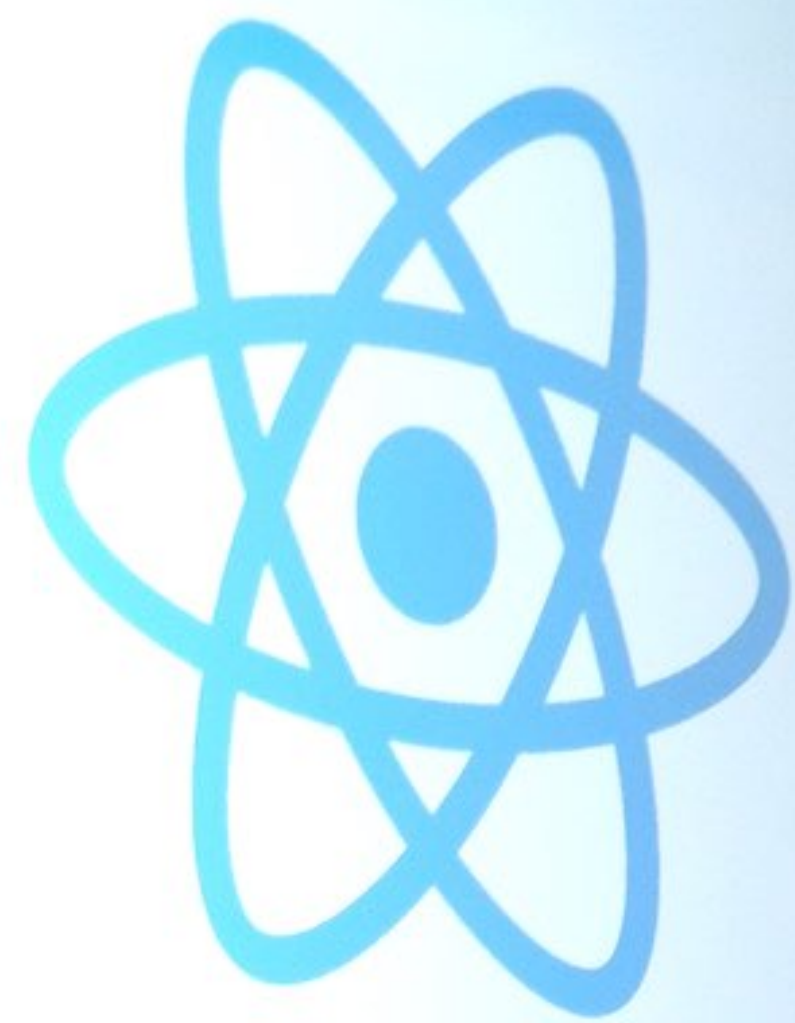


Lost in Translation

Manuel Matuzović

Paris Web, Paris 2022



Speaking at React Finland 2019

My Accessibility Journey: What I've Learned So Far

ARTICLES EVENTS TOPICS WRITE FOR US LANGUAGE (ENGLISH) ▾

A LIST APART

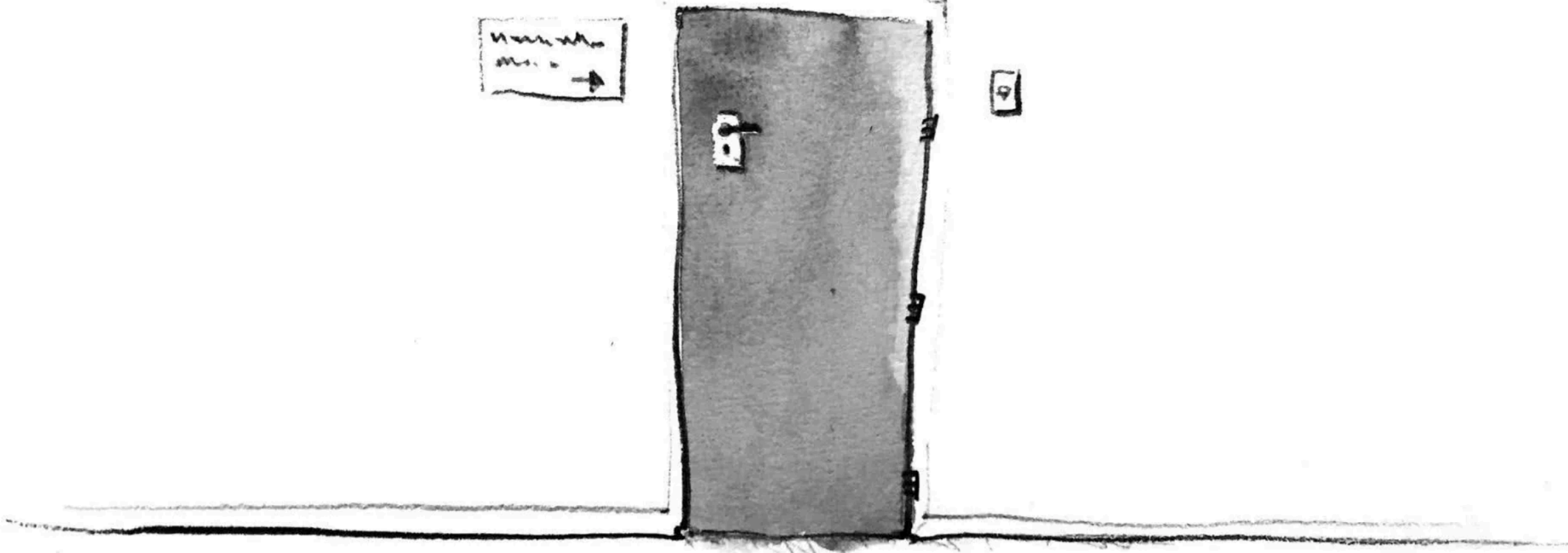


Illustration by Dougal MacPherson

My Accessibility Journey: What I've Learned So Far

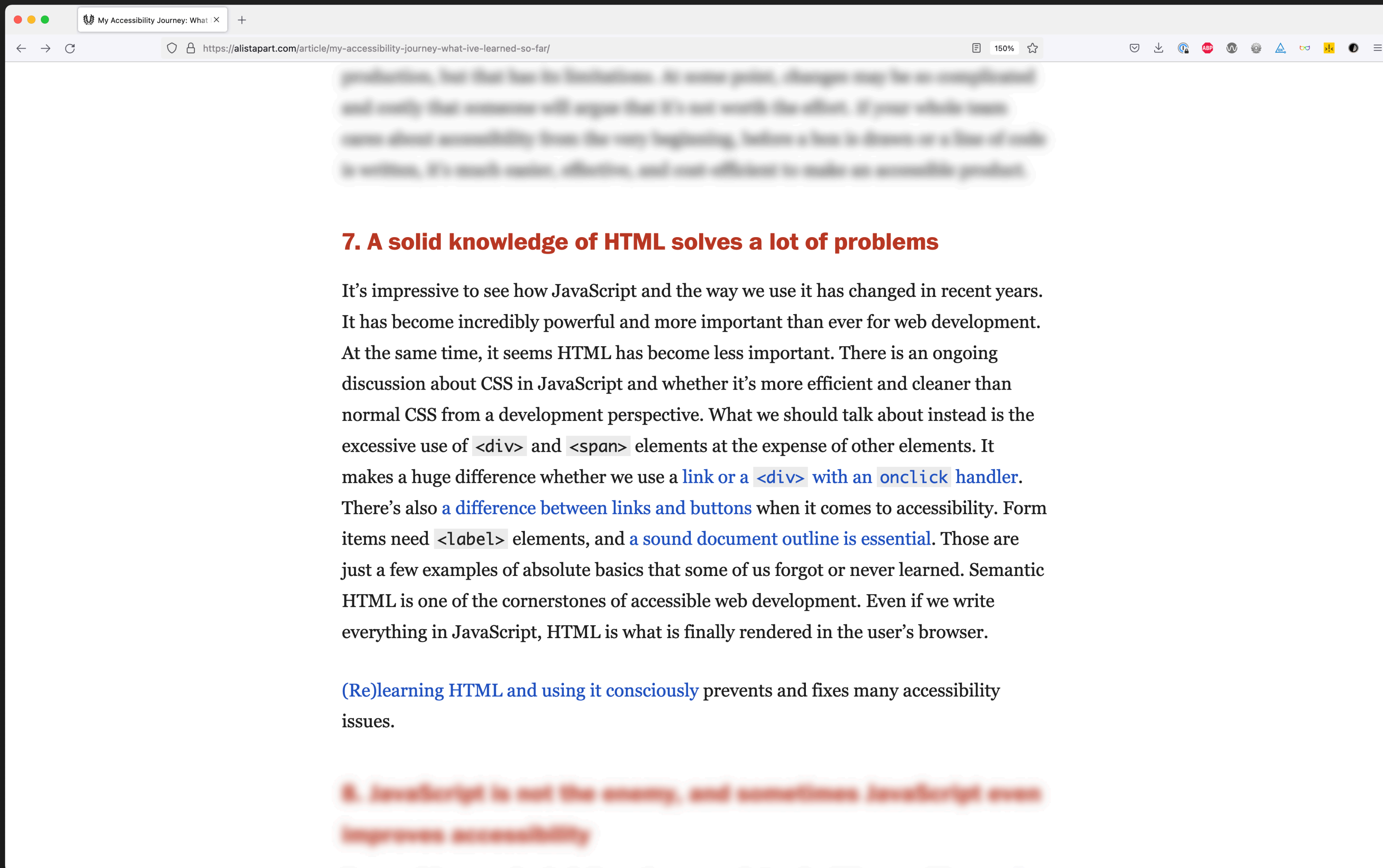
by [Manuel Matuzović](#) · February 06, 2018

Published in [Accessibility](#), [Usability](#)

Last year I gave a talk about CSS and accessibility at the [stahlstadt.js meetup](#) in Linz, Austria. Afterward, an attendee asked why I was interested in accessibility: Did I or someone in my life have a disability?

I'm used to answering this question—to which the answer is no—because I get it all the time. A lot of people seem to assume that a personal connection is the only reason

16 Comments





The Button Cheat Sheet

https://www.buttoncheatsheet.com

THE BUTTON CHEAT SHEET

Do you need a button for your next project but you're not sure about the right markup? Don't worry, **The Button Cheat Sheet**™ has got you covered.

If you're not sure whether you should use a button or a link, watch [The Links vs. Buttons Showdown](#) by [Marcy Sutton](#).

The HTML5 <button>

Rating: Good, use it! 🍌

```
<button>Yea or nay?</button>
```

YEA OR NAY?

Details

- That's a proper button!
- It's focusable
- It has correct key events by default
- Correct `button` role

Tags used in HTML

info.cern.ch/hypertext/WWW/MarkUp/Tags.html

HTML Tags

This is a list of tags used in the [HTML](#) language. Each tag starts with a tag opener (a less than sign) and ends with a tag closer (a greater than sign). Many tags have corresponding closing tags which identical except for a slash after the tag opener. (For example, the [TITLE](#) tag).

Some tags take parameters, called attributes. The attributes are given after the tag, separated by spaces. Certain attributes have an effect simply by their presence, others are followed by an equals sign and a value. (See the [Anchor](#) tag, for example). The names of tags and attributes are not case sensitive: they may be in lower, upper, or mixed case with exactly the same meaning. (In this document they are generally represented in upper case.)

Currently HTML documents are transmitted without the normal SGML framing tags, but if these are included parsers will ignore them.

Title

The title of a document is given between title tags:

```
<TITLE> ... </TITLE>
```

The text between the opening and the closing tags is a title for the hypertext node. There should only be one title in any node. It should identify the content of the node in a fairly wide context, and should ideally fit on one line.

The title is not strictly part of the text of the document, but is an attribute of the node. It may not contain anchors, paragraph marks, or highlighting. the title may be used to identify the node in a history list, to label the window displaying the node, etc. It is not normally displayed in the text of a document itself. Contrast titles with [headings](#) .

Next ID

Obsolete: NeXT Browser only. May be ignored. This tag takes a single attribute which is the number of the next document-wide numeric identifier to be allocated (not good SGML). Note that when modifying a document, old anchor ids should not be reused, as there may be references stored elsewhere which point to them. This is read and generated by hypertext editors. Human writers of HTML usually use mnemonic alpha identifiers. Browser software may ignore this tag. Example of use:

```
<NEXTID 27>
```

Base Address

[Anchors](#) specify addresses of other documents, in a from relative to the address of the current document. Normally, the address of a document is known to the browser because it was used to access the document. However, is a document is mailed, or is somehow visible with more than one address (for example, via its filename and also via its library name server catalogue number), then the browser needs to know the base address in order to correctly deduce external document addresses.

The format of this tag is not yet specified. NOT CURRENTLY USED

Anchors

The format of an anchor is as follows:

```
<A NAME=xxx HREF=xxx> ... </A>
```

The text between the opening tag and the closing tag is either the start or destination (or both) of a link. Attributes of the anchor tag are as follows.

HREF
If the HREF attribute is present, the anchor is sensitive text: the start of a link. If the reader selects this text, he should be presented with another document whose network address is defined by the value of the HREF attribute . The format of the network address is specified [elsewhere](#) . This allows for the form HREF=#identifier to refer to another anchor in the same document. If the anchor is in another document, the attribute is a [relative name](#) , relative to the documents address (or specified [base address](#) if any).

NAME
The attribute NAME allows the anchor to be the destination of a link. The value of the parameter is that part of a hypertext address which follows the [hash sign](#) .

TYPE
An attribute TYPE may give the relationship described by the hypertext link. The type is expressed by a string for extensibility. Strings for types with particular semantics will be registered by the W3 team. The default relationship if none other is given is void.

All attributes are optional, although one of NAME and HREF is necessary for the anchor to be useful.

IsIndex

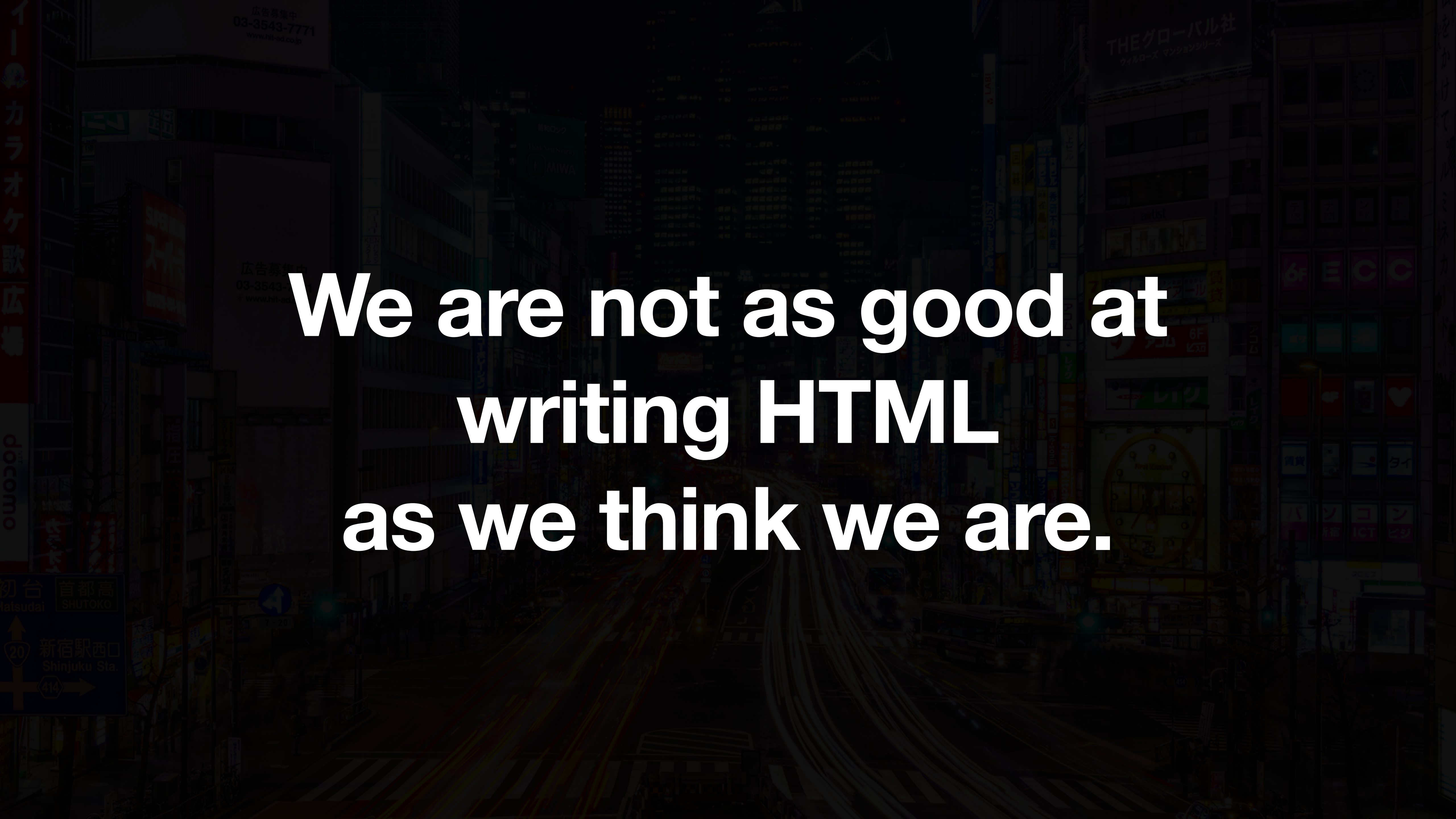
This tag informs the reader that the document is an index document. As well as reading it, the reader may use a keyword search.

Format:

```
<ISINDEX>
```

The node may be queried with a keyword search by suffixing the node address with a question mark, followed by a list of keywords separated by plus signs. See the [network address format](#) .






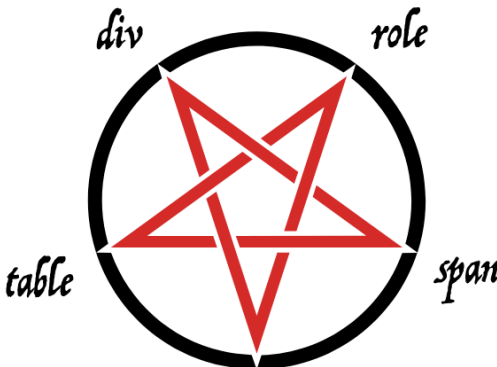
**We are not as good at
writing HTML
as we think we are.**

```
<main>
  <section id="main-content">
    <header id="main-header">
      <section class="container-fluid">
        <section class="row">
          <article class="content col-sm-12">
            <section class="content-inner">
              <div class="content__body">
                <article class="slider">
                  <section class="slide"> ... </section>
                </article>
              </div>
            </section>
          </article>
        </section>
      </section>
    </header>
  </section>
</main>
```

Markup from hell - HTMHell x +

https://www.htmhell.dev

Hell **Heaven** Newsletter 



HTMHell


A collection of bad practices in HTML, copied from real websites.

#31 additional “assistance”

submitted on Apr 12, 2022 by [Stefania](#)

```
<a href="/contact" aria-label="If you find that you need additional assistance in navigating or accessing the content of this website, please call our customer service toll free number 1-800-666-8654309" title="If you find that you need additional assistance in navigating or accessing the content of this website, please call our customer service toll free number 1-800-666-8654309">
  Contact
</a>
<a href="/login" aria-label="If you find that you need additional assistance
```

→ [Details and tips on how to fix the diabolic code of #31.](#)



#30 Bullet “list”



```
<h1>Product Status</h1>
<h2>Is the product available?</h2>
<div>
  <h3>
    <div>
      <div>
        <i>
          <h3>
            It's <a>available</a>.
          </h3>
        </i>
      </div>
    </div>
  </div>
</h3>
</div>
```

```
<div class="btn">Click here!</div>
```

The Elements of HTML

1. `<div>`
2. `<script>`

That's it, that's all. #yolo

The Elements of HTML

1. <header>
2. <footer>
3. <main>
4. <section>
5. <article>
6. <aside>
7. <progress>
8. <meter>
9. <details>
10. <summary>
11. <dialog>
12. <address>
13. <button>
14. <blockquote>
15. <abbr>
16. <kbd>
17. <mark>
18.
19. <ins>
20. <caption>
21. <track>
22. <output>
23. <label>
24. <fieldset>
25. <textarea>
26. <video>
27. ...

HTML Tags Memory Test

Paul [+ Follow](#)


View in Editor

HTML Tags Memory Test

How many HTML tags can you remember? 115 to recall...

Tag: [Guess](#)

HTML Tags Memory Test by [plfstr](#)

The background is a dark, stylized illustration of a city street scene. It features buildings with various signs and advertisements, including one for 'docomo' and another for 'THE グローバル社'. A road sign in the bottom left corner indicates '初台 首都高 Matsudai SHUTOKO' and '新宿駅西口 Shinjuku Sta.' with route numbers '20' and '414'. The overall aesthetic is that of a dense urban environment with a focus on digital and commercial elements.

**Designers know too little
about HTML.**



Some people just don't care.

```
<div class="nav">  
  <div>  
    <div>about</div>  
    <div>thoughts</div>  
  </div>  
</div>
```

```
<div class="nav">
  <a href="/about">
    <button>about</button>
  </a>

  <a href="/thoughts">
    <button>thoughts</button>
  </a>
</div>
```



**We put too much trust
and hope in ARIA.**

Loading... +
https://webaim.org/projects/million/


WebAIM
web accessibility in mind

services articles resources **projects** community

Search:

Introduction to Web Accessibility

WebAIM Training



The WebAIM Million

The 2022 report on the accessibility of the top 1,000,000 home pages

Last updated: Mar 31, 2022

[Home](#) > [WebAIM Projects](#) > The WebAIM Million

Article Contents

- Introduction
- The Sample
 - Method
- Detected Errors
- Home Page Complexity
- Error Density
- WCAG Conformance
- Low Contrast Text
- Images and Alternative Text
- Form Labeling
- Headings
- Regions
- ARIA
- Links
- Skip Links
- Doctypes
- Site Categories
- TLDs
- Languages
- Technologies
 - Content Management Systems
 - JavaScript Frameworks
 - JavaScript Libraries
 - Web Frameworks
 - Ecommerce Platforms
 - Advertisement Networks
 - Other Technologies

Home pages with ARIA present averaged 70% more detected errors than those without ARIA.



imgflip.com

If you can use a native HTML element or attribute with the semantics and behavior you require already built in, instead of re-purposing an element and adding an ARIA role, state or property to make it accessible, then do so.

***Use native HTML if possible.
No ARIA is better than bad ARIA.***

Building the main navigation for × +


← → ↻ https://web.dev/website-navigation/

Home > All articles

Building the main navigation for a website

This tutorial describes how to build an accessible main navigation of a website. You learn about semantic HTML, accessibility, and how using ARIA attributes can sometimes do more harm than good.

Sep 7, 2022



Manuel Matuzović
[Twitter](#)

There are many different ways of building the main navigation of a website, in terms of styling, functionality, and the underlying markup and semantic information. If the implementation is too minimalist, it works for most people, but the user experience (UX) might not be great. If it's over-engineered, it might confuse users or even hinder them from being able to access it at all.

For most websites, you want to build something that's neither too simple, nor too complicated.

Building layer by layer

In this tutorial you start with a basic setup and add features layer by layer up to a point where you provide just enough information, styling, and functionality to please most users. To achieve that you make use of the progressive enhancement principle, which states that you start with the most fundamental and robust solution and progressively add layers of functionality. If one layer fails to work for some reason, the navigation will still work because it never falls back to the underlying layers.

On this page

- Building layer by layer
- Basic structure
- Highlight the active page
- Announce the number of items
- Add a landmark
- Hide the navigation on narrow viewports
 - Adding a burger button
 - Hiding the list
 - Animating the list
- Improve focus styling
 - Browser support for :focus-visible
- Navigations versus menus
 - The navigation role
 - The menu role
 - The menubar role
 - The menuitem role
 - When is a menu appropriate?
 - Implications of using menu roles
- Additional Resources

[SHARE](#)

[SUBSCRIBE](#)



Most common issues

WCAG Failure Type	% of home pages in 2022	% of home pages in 2021	% of home pages in 2020	% of home pages in 2019
Low contrast text	83.9%	86.4%	86.3%	85.3%
Missing alternative text for images	55.4%	60.6%	66.0%	68.0%
Empty links	50.1%	51.3%	59.9%	58.1%
Missing form input labels	46.1%	54.4%	53.8%	52.8%
Empty buttons	27.2%	26.9%	28.7%	25.0%
Missing document language	22.3%	28.9%	28.0%	33.1%

The screenshot shows a web browser window with the following elements:

- Browser Tab:** "Common accessibility issues the X"
- Address Bar:** "https://hidde.blog/common-a11y-issues/"
- Navigation Bar:** "Hidde's blog" (highlighted in red), "Blog", "Talks", "About me", "Contact" (all in a dark blue bar).
- Metadata:** "Published 21 DAYS AGO · [thoughts](#)"
- Section Header:** "Common accessibility issues that you can fix today" (in large red font).
- Text Content:**
 - Paragraph 1: "WebAIM have come out with their latest [report](#) on which accessibility issues they found in the top million websites that they tested automatically. What is some low hanging fruit you could fix today?"
 - Paragraph 2: "For context, WebAIM, a non profit based out of Utah in the US, have done their '[WebAIM Million](#)' project since 2019. They post an extensive analysis every year, looking at trends and improvements/decline in web accessibility over time. I find these posts very insightful and use them to inform my own workshops and outreach. It's definitely recommended reading!"
 - Paragraph 3: "There are some caveats to be added with surveys based on automated accessibility testing. One is that 'ease of detectability' does not correlate with 'impact on end users'. There are issues that are easy to detect and issues that impact end users most, these are not necessarily the same. Automated tests also cover only a small part of all accessibility, as some things aren't detectable by machines (yet, or ever). I'm not suggesting this makes the survey less useful or good, but wanted to call it out explicitly. The [ACT-Rules Community Group](#) at the W3C works on harmonising test rules for things that *are* testable."
 - Paragraph 4: "Ok, let's look at the top issues and how developers, browsers and CMSes can take away barriers today. Some of these include ideas about what users can do (important caveat: none of this should be user responsibility, website owners should not expect users to use or know about these tools)."

Christian Heilmann

About this Archives Slides Bookmarks codepo8 @codepo8

One million broken web sites – and a way to prevent that

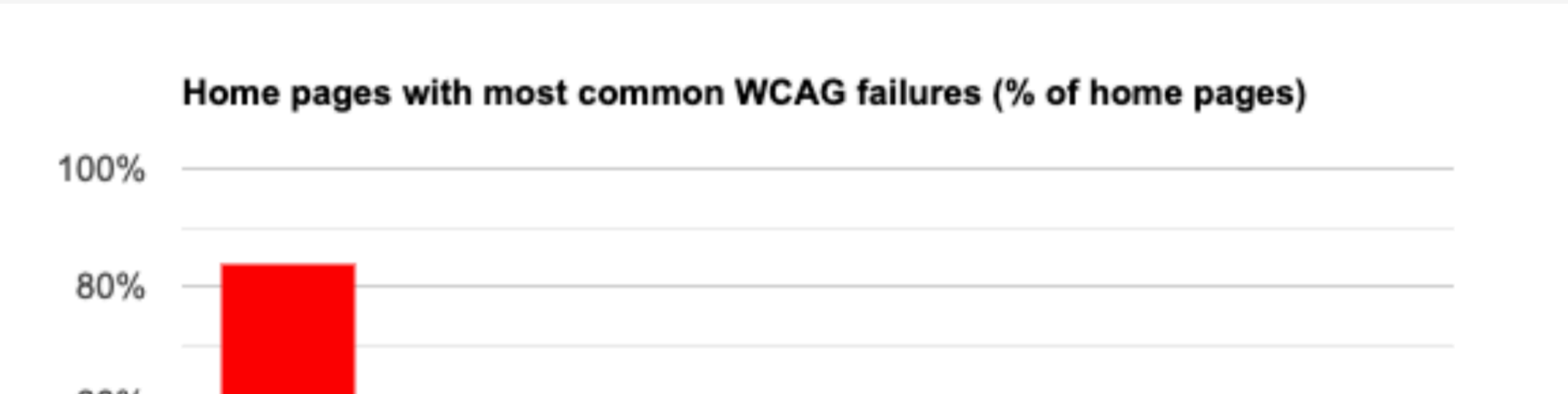
Tuesday, April 12th, 2022 at 12:37 am

[Webaim.org](#) releases an annual report called the [WebAIM Million](#). This is a deep analysis of the one million most visited web sites how accessible they are. And every year the result is pretty grim.

// Across the one million home pages, 50,829,406 distinct accessibility errors were detected—an average of 50.8 errors per page.

This is 1.1% fewer errors than last year, but at the same time home pages of sites have become more complex. On average, home pages have 955 elements and 5.3% of those have accessibility errors. This means that users with disabilities would expect to encounter errors on 1 in every 19 home page elements with which they engage.

The most disappointing part about this state of affairs is the type of problems.



Failure Type	Percentage
Most common failure	~80%

Most common issues

- Low contrast text
- Missing alternative text for images
- Empty links
- Missing form input labels
- Empty buttons
- Missing document language

**We're wrongfully downplaying
the complexity of HTML due to
the simplicity of its syntax.**



There's more to HTML than tags and attributes.

Most common issues

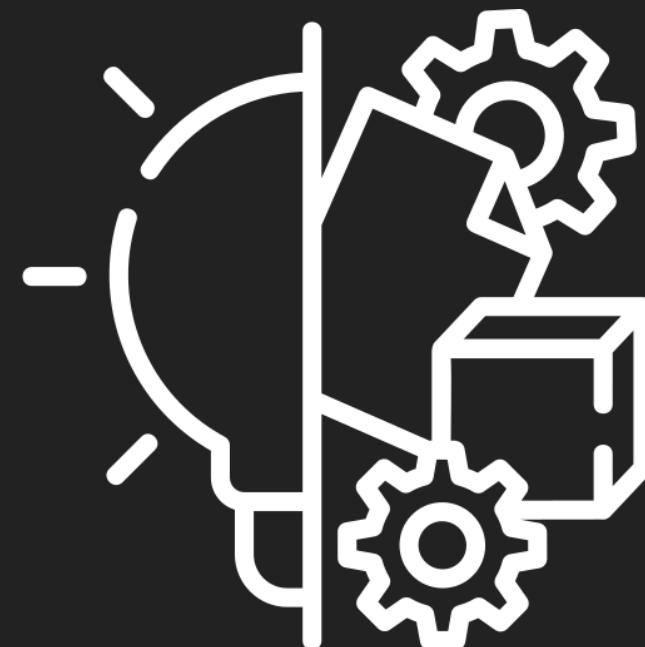
- Low contrast text
- Missing alternative text for images
- Empty links
- Missing form input labels
- Empty buttons
- Missing document language

**Accessibility issues don't
just come from what's
visible in a design,
but from what's not visible.**

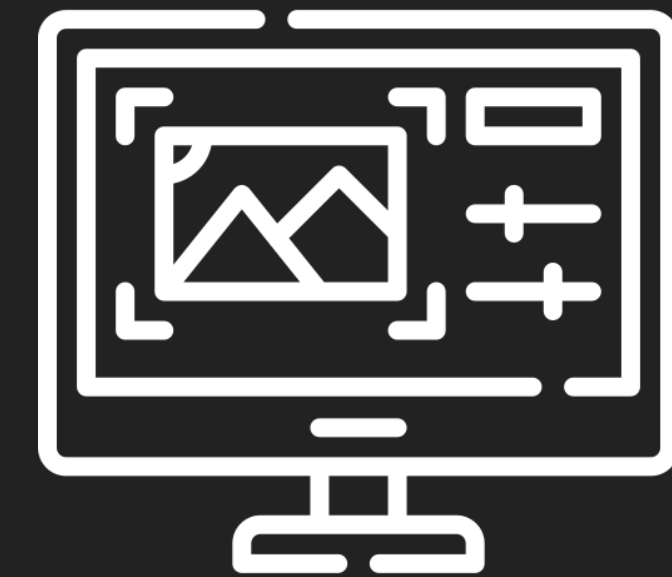
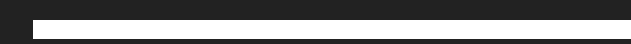
Design to Code Translation



Exploration



Understanding

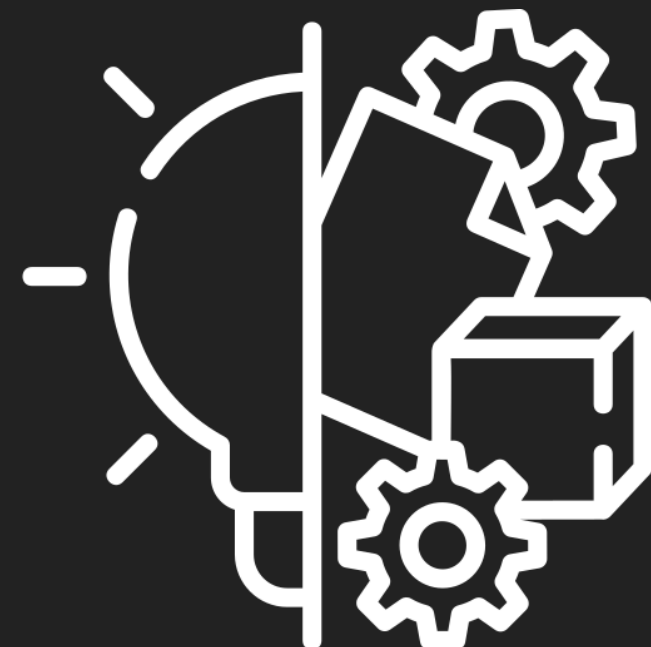
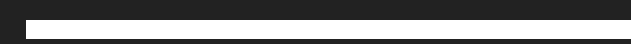


Development

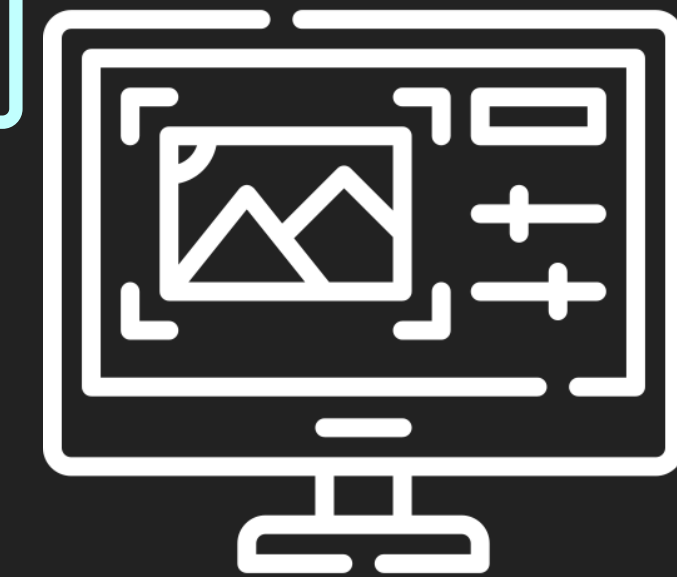
Design to Code Translation



Exploration

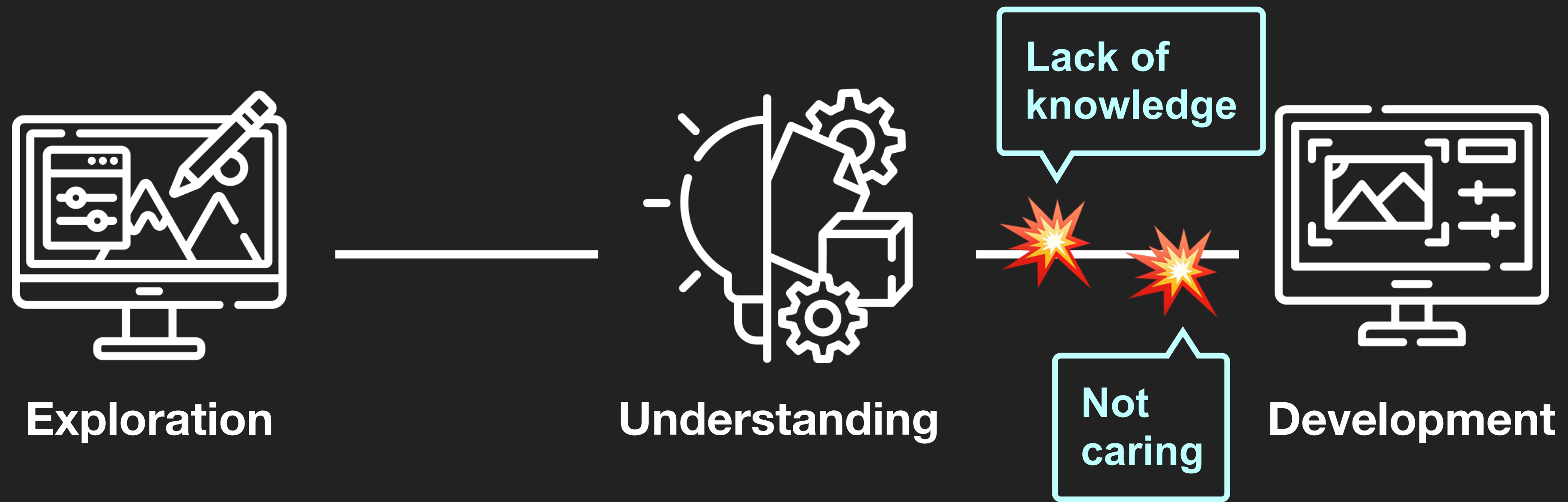


Understanding

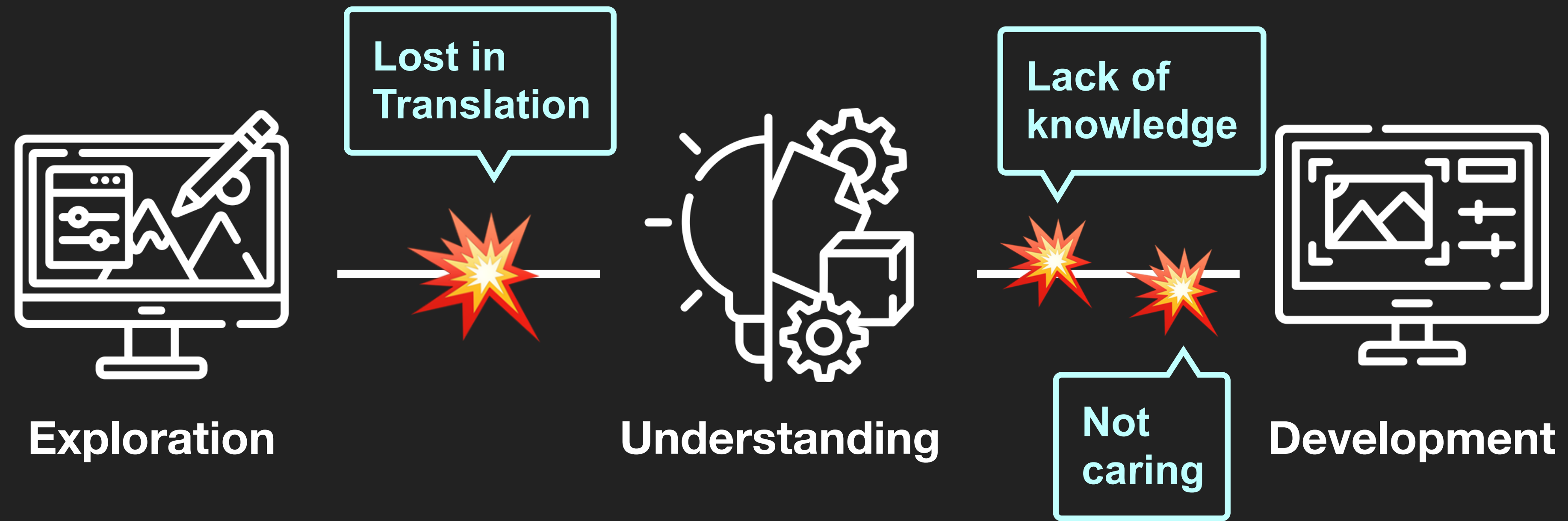


Development

Design to Code Translation

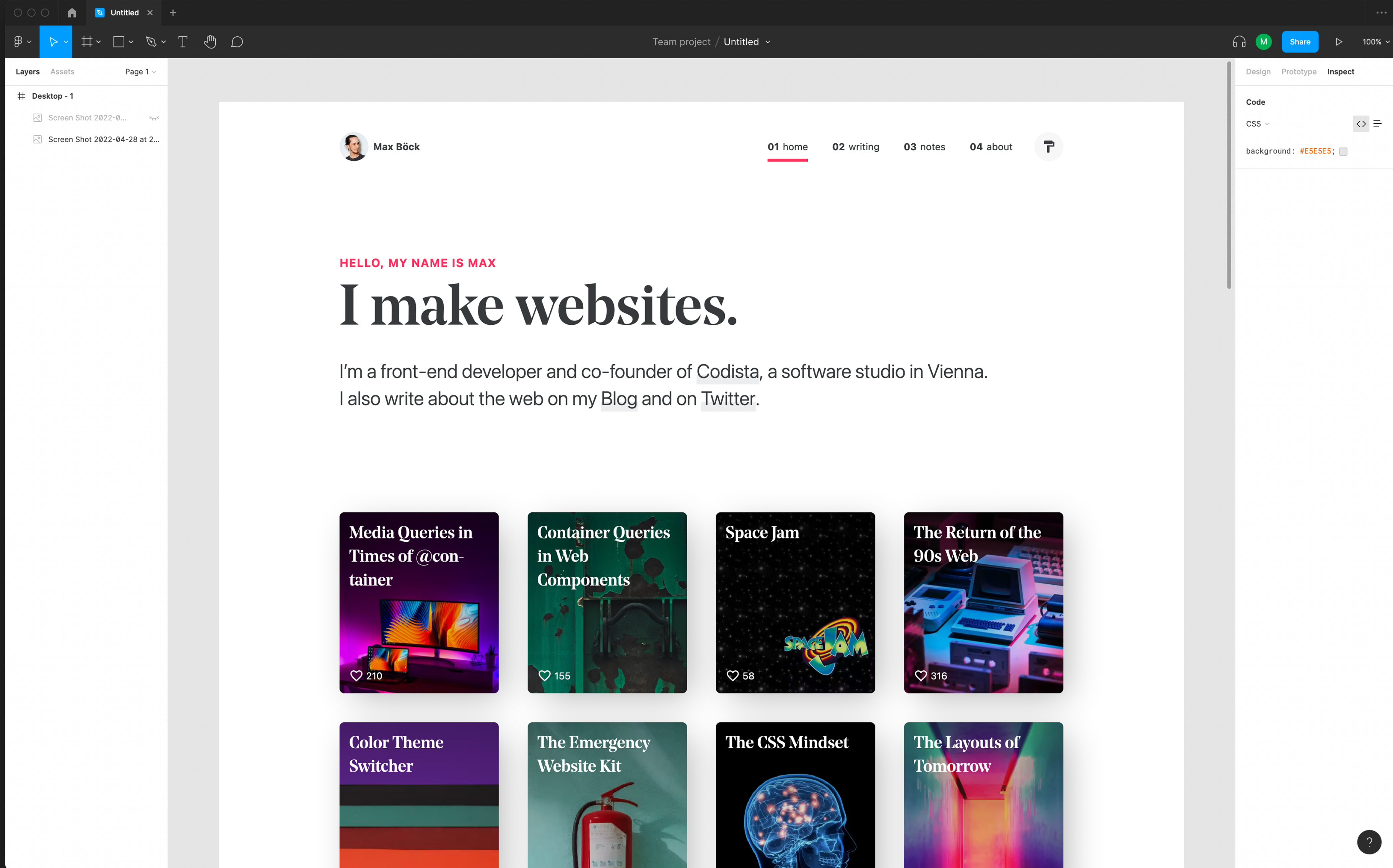


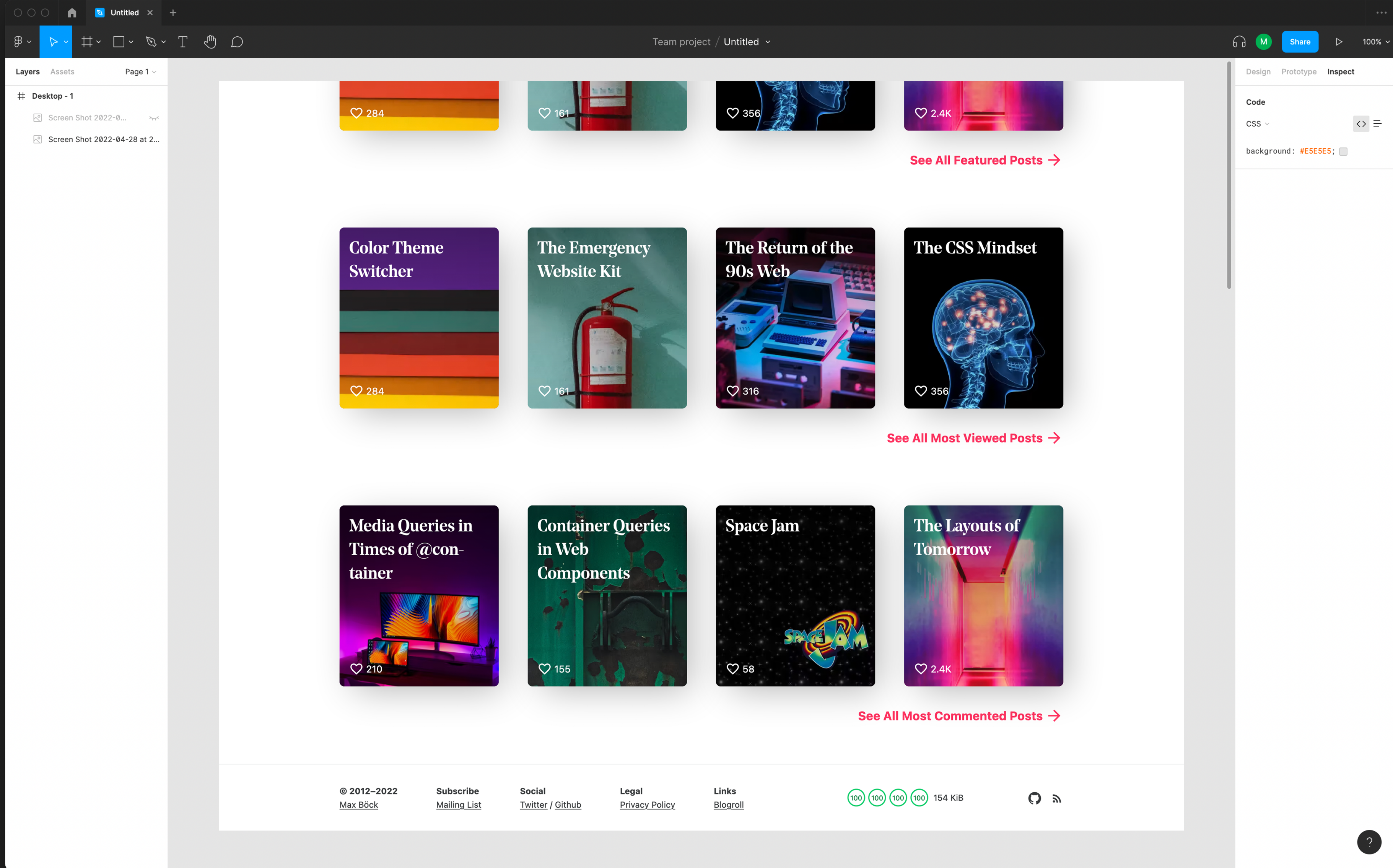
Design to Code Translation



Our decisions have consequences

- Basic understanding
- Information
- Orientation
- Overview
- Navigation





01 home

02 writing

03 notes

04 about



sites.

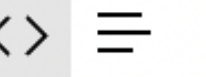
under of [Codista](#), a software studio in Vienna.

og and on [Twitter](#).

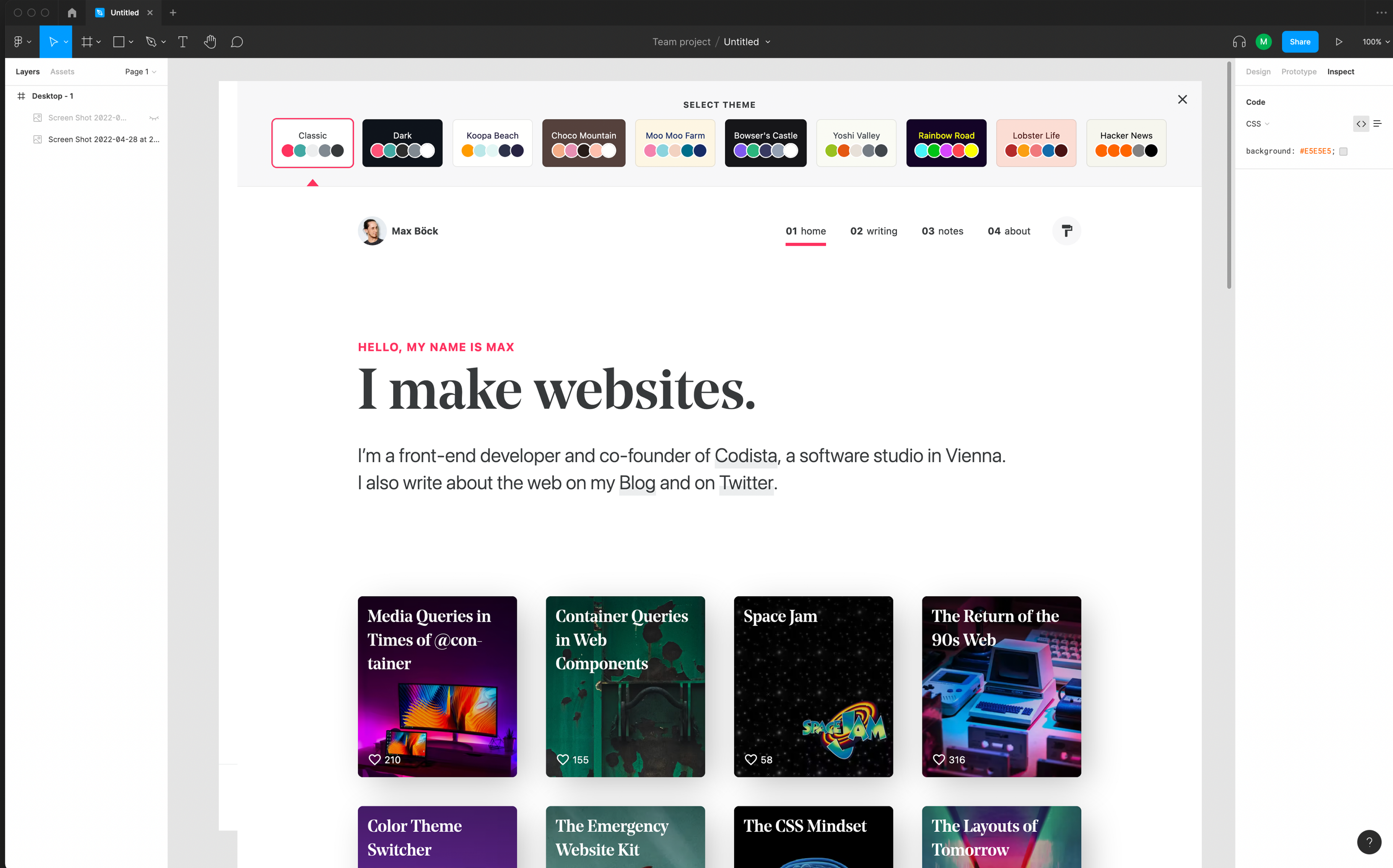
Design Prototype Inspect

Code

CSS ▾



background: #E5E5E5;



01 home

02 writing

03 notes

04 about



sites.

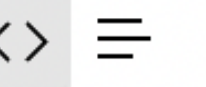
under of [Codista](#), a software studio in Vienna.

og and on [Twitter](#).

Design Prototype Inspect

Code

CSS ▾



background: #E5E5E5;



```
<svg>  
  <path d="" />  
</svg>
```



```
<div class="btn">  
  <svg>  
    <path d="" />  
  </svg>  
</div>
```



```
<button>  
  <svg>  
    <path d="" />  
  </svg>  
</button>
```



```
<button aria-label="Change theme">  
  <svg aria-hidden="true">  
    <path d="" />  
  </svg>  
</button>
```



```
<button>  
  Change theme  
  <svg aria-hidden="true">  
    <path d="" />  
  </svg>  
</button>
```



```
<button>  
  <span class="sr-only">Change theme</span>  
  <svg aria-hidden="true">  
    <path d="" />  
  </svg>  
</button>
```



```
<button aria-expanded="false">  
  <span class="sr-only">Change theme</span>  
  <svg aria-hidden="true">  
    <path d="" />  
  </svg>  
</button>
```



```
<button aria-expanded="false" aria-controls="theme-panel">  
  <span class="sr-only">Change theme</span>  
  <svg aria-hidden="true">  
    <path d="" />  
  </svg>  
</button>
```


The Perfect Button

- Focusable, interactive element (<button>)

The Perfect Button

- Focusable, interactive element (<button>)
- Accessible name (a label)

The Perfect Button

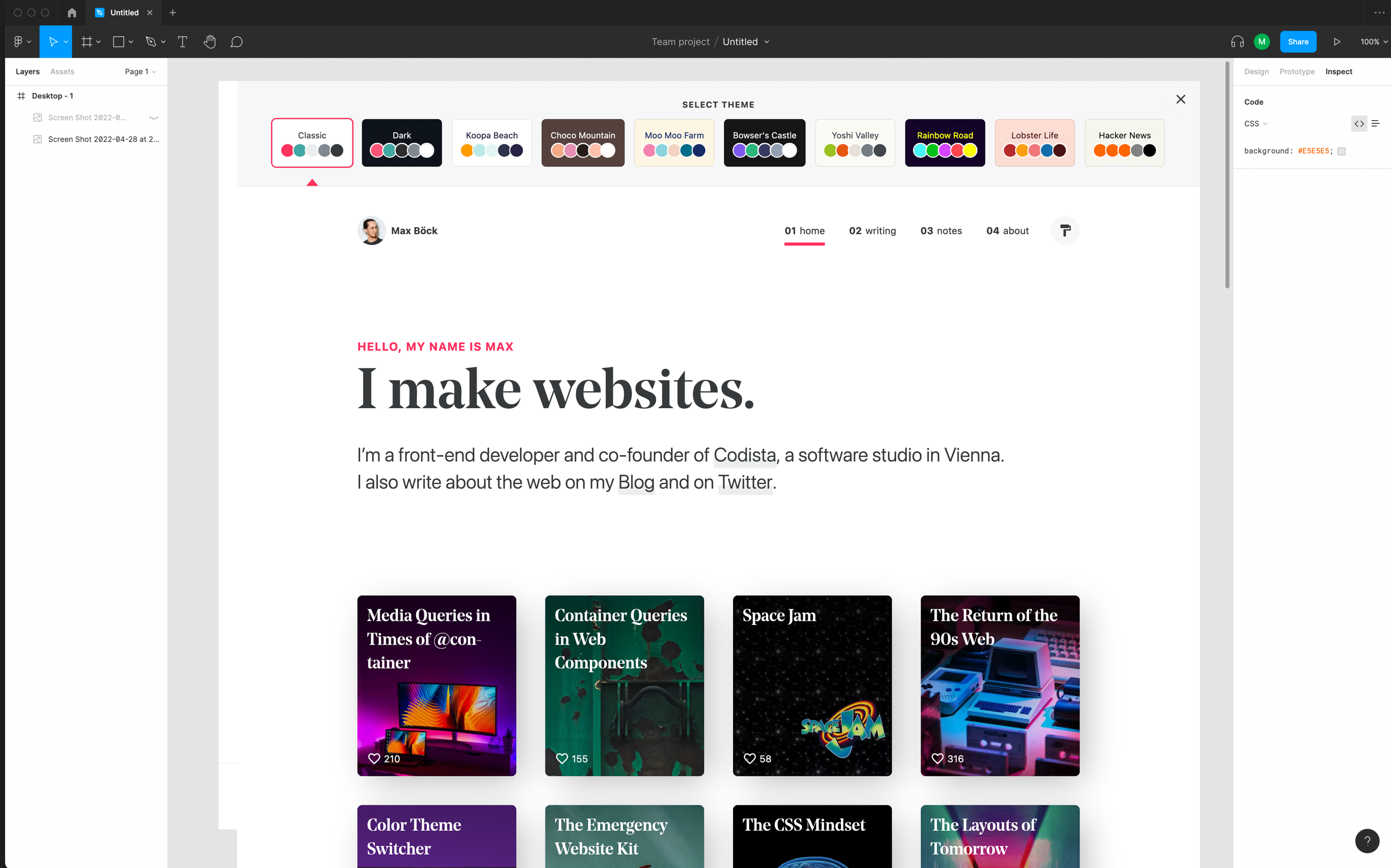
- Focusable, interactive element (<button>)
- Accessible name (a label)
- Correct role (button)

The Perfect Button

- Focusable, interactive element (<button>)
- Accessible name (a label)
- Correct role (button)
- Communicate state ([aria-expanded])

The Perfect Button

- Focusable, interactive element (<button>)
- Accessible name (a label)
- Correct role (button)
- Communicate state ([aria-expanded])
- Communicate relationship ([aria-controls])





Share



100% ▾

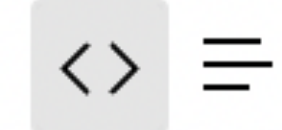
Design

Prototype

Inspect

Code

CSS ▾



background: #E5E5E5;

Valley

Rainbow Road

Lobster Life

Hacker News

02 writing

03 notes

04 about





Share

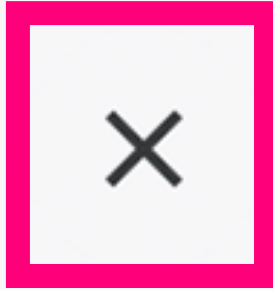


100% ▾

Design

Prototype

Inspect



Valley

Rainbow Road

Lobster Life

Hacker News

02 writing

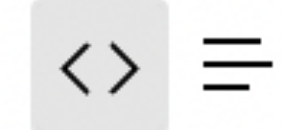
03 notes

04 about



Code

CSS ▾




background: #E5E5E5;

#20 HTMHell special: close butt X +

https://www.htmhell.dev/20-close-buttons/

Hell Heaven Newsletter



#20 HTMHell special: close buttons

submitted on May 23, 2020 by [Manuel](#)

This first HTMHell special inspects one of the most complicated and most controversial patterns in front-end development:

🔥 the close button. 🔥

In modals, ads, and other overlays you often find a button with a close symbol that allows users, or at least some of them, to close the overlay. This functionality is often limited to mouse users, because most implementations of close buttons suck.

After less than 2 hours of research, HTMHell presents a collection of 11 different bad practices.

Pattern 1: div and background image

```
<div class="close"></div>
```

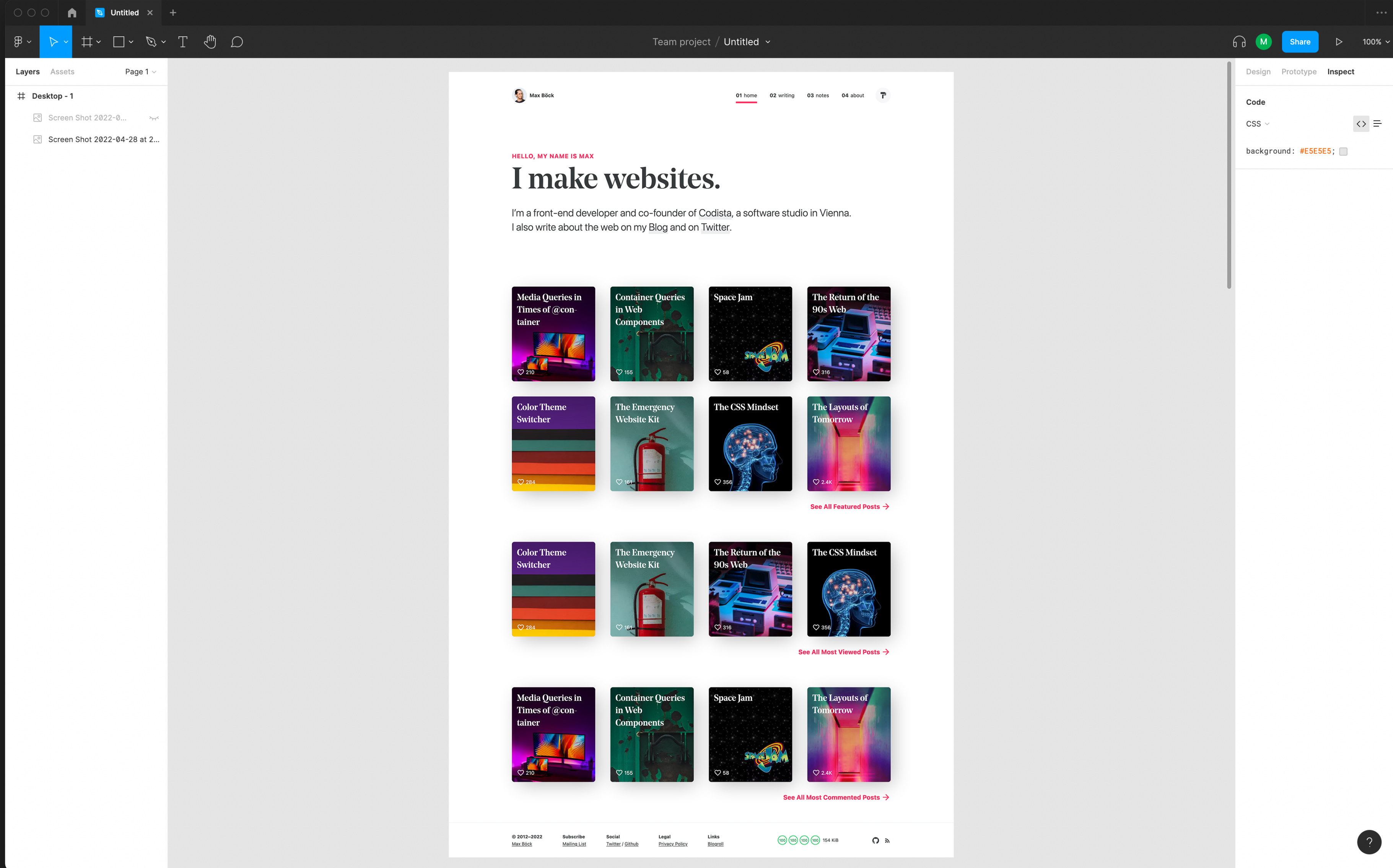
```
.close::after {
  background: url("close.png");
  content: "";
}
```



```
<button aria-expanded="true" aria-controls="theme-panel">  
  ×  
</button>
```



```
<button aria-expanded="true" aria-controls="theme-panel">  
  <span class="sr-only">Close</span>  
  <span aria-hidden="true">  
    ×  
  </span>  
</button>
```





Source: nordbayern.de

@mmatuzo



Source: natur-im-siedlungsraum.ch

@mmatuzo



Source: nabu-hadamar.de

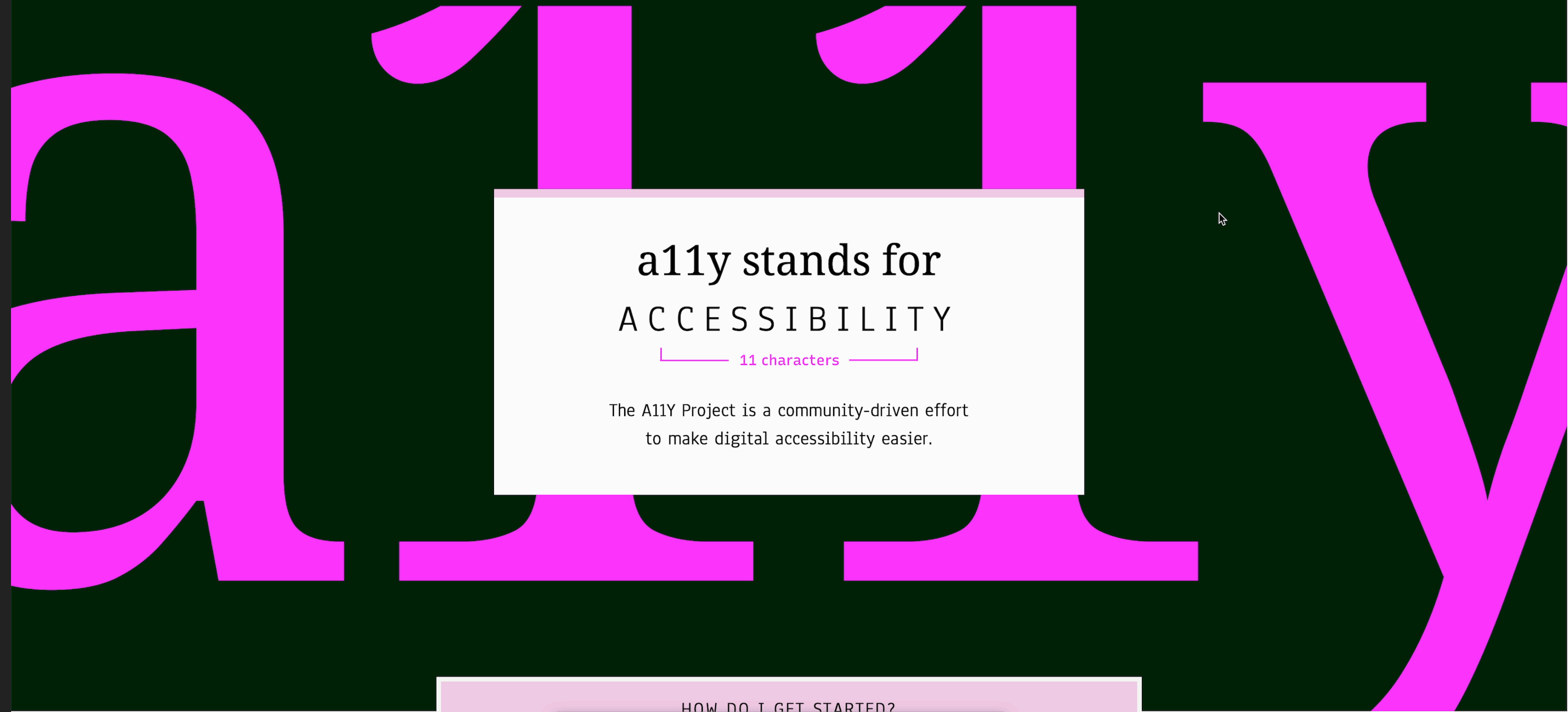
@mmatuzo



INHALT

1	Vielfalt der Wildbienen	7
2	Lebensweise und Entwicklungszyklus	9
3	Das Nest	15
3.1	Nistplatzwahl	17
3.2	Nistverhalten ausgewählter Wildbienen	21
3.2.1	Kunstvolle Nester aus Blüten- oder Laubblättern	21
3.2.2	Leere Schneckenschalen	27
3.2.3	Verlassene Gallen	30
3.2.4	Nester aus Mörtel und Steinchen	34
3.2.5	Zusammenfassende Darstellung	36
4	Beziehung zwischen Pflanzen und Bienen	41
4.1	Oligolektische und polyektische Arten	42
4.2	Pollenernte	44
4.3	Nektarernte	48
4.4.1	Ragwurz – trickreiche Blütenökologie	51
4.4.2	Schenkelbienen	52
5	Kuckucksbienen und andere Antagonisten	55
5.1	Kuckucksbienen (parasitische Bienen)	55
5.2	Kuckuckswespen (parasitische Wespen)	58
5.3	Ölkäfer	64
5.4	Fächerflügler	67
5.5	Wollschweber und andere Fliegen	68
5.6	Weitere Antagonisten	69
6	Veränderung der Kulturlandschaft	75
6.1	Stadt, Dorf, Siedlung	75
6.1.1	Gärten und Parks	76
6.1.2	Stadtbrachen	80
6.2	Ackerbaulich genutzte Kulturlandschaft	82
6.3	Wiesendominierte Kulturlandschaft	86
6.4	Wälder	90
6.5	Montane bis alpine Lagen	96
6.6	Hochalpine Lebensräume (Trümmerschuttfluren und Gletschervorfeld)	98
7	Besonders wertvolle Lebensräume (Auswahl)	101
7.1	Hohlwege und Lössterrassen	101
7.2	Sandlebensräume	106
7.3	Trockenrasen, Felssteppen und andere Trockenstandorte	108
7.4	Feuchtgebiete	115
7.5	Salzlebensräume	117
8	Gefährdung der Wildbienen	121
8.1	Nahrungsmangel	123

The A11Y Project supports the Black community and the Black Lives Matter movement. #BlackDisabledLivesMatter



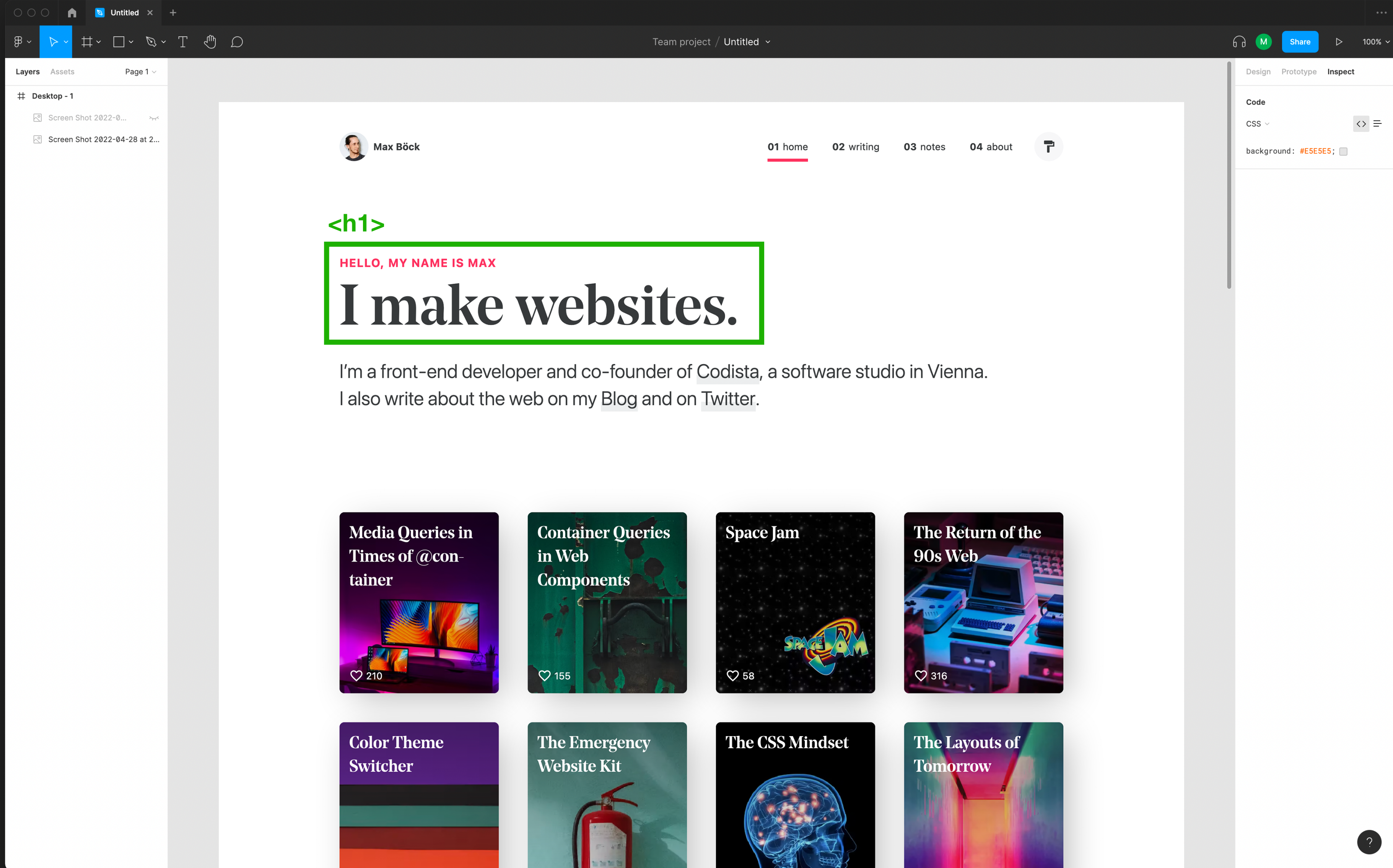
a11y stands for
ACCESSIBILITY
11 characters
The A11Y Project is a community-driven effort
to make digital accessibility easier.

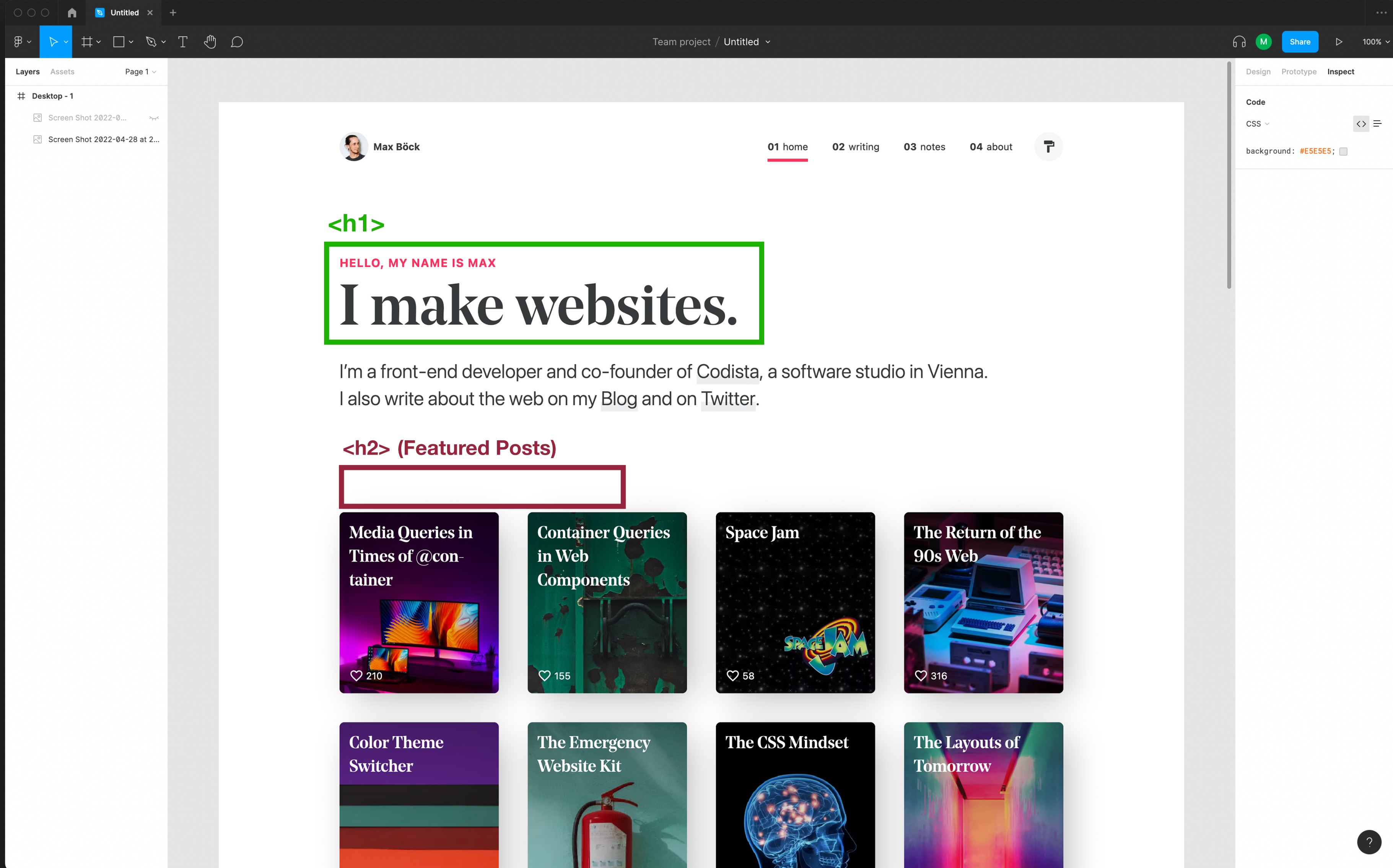
HOW DO I GET STARTED?

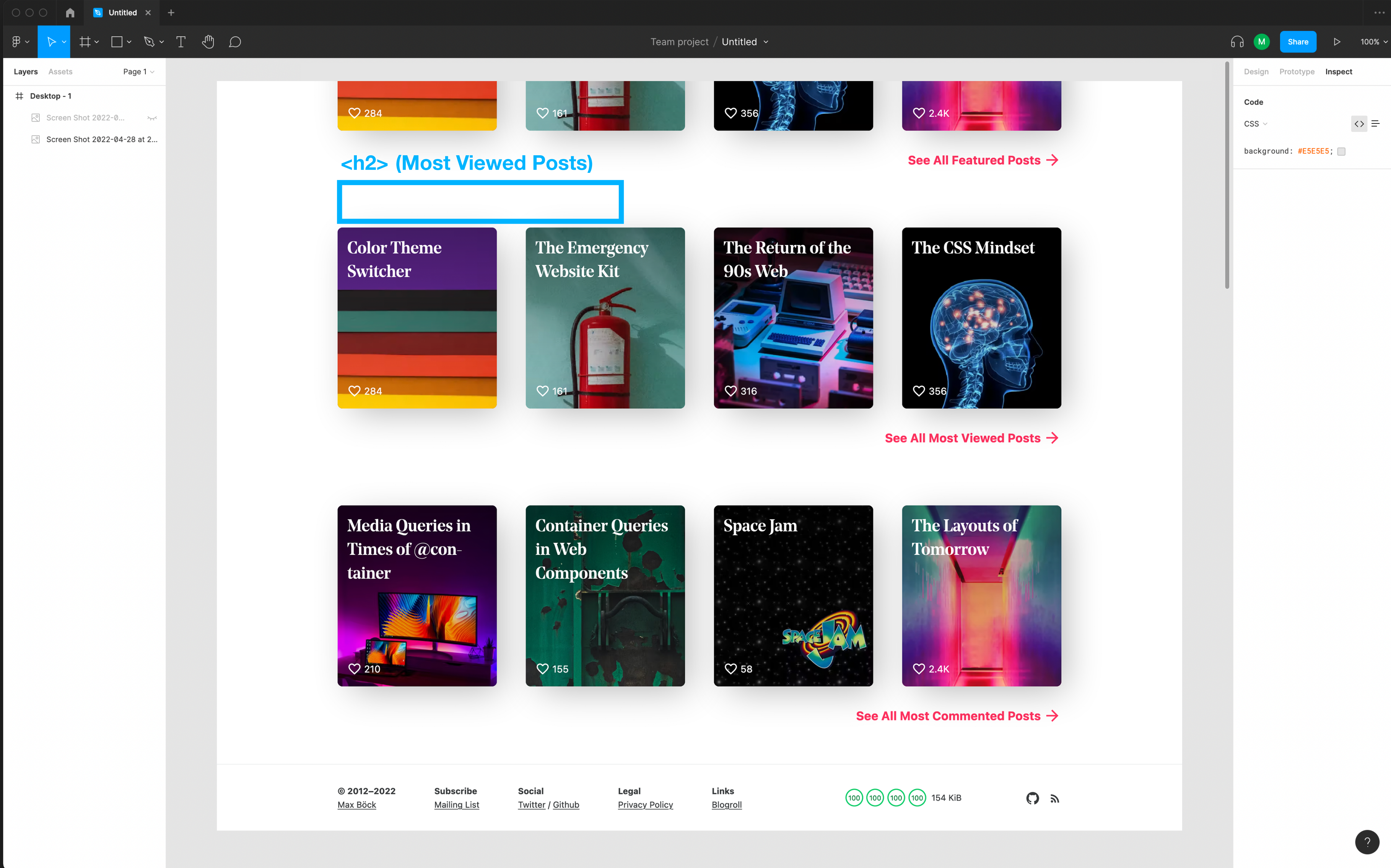
× VoiceOver on Safari, Home - The A11Y Project, window, Home - The A11Y Project, web content, has keyboard focus

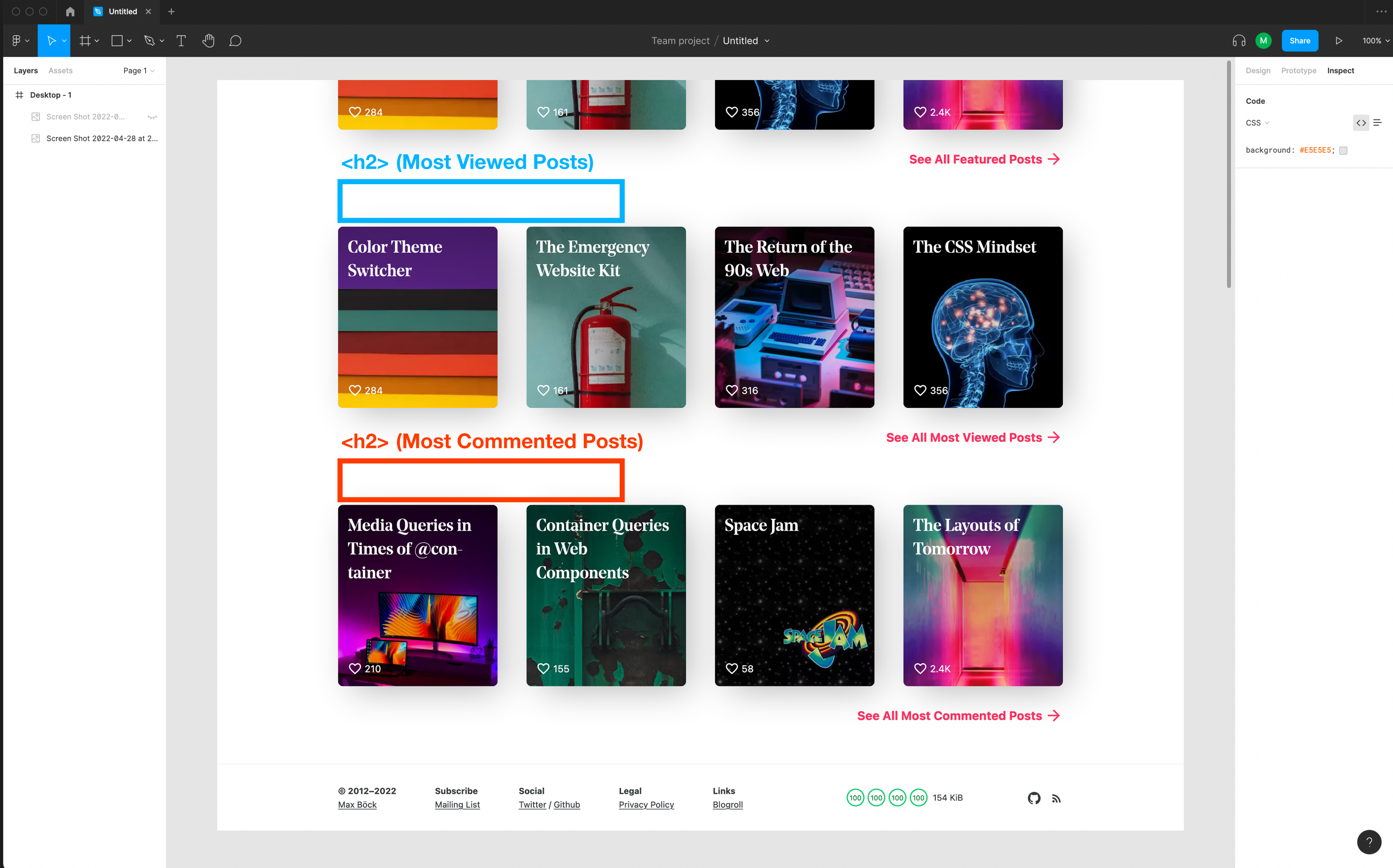
Build basics

Learn the fundamentals and principles behind accessible design



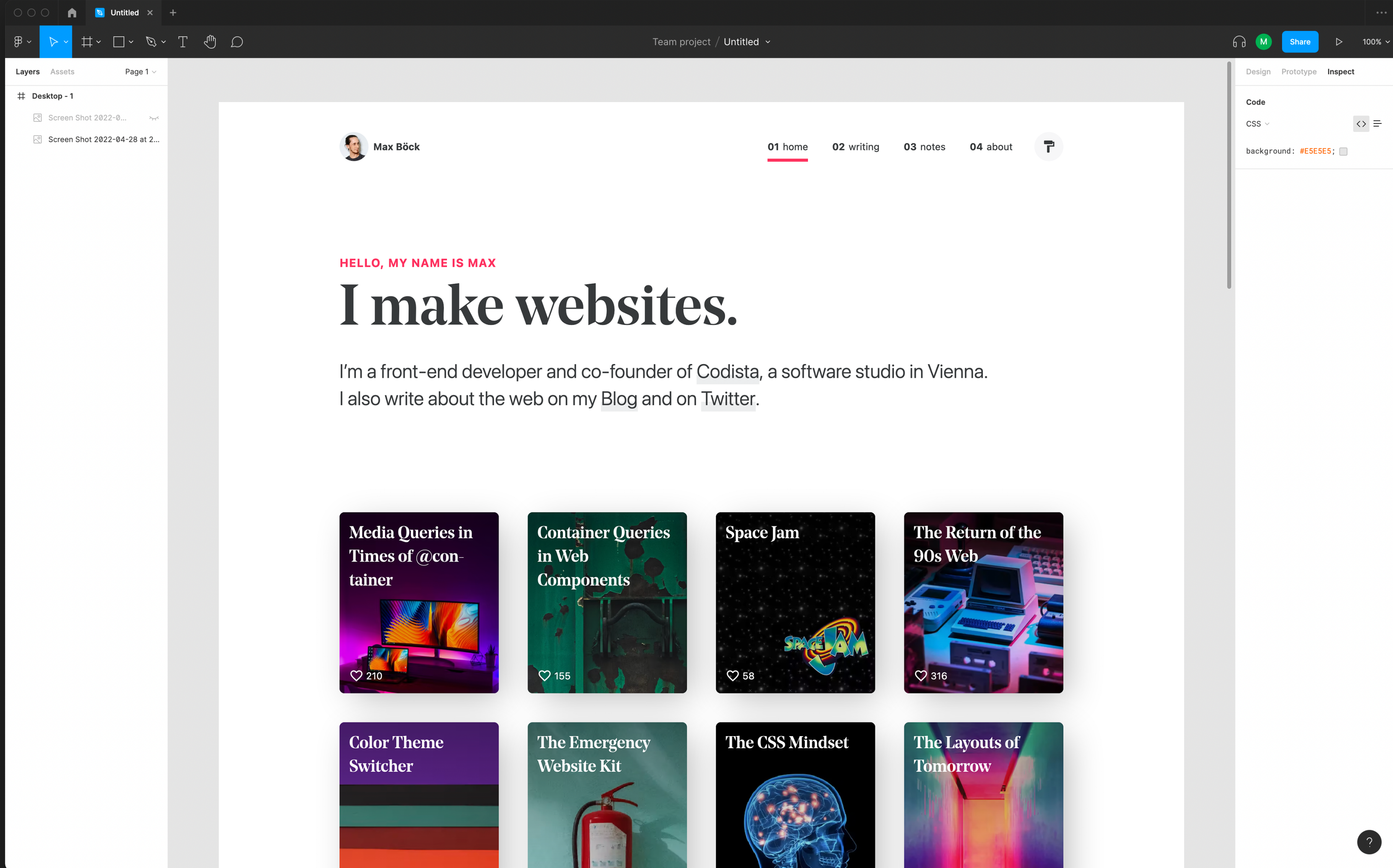


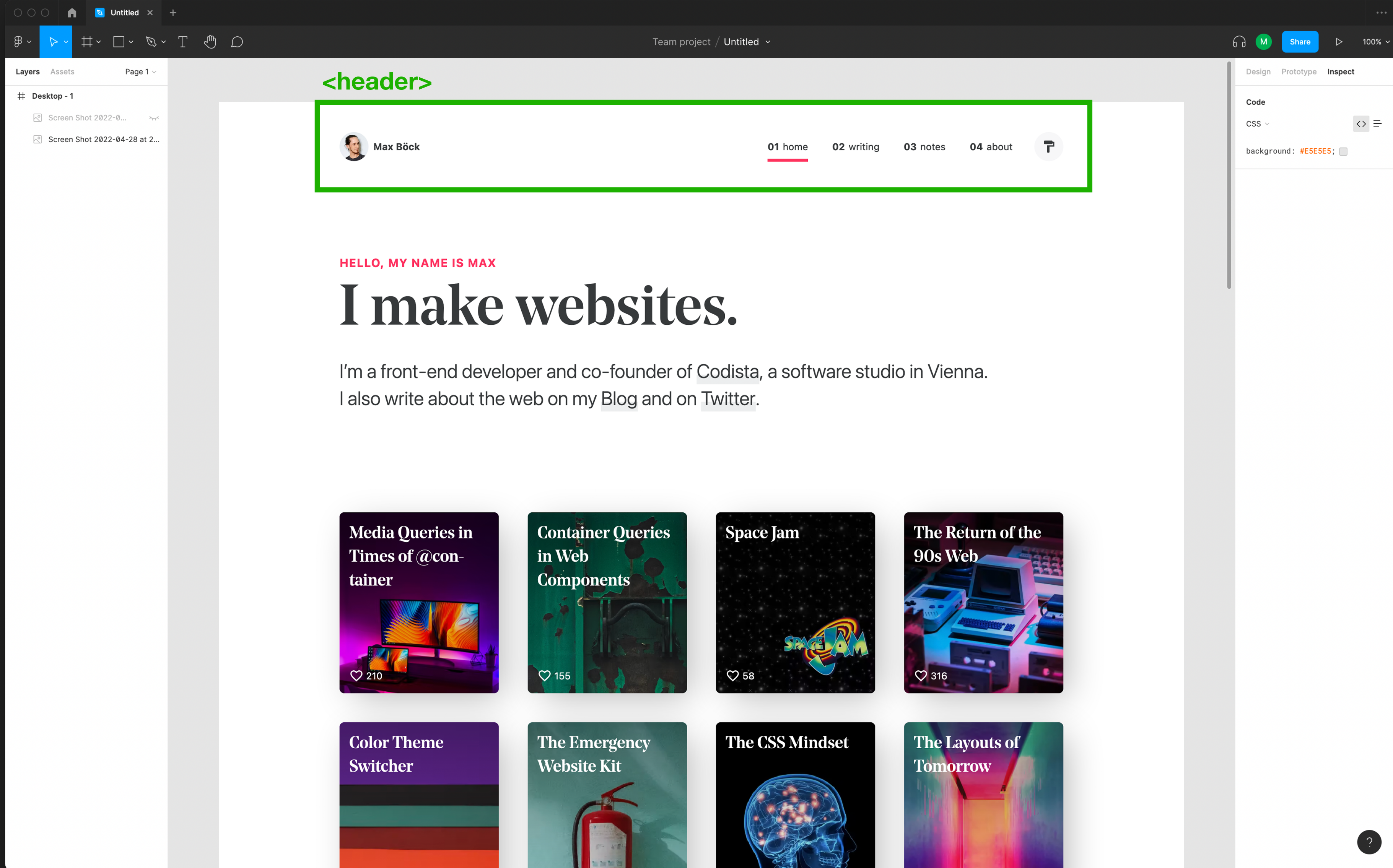


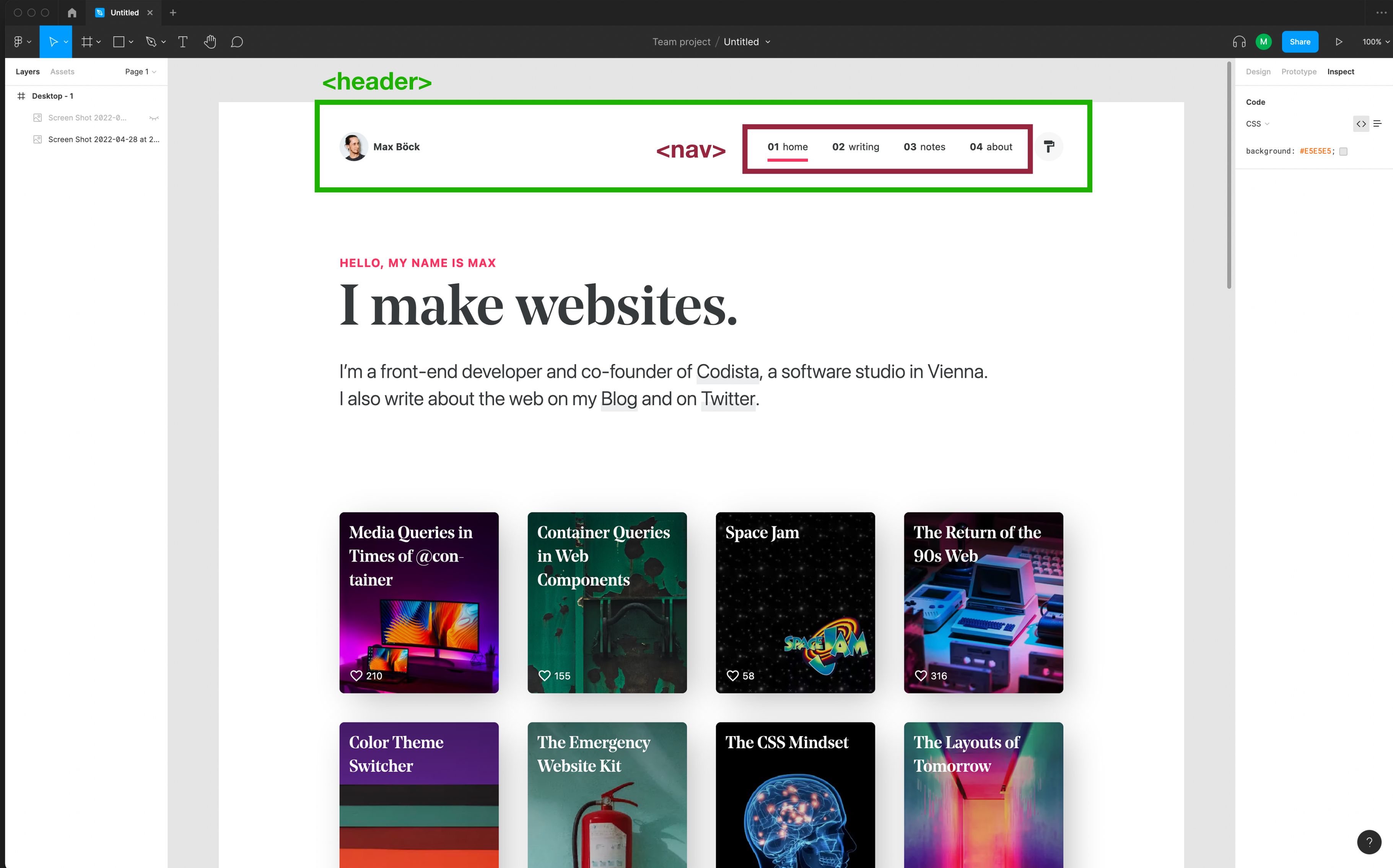


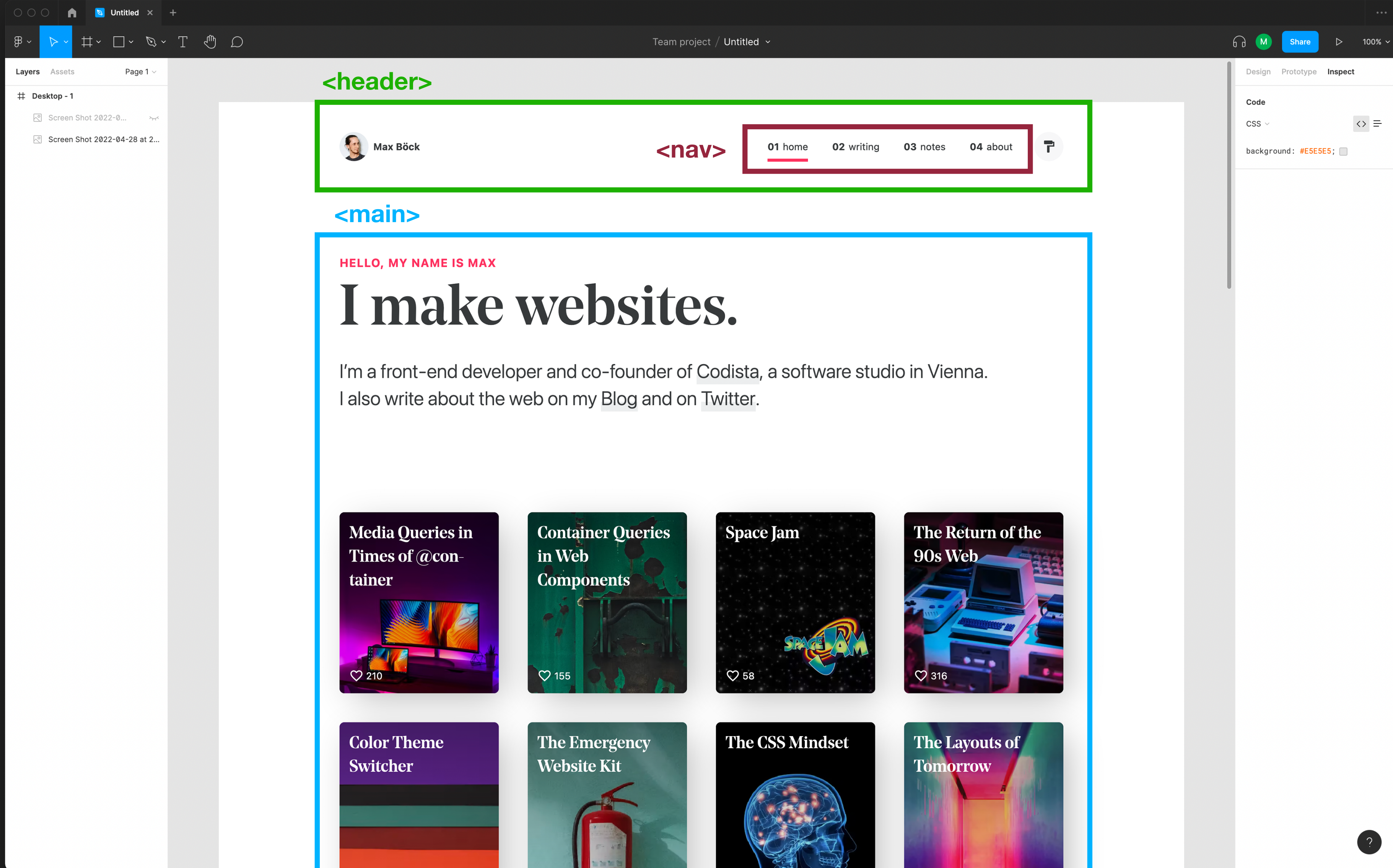
A Sound Document Outline

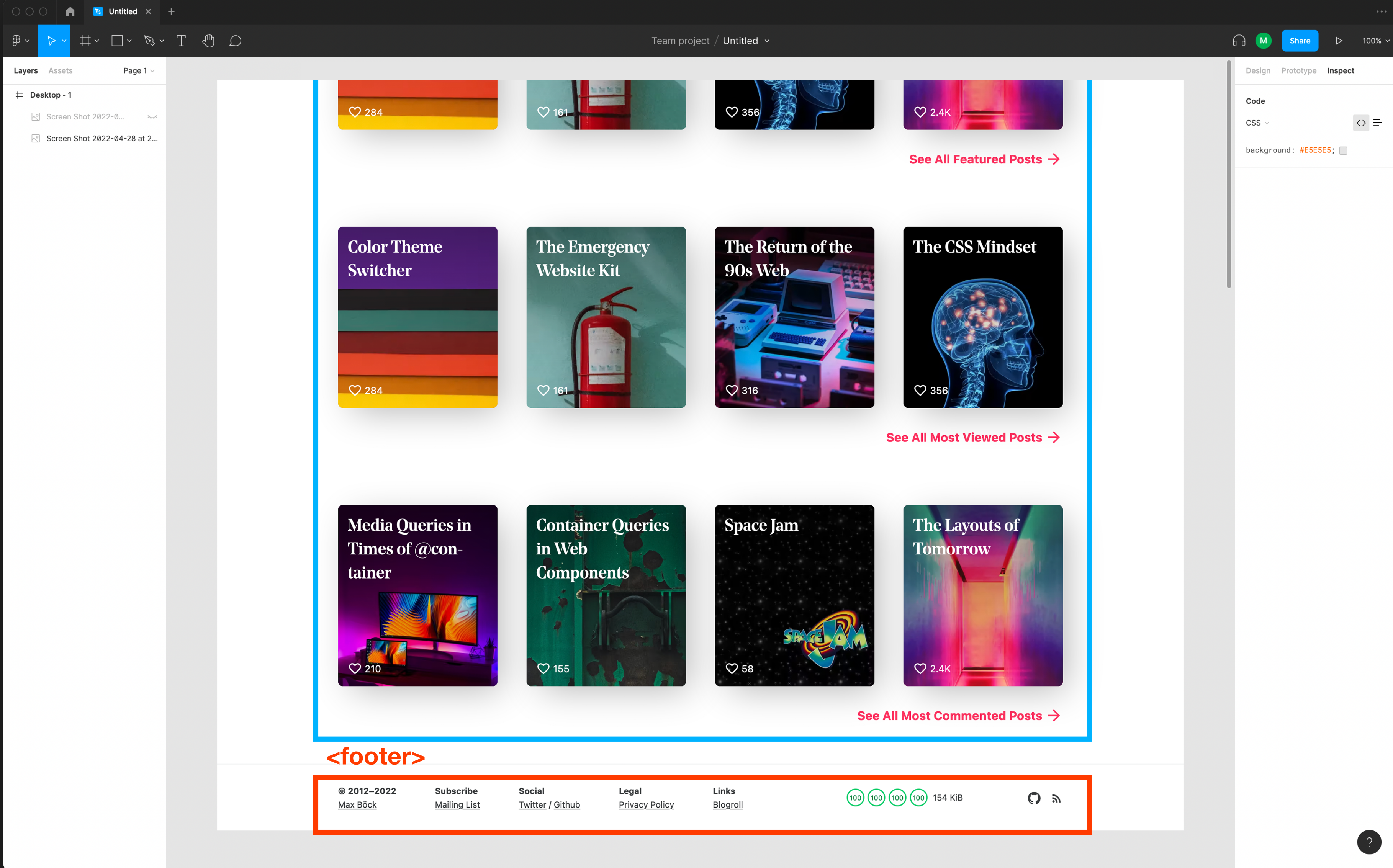
- Meaningful labels
- Clean hierarchy (Avoid skipping levels)
- Enough but not too many headings











<footer>

© 2012–2022
Max Böck

Subscribe
Mailing List

Social
Twitter / Github

Legal
Privacy Policy

Links
Blogroll

100 100 100 100 154 KiB



Voice Control Usability Considerations For Partially Visually Hidden Link Names

SMASHING MAGAZINE

Articles Books Workshops Conferences Membership Job Board Newsletter Podcasts More

Search articles...

Accessibility UX CSS JavaScript Performance Design Figma Wallpapers React Vue Round-Ups Web Design Guides Business Career Privacy Jump to all articles

Eric Bailey / JUN 24, 2022 / 1 comment

Voice Control Usability Considerations For Partially Visually Hidden Link Names

10 min read Accessibility, Usability, User Experience, Voice

Share on Twitter, LinkedIn

QUICK SUMMARY ↔ Overcorrecting for one form of disability may unintentionally negatively impact the experience for other forms of disability. For example, partially visually hidden link names may work great for people who use screen readers, but this approach can be problematic for people who rely on voice control software. Because of this, your designs need to be flexible and adaptable, as well as accommodate the many different ways people can interact with them.

Digital accessibility tends to be taught through the lens of how your experience works (or fails to work) with a screen reader. It makes sense to think that, if [it works for a screen reader](#), it will also work for a lot of other kinds of assistive technology.

However, this approach also indirectly reinforces the narrative that blindness is

ABOUT THE AUTHOR

Eric is a Boston-based designer who helps create straightforward solutions that address a person's practical, physical, cognitive, and emotional needs. [More about Eric](#)

Email Newsletter

Your email Meow!

Weekly tips on front-end & UX. Trusted by 200,000+ folks.

Source: [smashingmagazine.com/2022/06/voice-control-usability-considerations-partially-visually-hidden-link-names](https://www.smashingmagazine.com/2022/06/voice-control-usability-considerations-partially-visually-hidden-link-names)

@mmatuzo

```
<section aria-label="Quick summary" class="article__summary">
  <span id="article__start" class="summary__heading" aria-hidden="true">
    Quick summary ↗
  </span>
  Overcorrecting for one form of disability may...
</section>
```

Untitled x +

Team project / Untitled

Design Prototype Inspect

Code

CSS

background: #E9E9E9;

Layers Assets Page 1

Desktop - 1

Screen Shot 2022-0...

Screen Shot 2022-04-28 at 2...

Max Böck

01 home 02 writing 03 notes 04 about

HELLO, MY NAME IS MAX

I make websites.

I'm a front-end developer and co-founder of Codista, a software studio in Vienna.
I also write about the web on my [Blog](#) and on [Twitter](#).

Media Queries in Times of @container 210

Container Queries in Web Components 155

Space Jam 58

The Return of the 90s Web 316

Color Theme Switcher 284

The Emergency Website Kit 161

The CSS Mindset 356

The Layouts of Tomorrow 2.4K

See All Featured Posts →

Color Theme Switcher 284

The Emergency Website Kit 161

The Return of the 90s Web 316

The CSS Mindset 356

See All Most Viewed Posts →

Media Queries in Times of @container 210

Container Queries in Web Components 155

Space Jam 58

The Layouts of Tomorrow 2.4K

See All Most Commented Posts →

© 2012-2022 Max Böck

Subscribe [Mailbox List](#)

Social [Twitter](#) / [Github](#)

Legal [Privacy Policy](#)

Links [Biosroll](#)

154 KIB

?

```
<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Ein div ist keine Schaltfläche</title>
</head>
<body>
  <h1>Ein div ist keine Schaltfläche</h1>

</body>
</html>
```



```
<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Ein div ist keine Schaltfläche</title>
</head>
<body>
  <h1>Ein div ist keine Schaltfläche</h1>

</body>
</html>
```

The lang Attribute

- Screen readers
- Hyphenation
- Quotation marks
- Spell checking
- The default font selection for CJK languages
- Translation

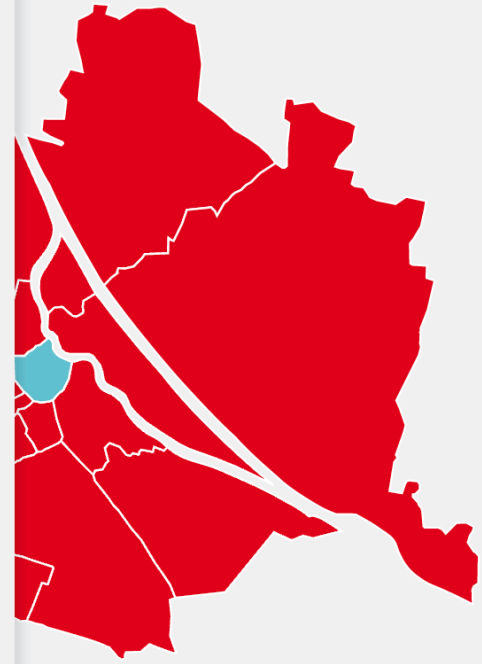
news ORF.at WAHL20 WIEN

Fernsehen TVthek Radiothek Debatte Österreich Wetter Sport News ORF.at im Überblick

Ergebnis Gemeinderat Ergebnisse Ergebniskarten Motive Wählerströme

Wien ▾

- 1., Innere Stadt
- 2., Leopoldstadt
- 3., Landstraße
- 4., Wieden
- 5., Margareten
- 6., Mariahilf
- 7., Neubau
- 8., Josefstadt
- 9., Alsergrund
- 10., Favoriten
- 11., Sieden
- 12., Meidling
- 13., Hietzing
- 14., Penzing
- 15., Rudolfsheim-Fünfhaus
- 16., Ottakring
- 17., Hernals
- 18., Währing
- 19., Döbling
- 20., Brigittenau
- 21., Floridsdorf
- 22., Donaustadt
- 23., Lügen




inkl. Briefwahl

ahl und der Bezirksvertretungswahlen sind inklusive etroffen.

Partei	Stimmen	Prozent	Veränderung
SPÖ	301.967 Stimmen	30,11 %	+2,03 %
FPÖ	51.603 Stimmen	7,11 %	-23,67 %
GRÜNE	107.397 Stimmen	14,80 %	+2,96 %
ÖVP	148.238 Stimmen	20,43 %	+11,19 %
NEOS	54.173 Stimmen	7,47 %	+1,31 %
HC	23.688 Stimmen	3,27 %	+3,27 %
LINKS	14.919 Stimmen	2,06 %	+2,06 %

The background is a dark, stylized illustration of a city street. It features buildings with various signs and advertisements, including 'docomo', 'THEグローバル社', and 'ECC'. A road sign in the bottom left corner indicates '初台 首都高 Matsudai SHUTOKO' and '新宿駅西口 Shinjuku Sta.' with route numbers '20' and '414'. The overall scene is rendered in a muted, dark color palette with some highlights on the buildings and signs.

**Developers,
prioritise HTML!**

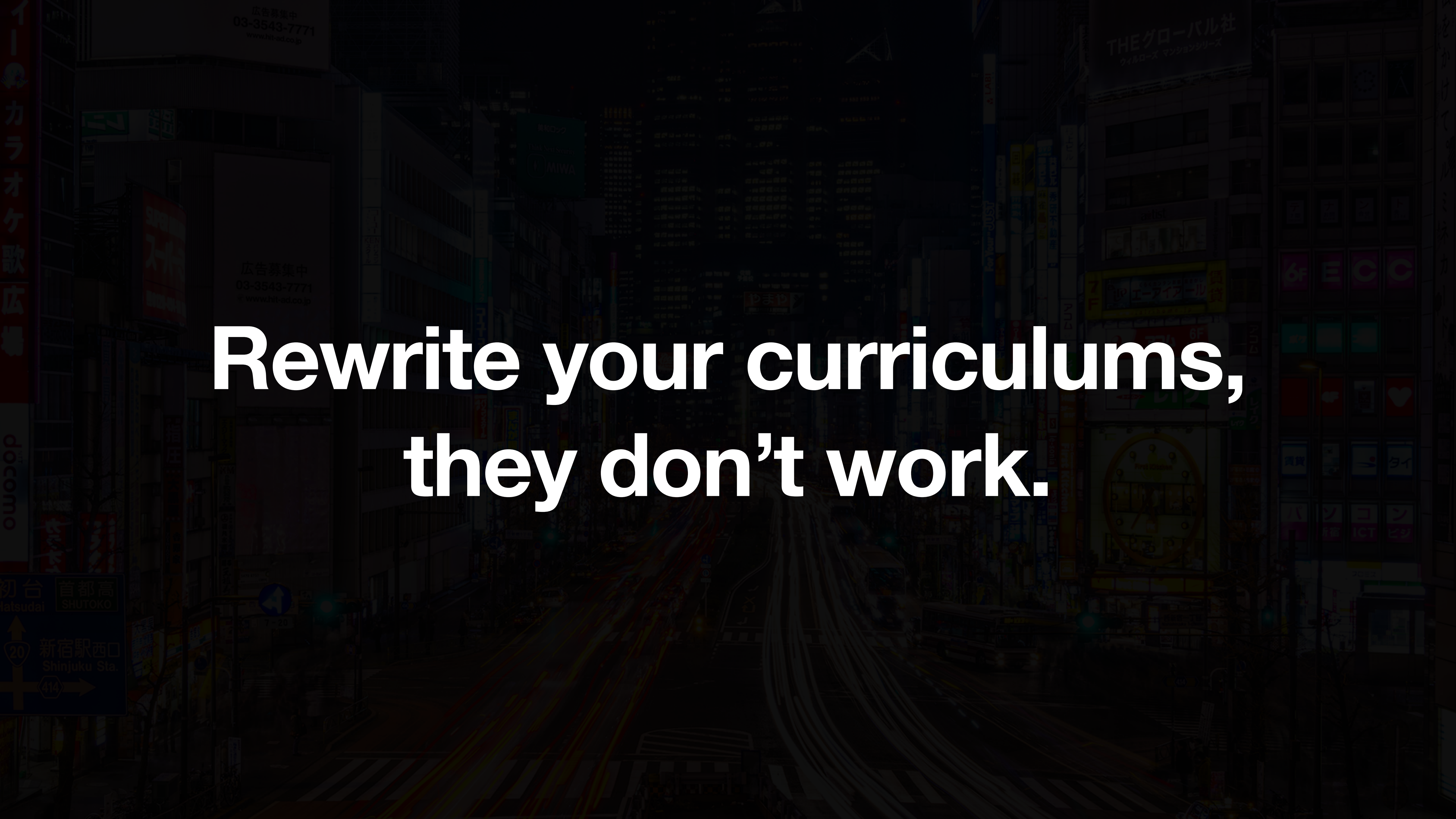


**Learn who your users are and
how they're using the web.**



Stop belittling HTML.

**Universities, schools, teachers,
course coordinators,
prioritise HTML!**

The background is a dark, stylized illustration of a city street. It features various signs and buildings. On the left, there's a vertical sign for 'docomo'. In the center, there's a large sign for 'MIWA'. On the right, there's a sign for 'THE グローバル社' and another for 'ECC'. At the bottom left, there's a road sign for '新宿駅西口 Shinjuku Sta.' with a '414' marker. The overall scene is rendered in a dark, almost black color palette with some highlights in blue and yellow.

**Rewrite your curriculums,
they don't work.**



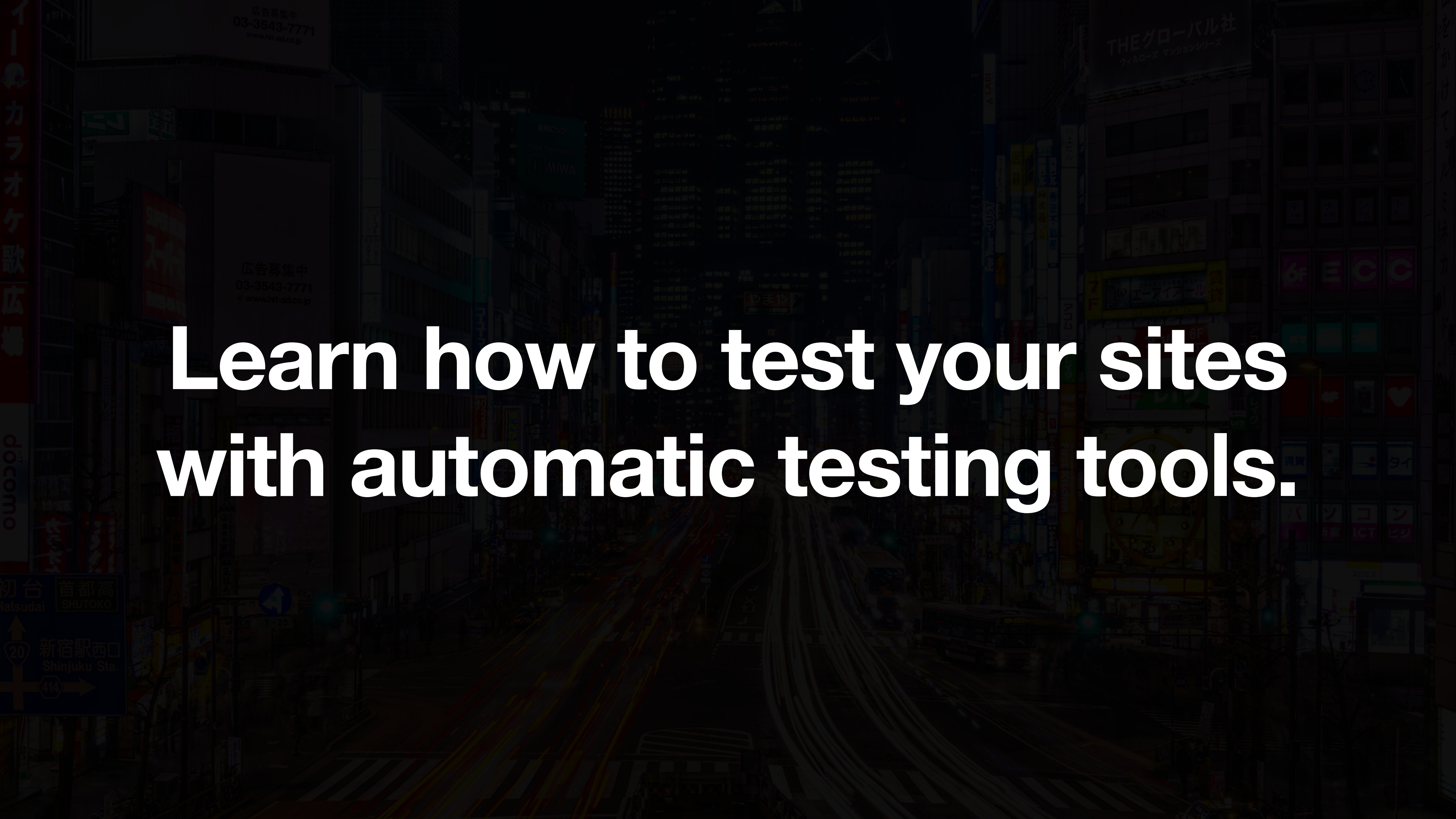
**Authors,
prioritise HTML!**

The background is a dark, stylized illustration of a city street. It features buildings with various signs and advertisements, including one for 'docomo' and another for 'THE グローバル社'. A road sign in the bottom left corner indicates '初台 首都高 Matsudai SHUTOKO' and '新宿駅西口 Shinjuku Sta.' with a '414' marker. The overall scene is rendered in a muted, dark color palette with some highlights on the buildings and signs.

**Write and speak more
about HTML.**



Use accessible HTML in your demos.



**Learn how to test your sites
with automatic testing tools.**

***96.8% of home pages had detected
WCAG 2 failures!***



**Learn how to test your sites
with the keyboard and screen
readers.**







Merci!

Additional Resources

#20 HTMHell special: close butt X +

https://www.htmhell.dev/20-close-buttons/

Hell Heaven Newsletter 



#20 HTMHell special: close buttons

submitted on May 23, 2020 by [Manuel](#)

This first HTMHell special inspects one of the most complicated and most controversial patterns in front-end development:

🔥 the close button. 🔥

In modals, ads, and other overlays you often find a button with a close symbol that allows users, or at least some of them, to close the overlay. This functionality is often limited to mouse users, because most implementations of close buttons suck.

After less than 2 hours of research, HTMHell presents a collection of 11 different bad practices.


Pattern 1: div and background image


```
<div class="close"></div>
```

```
.close::after {  
  background: url("close.png");  
  content: "";  
}
```

#11 The trigram for heaven - HTX +

https://www.htmhell.dev/11-the-trigram-for-heaven/

Hell Heaven Newsletter 








#11 The trigram for heaven

submitted on Nov 29, 2019 by [Manuel](#)

Bad code

```
<span class="nav-toggle"> ≡ Menu </span>
```

Issues and how to fix them

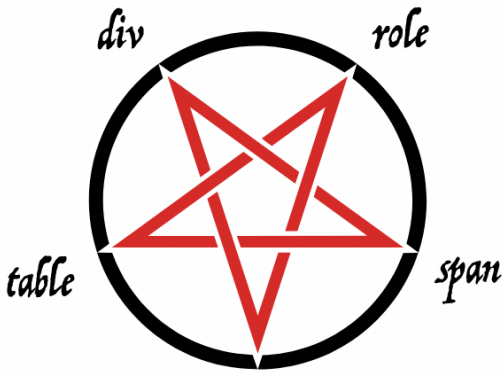
-  A screen reader may announce this as *trigram for heaven menu*, because ≡ is the unicode character for the [trigram for heaven](#).
-  The purpose of the icon is decorative, it should be hidden from screen readers. Consider adding decorative images using background properties in CSS.
-  A click event on a `span` triggers only on click. A click event on a `button` triggers on click and if the user presses the `Enter` or `Space` key.
-  A `span` isn't keyboard focusable, but this element must be focusable, because it's used for opening and closing the main navigation.
-  `aria-expanded` must be added to indicate whether the main navigation is

Source: [htmhell.dev/11-the-trigram-for-heaven](https://www.htmhell.dev/11-the-trigram-for-heaven)

#26 HTMHell special: tasty butt... X

https://www.htmhell.dev/26-tasty-buttons/

Hell Heaven Newsletter



HTMHell

#26 HTMHell special: tasty buttons

submitted on Oct 25, 2021 by [Manuel](#)

The second HTMHell special focuses on another highly controversial pattern in front-end development:

🔥 the burger button. 🔥

The burger button and his tasty friends (kebab, meatball and bento) usually reveal a list of links when activated. According to our studies, these buttons are highly optimized for mouse and touch users, but lack support for keyboard and screen reader users in most cases.

After less than 1 hours of research, HTMHell presents a collection of 18 different bad practices found on real websites.

Pattern 1: the unsemantic burger

```
<div class="burger">
  <span></span>
  <span></span>
  <span></span>
</div>
```

