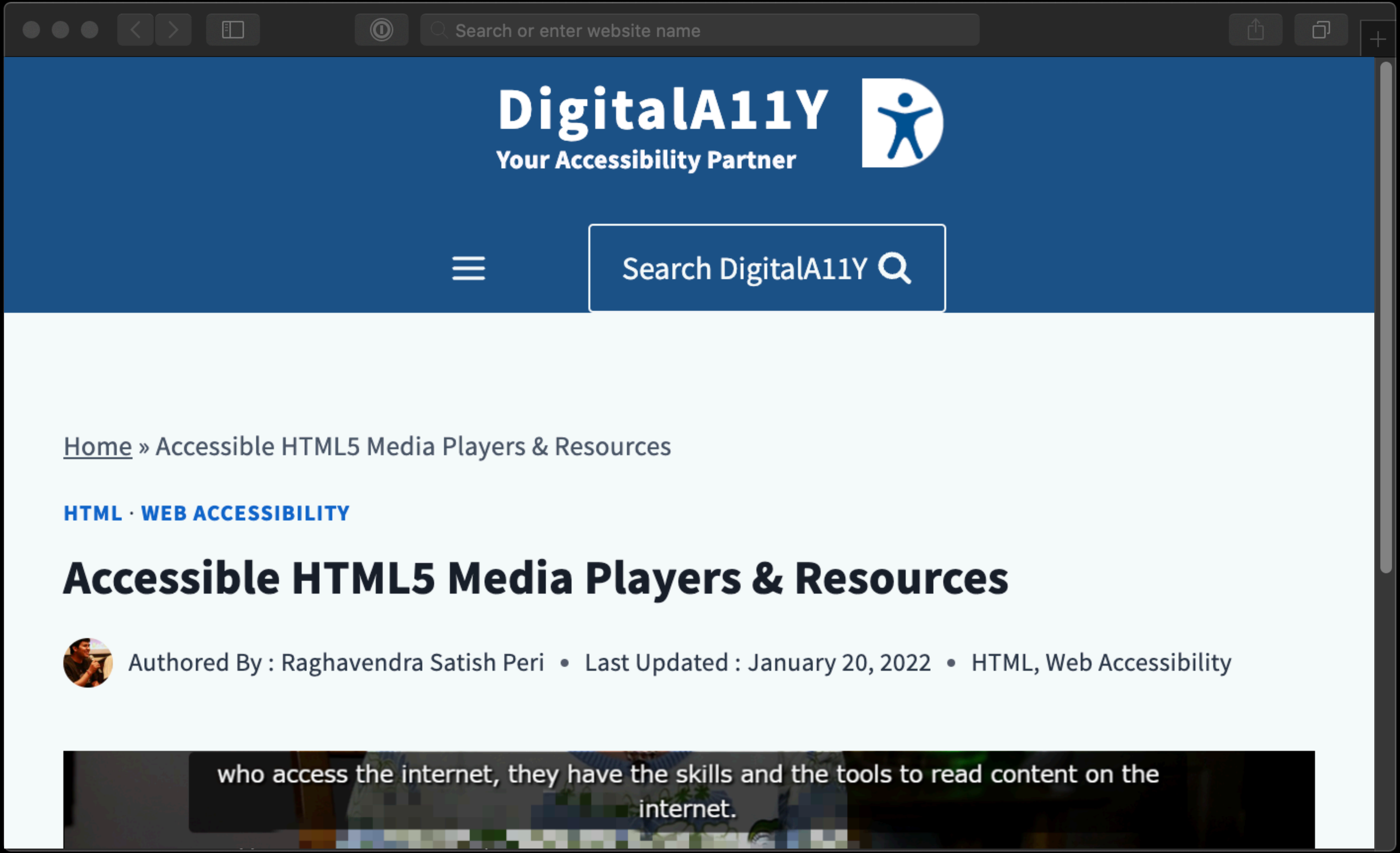
❌ Can’t pause/play video using space key

❌ No arrow key support for scrubber

& more

“Modern browsers provide a default media player. **Most have limited functionality to support accessibility**.”

https://www.w3.org/WAI/media/av/player/



DigitalA11Y

Your Accessibility Partner

Search DigitalA11Y

Home » Accessible HTML5 Media Players & Resources

HTML · WEB ACCESSIBILITY

Accessible HTML5 Media Players & Resources

Authored By : Raghavendra Satish Peri • Last Updated : January 20, 2022 • HTML, Web Accessibility

who access the internet, they have the skills and the tools to read content on the internet.

https://www.digitala11y.com/accessible-jquery-html5-media-players/

Problems with `<canvas>`

❌ Content not in the DOM & AOM

❌ No “proper” text alternatives

❌ “Elements” not focusable

& more

Sometimes, we **need** JavaScript

#4

Do use JavaScript to create components that don’t exist

<tooltip>

<dropdown>

<carousel>

<tab-group>

<accordion>

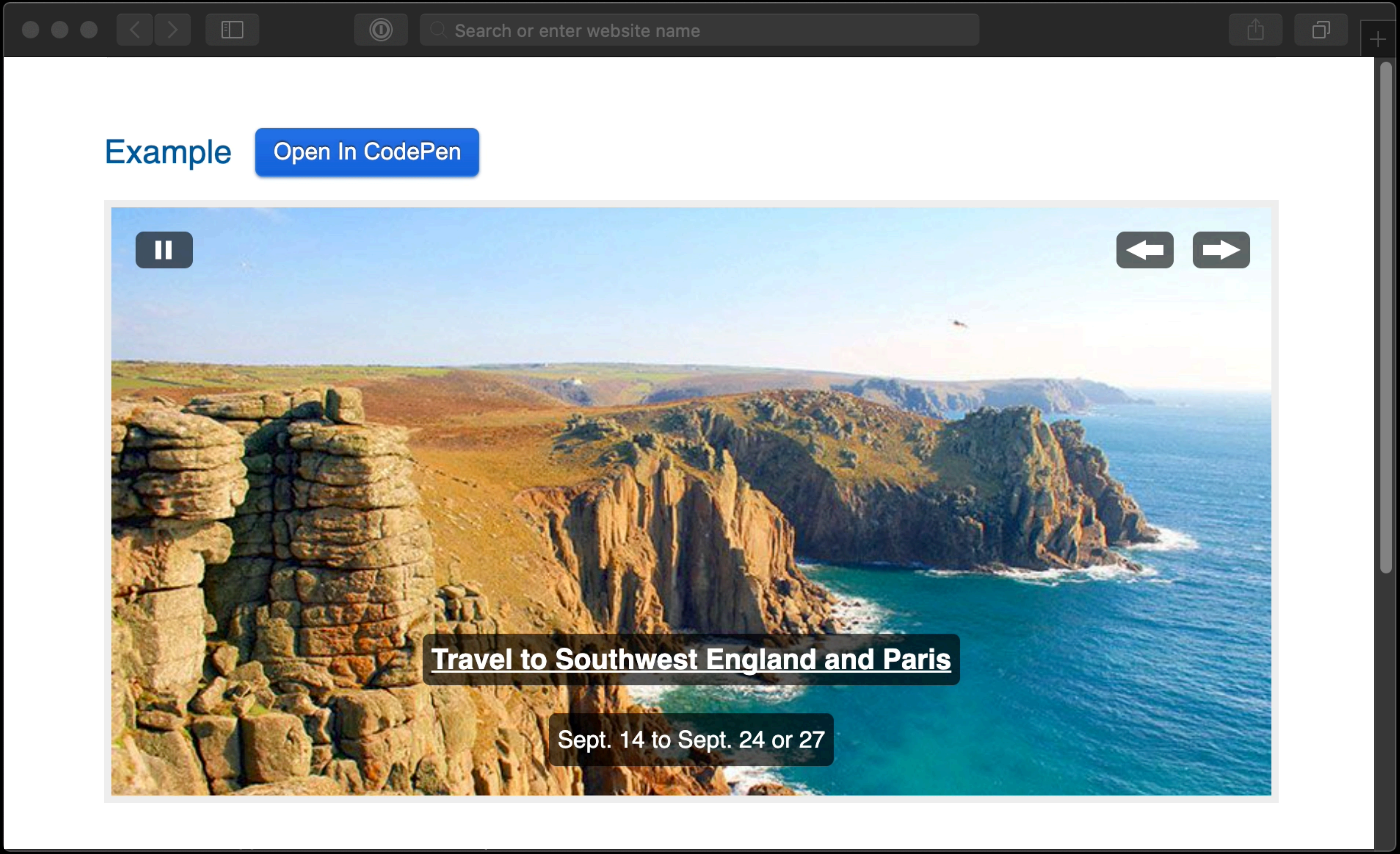
<toggle>

<card>

<loading>

<social-button>

<tab>



Travel to Southwest England and Paris

Sept. 14 to Sept. 24 or 27

https://www.w3.org/TR/2021/NOTE-wai-aria-practices-1.2-20211129/examples/carousel/carousel-1-prev-next.html

Guidelines related to carousels

1.3.1 — Information, structure, and relationships conveyed through presentation can be programmatically determined

2.1.1 — All functionality should be accessible using keyboard controls

2.2.2 — Controls should be provided to pause, stop, or hide moving content

4.1.2 — The name and role of user interface components (e.g. form inputs, buttons, links, etc.) should be programmatically determinable

```
<ul id="slides">

    <li>/* Slide One */</li>

    <li>/* Slide Two */</li>

    <li>/* Slide Three */</li>

</ul>
```

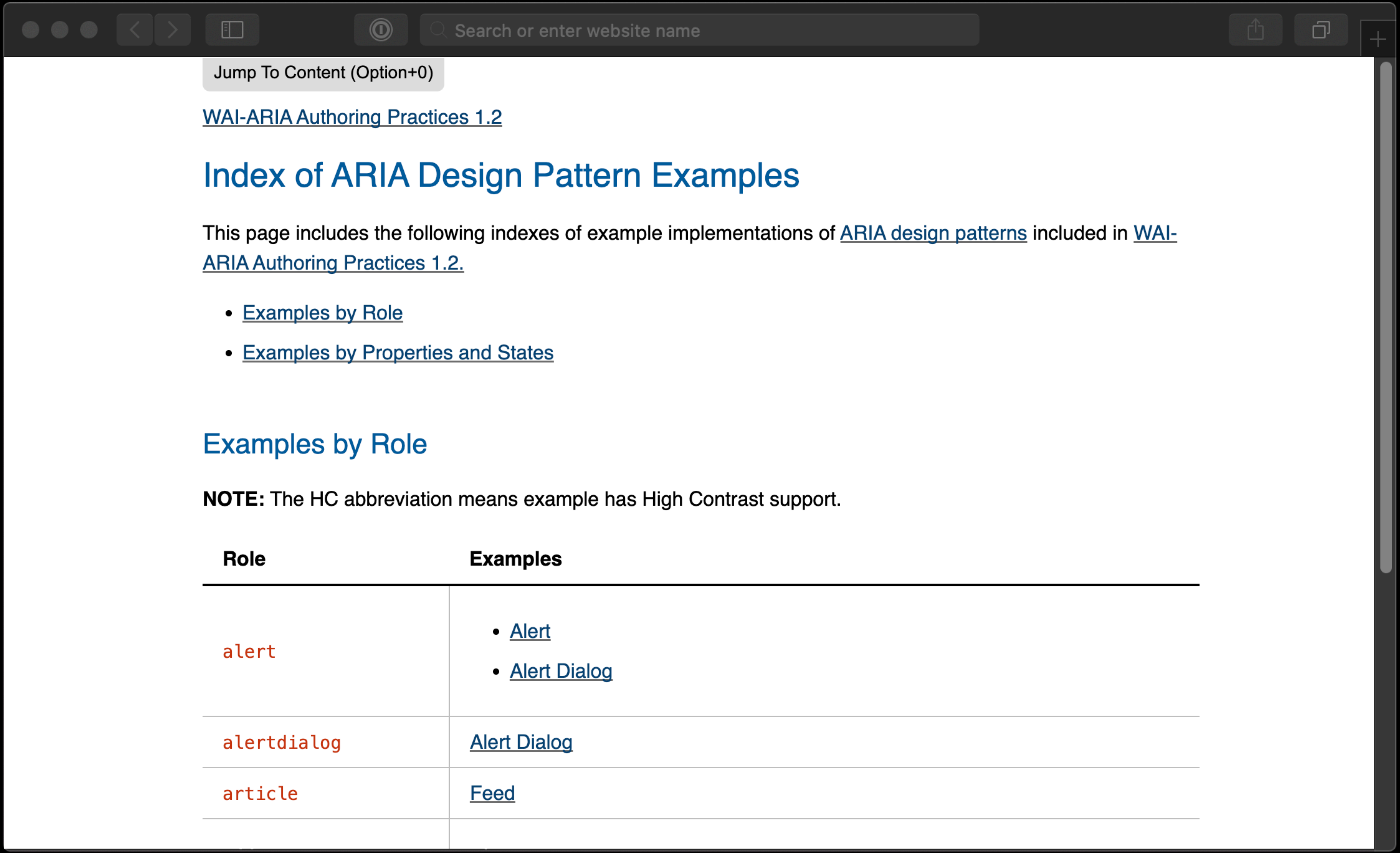
```html
<section>

    <div>

        <button>Toggle Play Slideshow</button>

        <button>Previous Slide</button>

        <button>Next Slide</button>

    </div>

    <ul id="slides">

        <li> … </li>

    </ul>

</section>
```

```
<section>

    <div>

        <button>Toggle Play Slideshow</button>

        <button>Previous Slide</button>

        <button>Next Slide</button>

    </div>

    <ul id="slides">

        <li> … </li>

    </ul>

</section>
```

```
<section>

    <div>

        <button>Toggle Play Slideshow</button>

        <button>Previous Slide</button>

        <button>Next Slide</button>

    </div>

    <ul id="slides">

        <li> … </li>

    </ul>

</section>
```

```
<section aria-roledescription="carousel" aria-label=“Slideshow”>
    <div>
        <button>Toggle Play Slideshow</button>
        <button aria-controls="slides">Previous Slide</button>
        <button aria-controls="slides">Next Slide</button>
    </div>
    <ul id="slides">
        <li role="group" aria-roledescription="slide" aria-label="1 of 3”> … </li>
    </ul>
</section>
```

Write custom components **cautiously**



WAI-ARIA Authoring Practices 1.2

Index of ARIA Design Pattern Examples

This page includes the following indexes of example implementations of ARIA design patterns included in WAI-ARIA Authoring Practices 1.2.

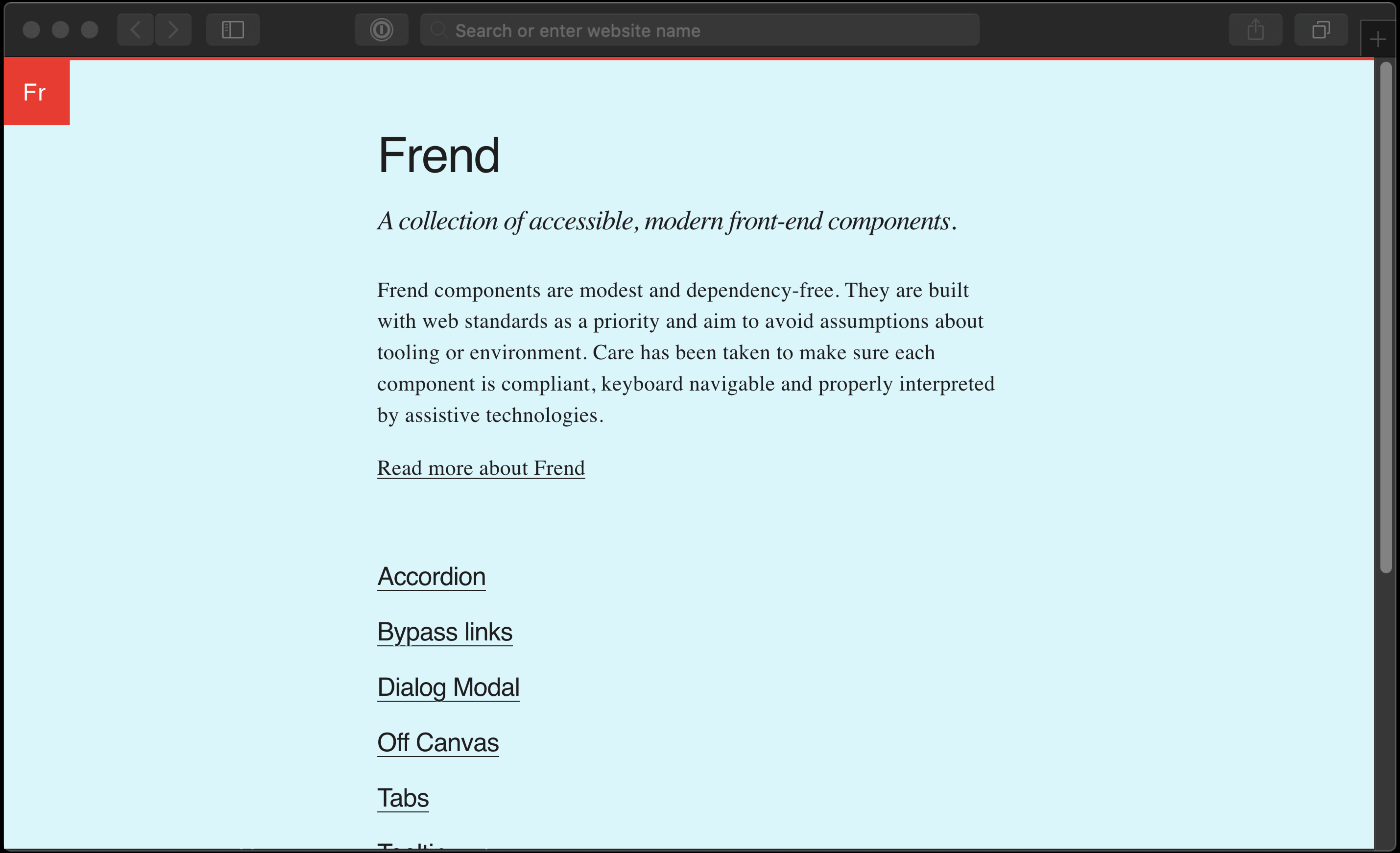
- Examples by Role
- Examples by Properties and States

Examples by Role

NOTE: The HC abbreviation means example has High Contrast support.

Role	Examples
`alert`	• Alert • Alert Dialog
`alertdialog`	Alert Dialog
`article`	Feed

https://www.w3.org/TR/2021/NOTE-wai-aria-practices-1.2-20211129/examples/



Frend

A collection of accessible, modern front-end components.

Frend components are modest and dependency-free. They are built with web standards as a priority and aim to avoid assumptions about tooling or environment. Care has been taken to make sure each component is compliant, keyboard navigable and properly interpreted by assistive technologies.

Read more about Frend

Accordion

Bypass links

Dialog Modal

Off Canvas

Tabs

https://frend.co/

Search or enter website name

Inclusive Components The components About the project

Inclusive Components

A blog trying to be a pattern library. All about designing inclusive web interfaces, piece by piece.

https://inclusive-components.design/

JavaScript & Accessibility

1. **Don’t** use JavaScript for functionality HTML provides
2. Where possible, **don’t** require JavaScript for critical features

1. **Do** use JavaScript to improve on the default accessible behaviour
2. **Do** use JavaScript to create components that don’t exist

"**By default, HTML is accessible**, if used correctly.

Web accessibility involves ensuring that content remains accessible."

https://developer.mozilla.org/en-US/docs/Learn/Accessibility

“So next time someone tells you to make things accessible, tell them that instead **you don't intend to make it inaccessible in the first place.**”

— George Kemp

https://dev.to/gkemp94/the-web-is-accessible-by-default-stop-breaking-it-4cg4

Thank you!

Ire Aderinokun 🇳🇬

COO & VP Engineering of **Helicarrier**

Google **Web Expert**

ireaderinokun.com

bitsofco.de

@ireaderinokun