

Webtechnologies

JAVASCRIPT, THE DOM, OBJECTS, FUNCTIONS, DATA FETCHING



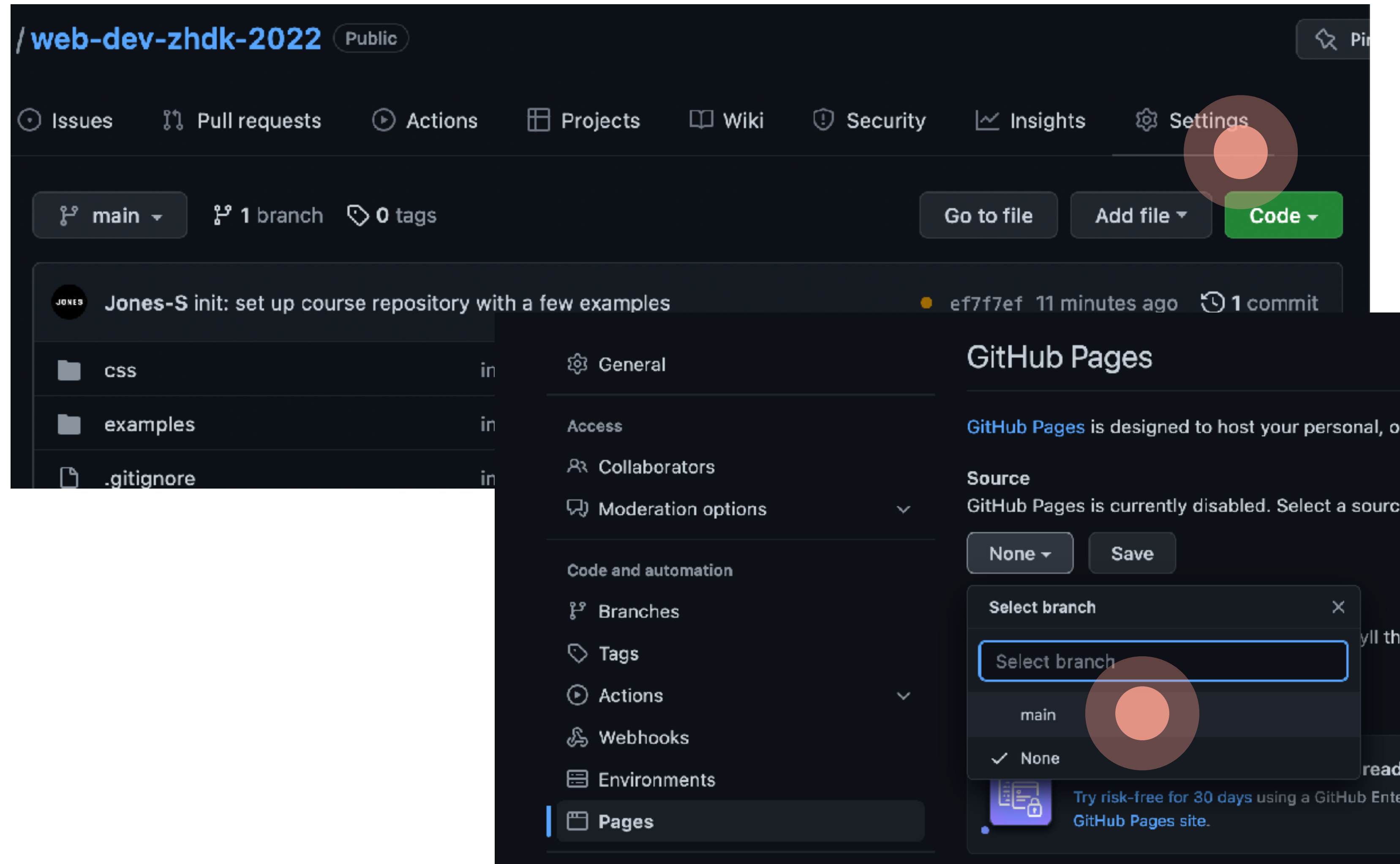
Q&A → QUESTIONS ABOUT YESTERDAY?

- Questions
- 12 cols, prompting
- Pos absolute correction

GITHUB PAGES

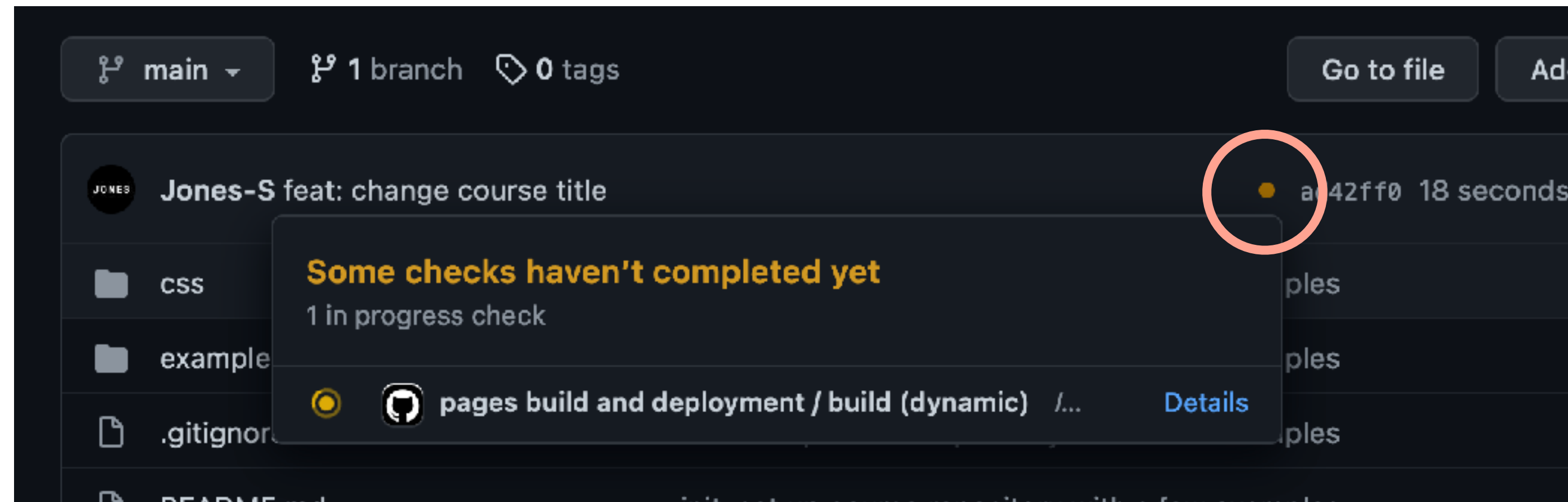
- Open your Github Repository
→ Settings
→ Pages

Select source branch (main)
and folder. Done.

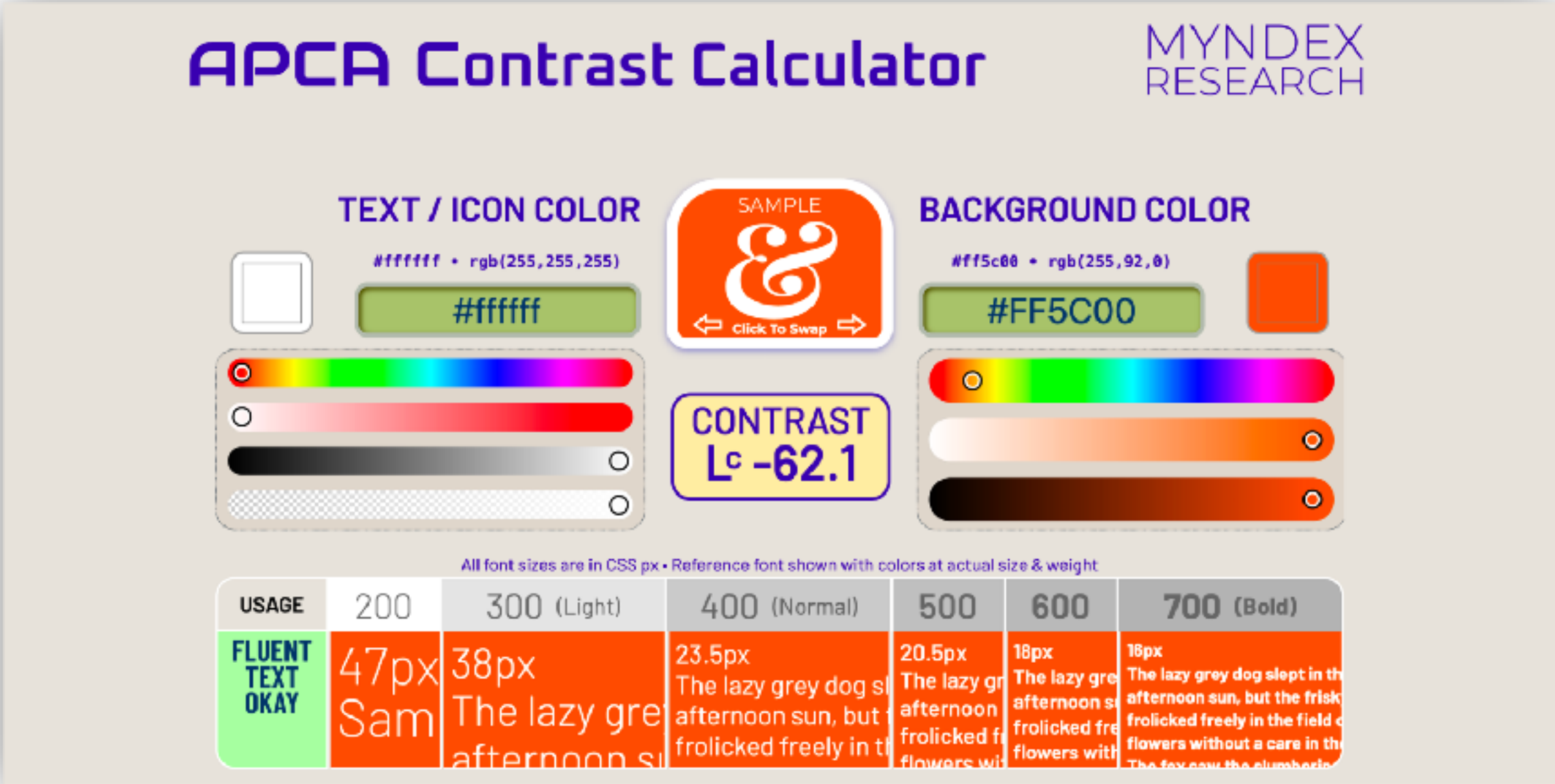
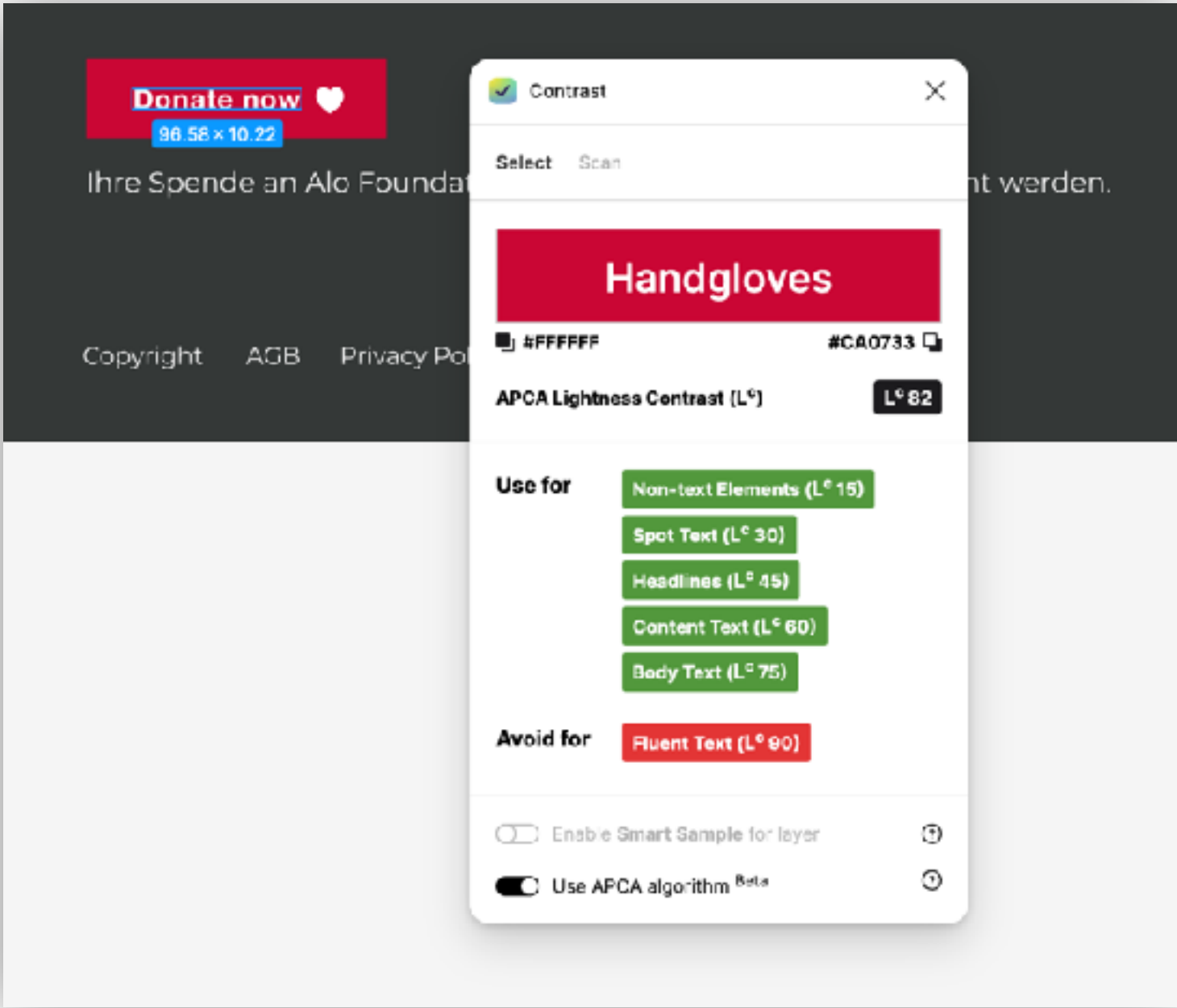


HOSTING A WEBSITE ON GITHUB

- Site still not online?
- Site is deployed on your next “push”.
- The build might take a bit of time, but after completion you can access your page at:
`https://USER-NAME.github.io/REPO-NAME/`



ACCESSIBILITY (A11Y)



ACCESSIBILITY (A11Y)



Accessible

Favorite Snack

Favorite Snack



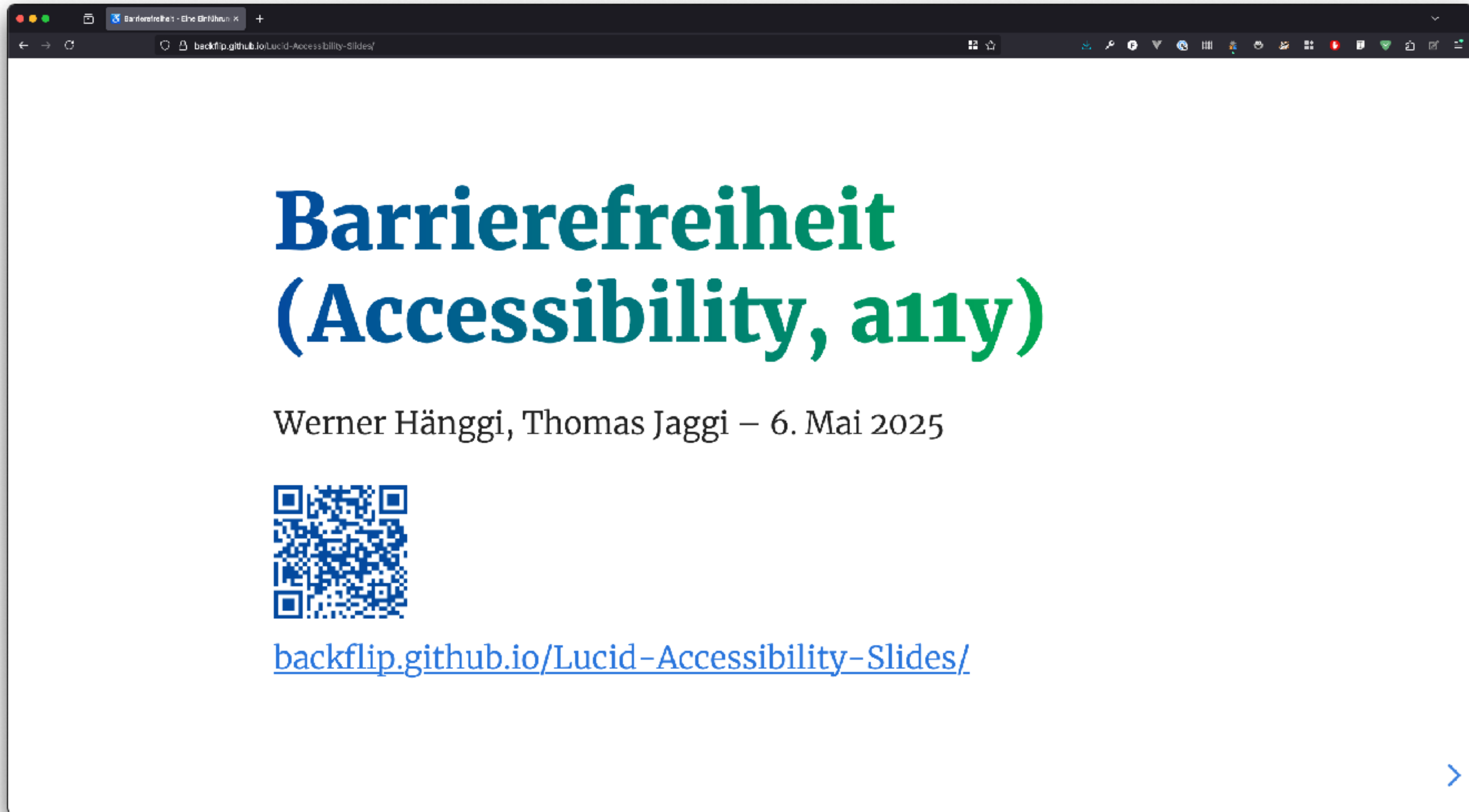
Not Accessible

What's your favorite snack?

ACCESSIBILITY (A11Y)



ACCESSIBILITY (A11Y)



The image is a screenshot of a web browser window. The browser's address bar shows the URL `backflip.github.io/Lucid-Accessibility-Slides/`. The slide content features a large title in blue and green, followed by the authors' names and a date. A QR code is positioned below the text, and a blue link is at the bottom. A blue arrow icon is in the bottom right corner of the slide.

Barrierefreiheit (Accessibility, a11y)

Werner Hänggi, Thomas Jaggi – 6. Mai 2025



backflip.github.io/Lucid-Accessibility-Slides/

ACCESSIBILITY (A11Y)

Weltweit: 🍌

Auszug aus [Auswertung des HTTP Archive für 2024](#) (17 Mio Webseiten):

- **2/3** der Seiten haben **ungenügenden Farbkontrast**.
- **1/4** der Seiten versucht, Zooming auf Mobile zu deaktivieren.
- **1/3** der Bilder haben **keinen Alternativtext**.
- **2/3** der Seiten haben **Formularfelder ohne Labels**.



In der Schweiz: 🇨🇭

Auszug aus [Schweizer Accessibility-Studie Onlineshops \(2020\)](#):

- **Hilfestellung** bei Interaktionen: 36 von 41 Shops sind in diesem Bereich ungenügend.
- Alle 11 Onlineshops, die [...] **Multimediainhalte** [wie Videos] einsetzen, erfüllen diese Anforderungen in keiner Weise.
- Mehr als die Hälfte der getesteten Onlineshops weist Verbesserungsbedarf im Bereich **Nicht-Textinhalte** auf.
- 20 Onlineshops, die nur 1 von 5 möglichen Punkten im Bereich **Syntax/Kompatibilität** erzielen.
- Die **Tastaturbedienbarkeit** lässt in vielen Onlineshops ebenfalls zu wünschen übrig.



c) Interaktive Elemente kommunizieren keine Zustandsänderung (1)

Probleme

- **Formularvalidierung** wird nicht kommuniziert.
- Es werden **nicht-native Formularelemente** verwendet.

Folge

Screenreader kann Formular nicht korrekt ausfüllen.

Lösung

- Verwendung von nativer Validierung.
- Verwendung von nativen Formularelementen.



ACCESSIBILITY (A11Y)

Accessibility

cas-fe.github.io/Accessibility

INACCESSIBLE FORM

Required fields are bold.

Name

Error: Field must not be empty

E-mail

Error: Field must not be empty

Date

2025-06-04

?

Land

Schweiz

☐ AGB gelesen

Submit

Show issues

« Back to "Accessible images and icons"

Index

Forward to "Accessible form" »

ACCESSIBILITY (A11Y)

EN



Buy EPUB/PDF







The Modern JavaScript Tutorial

Last updated on June 3, 2025

How it's done now. From the basics to advanced topics with simple, but detailed explanations.

Buy EPUB/PDF

Share:

Search in the tutorial

Search

24394  github

 Discord Chat

Table of contents

Main course contains 2 parts which cover JavaScript as a programming language and working with a browser. There are also additional series of thematic articles.

PART 1
The JavaScript language

PART 2
Browser: Document, Events, Interfaces

PART 3
Additional articles

The JavaScript language

Here we learn JavaScript, starting from scratch and go on to advanced concepts like OOP. We concentrate on the language itself here, with the minimum of environment-specific notes.

An introduction

1.1 An Introduction to JavaScript

1.3 Code editors

Inspector

Console

Debugger

Network

Style Editor

Accessibility

Check for issues: None

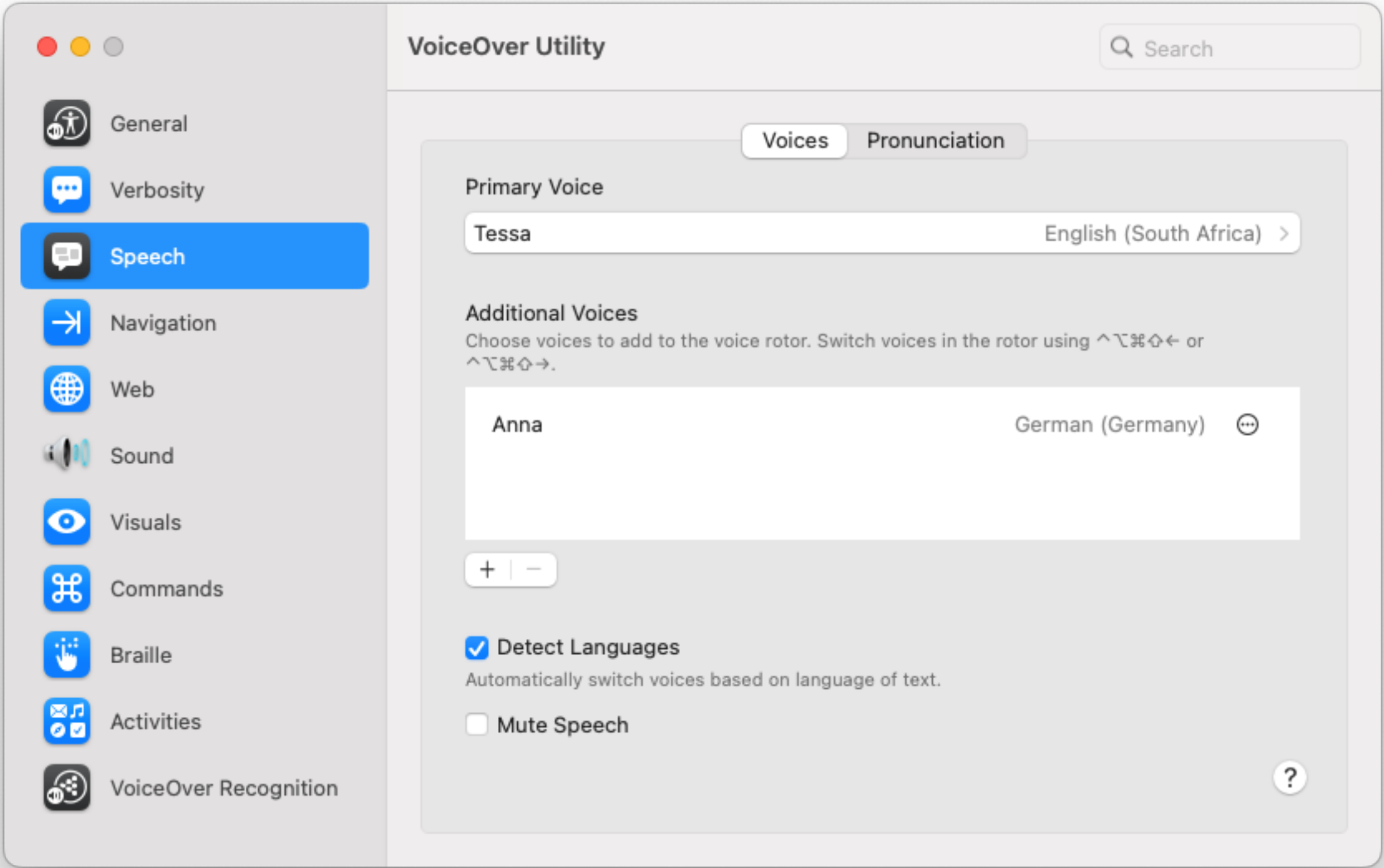
Simulate:

Achromatopsia (no color)

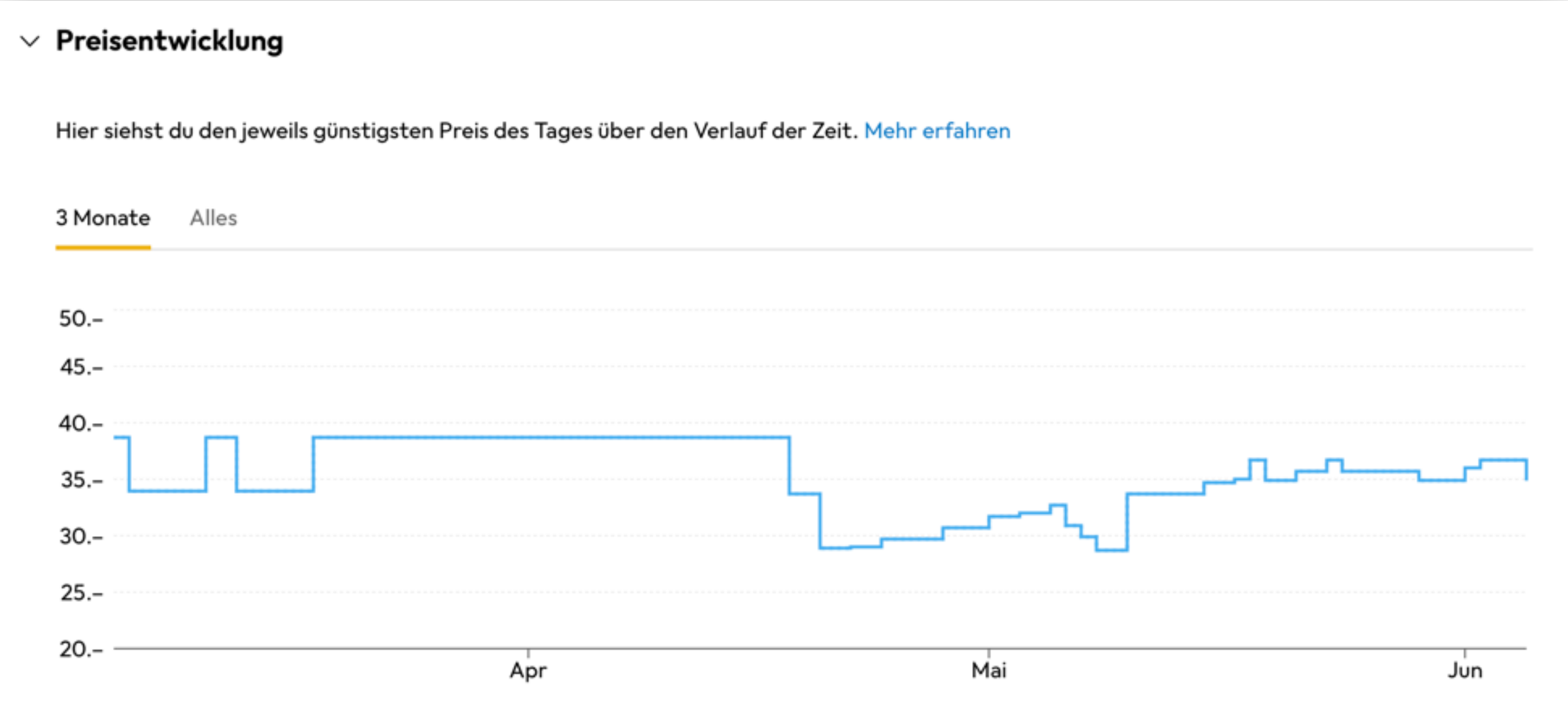
Show Tabbing Order

Role	Name	Checks
▶ document:	"The Modern JavaScript Tutorial"	No checks for this node.

ACCESSIBILITY (A11Y)

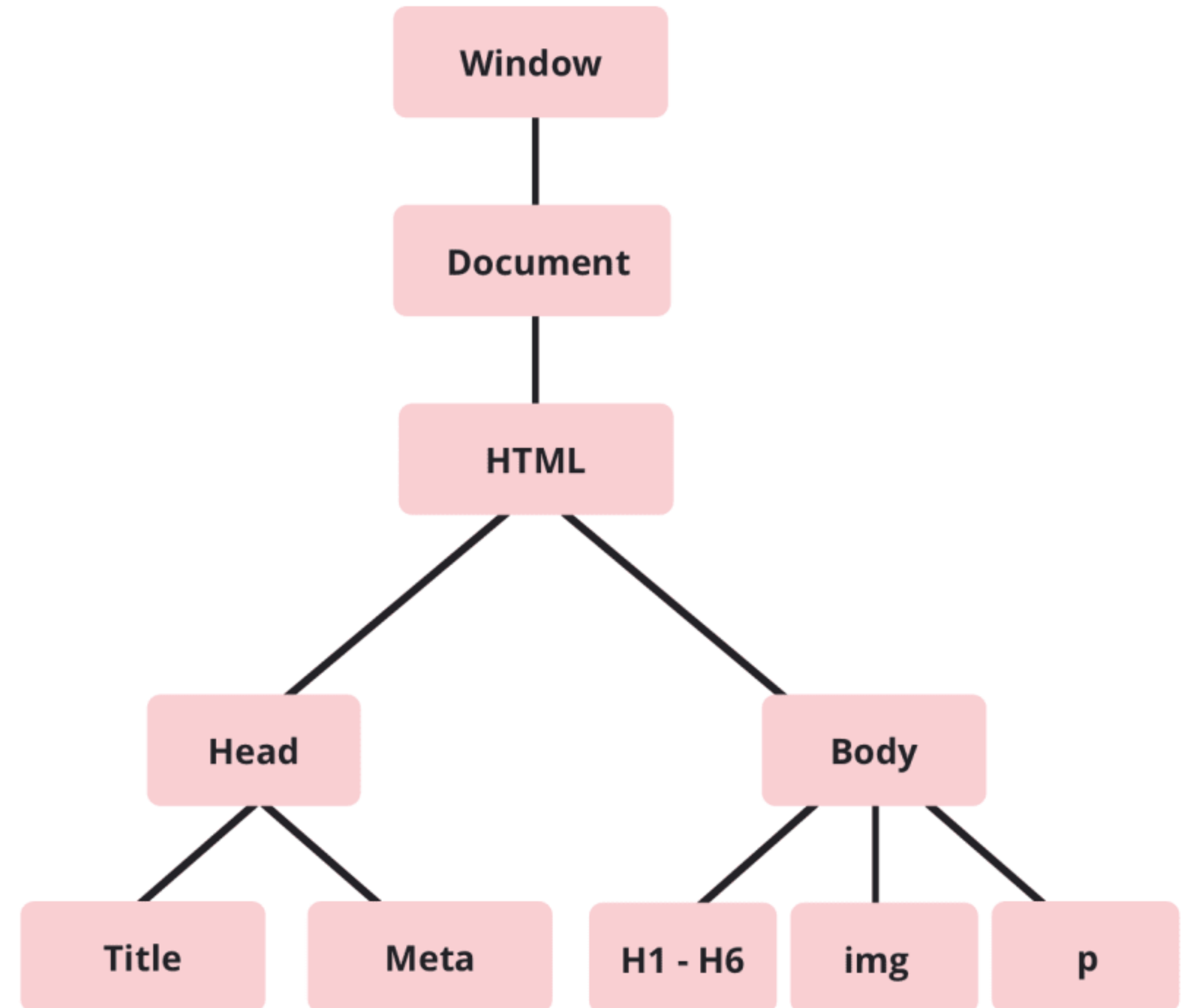


ACCESSIBILITY (A11Y)



JAVASCRIPT

- High-Level programming language
- Mostly in the browser (also on the server (Node.js, Deno.js))
- We can use it for DOM-Manipulation
- But basics first...



Document Object Model (DOM) Tree

1. JAVASCRIPT-FILE

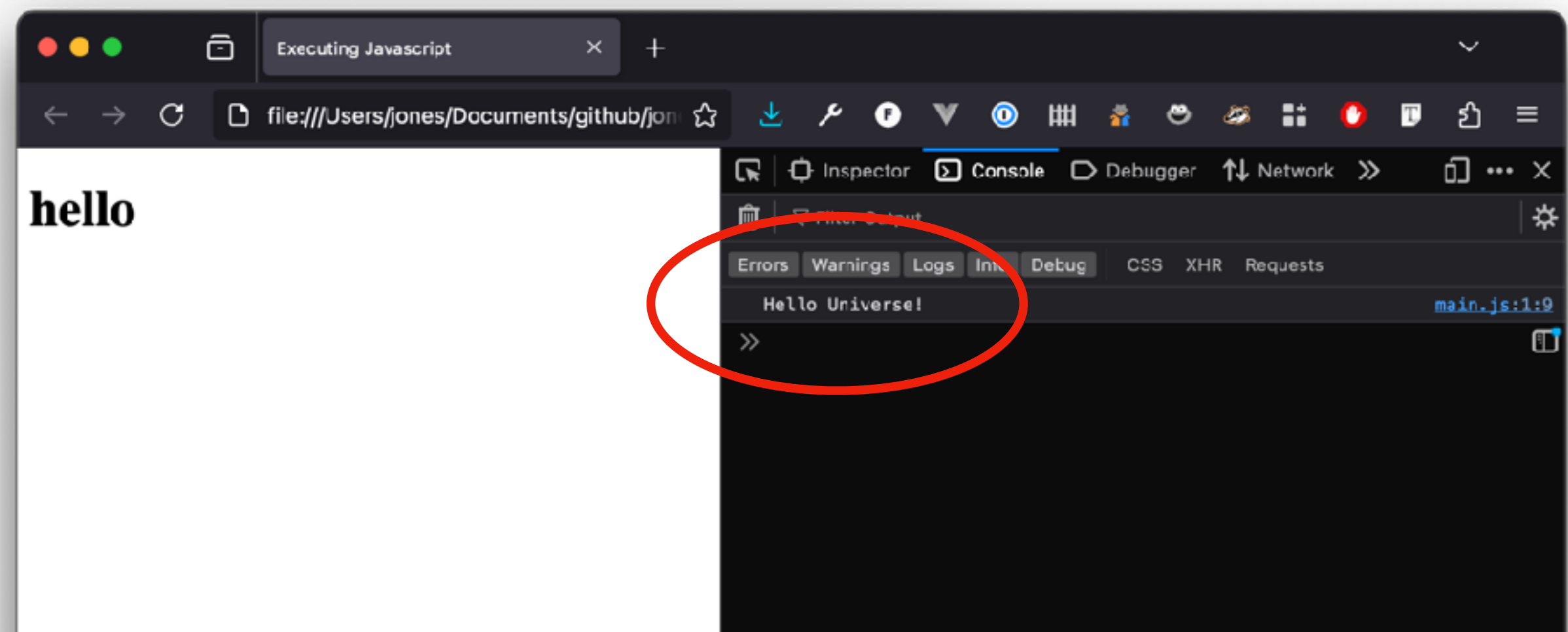
- To try out JavaScript we need to create a JavaScript file and execute it in the browser.

index.html

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>Executing Javascript</title>
7   <link href="style.css" rel="stylesheet" type="text/css">
8   <script src="main.js" defer></script>
9 </head>
10
11 <body>
```

main.js

```
1 console.log("Hello Universe!");
2
```





```
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <meta charset="UTF-8">
6      <title>Executing Javascript</title>
7      <link href="style.css" rel="stylesheet" type="text/css">
8      <script src="main.js" defer></script>
9  </head>
10
11 <body>
```

EXERCISE 11

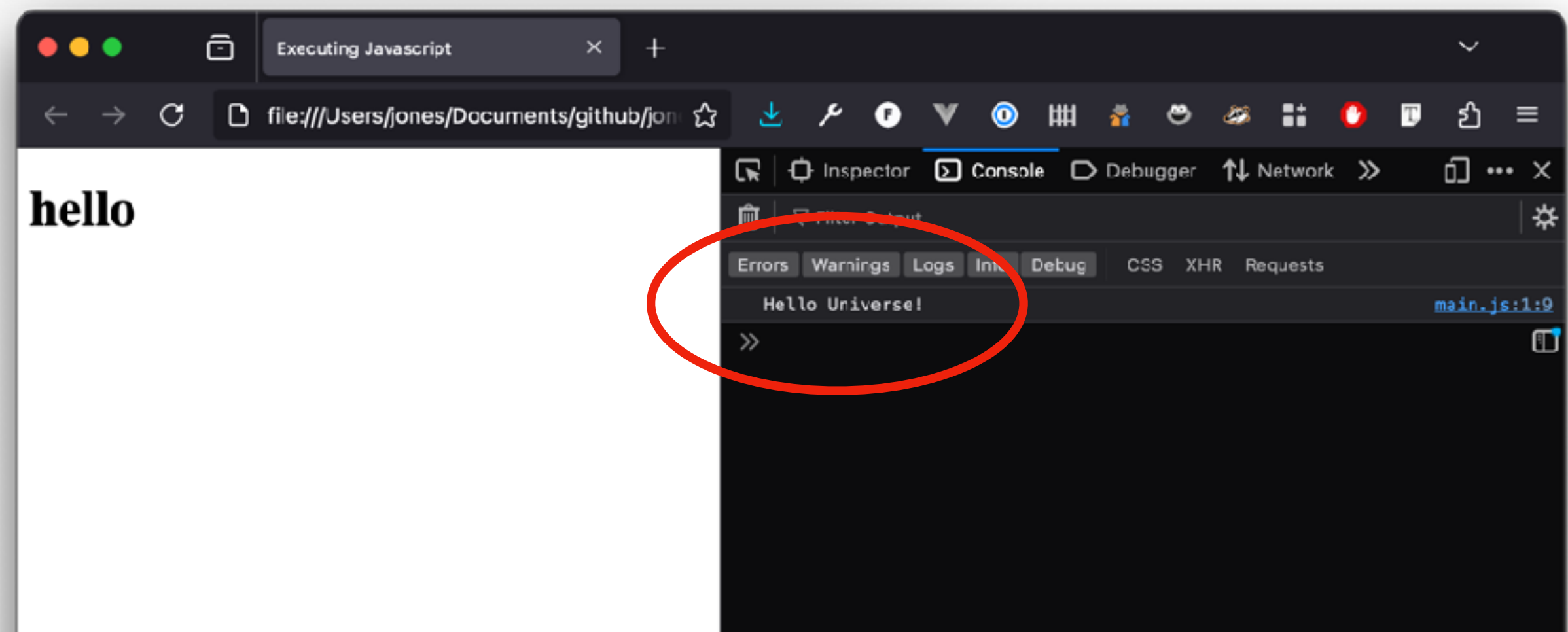
- Create a new html and a JavaScript file (*main.js*) and link them together
- Add a console log in the JS file and find it in the console of the web browser.

index.html

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>Executing Javascript</title>
7   <link href="style.css" rel="stylesheet" type="text/css">
8   <script src="main.js" defer></script>
9 </head>
10
11 <body>
```

main.js

```
1 console.log("Hello Universe!");
2
```



ARRAYS



```
1  const ingredients = ["Tomatoes", "Onions", "Garlic", "Olive oil", "Salt"];
```

ARRAYS — EXERCISE 12

- Solve little array exercises
- Download files from:
<https://short.jonasscheiwiller.ch/ex12>

```
1 // =====
2 // EXERCISE: Arrays & Loops
3 // =====
4 //
5 // Open the browser console to see your results.
6 // (Right-click the page → Inspect → Console)
7 //
8 // =====
9
10 const ingredients = ["Tomatoes", "Onions", "Garlic", "Olive oil", "Salt"];
11
12 // TODO 1: log the entire array
13 console.log(/* ??? */);
14
15 // TODO 2: log just the first ingredient
16 console.log(/* ??? */);
17
18 // TODO 3: log how many ingredients there are
19 console.log(/* ??? */);
20
21 // TODO 4: loop over the array and log each ingredient one by one
22 /* ??? */
23
```

OBJECTS



```
1  const recipe = {
2    title: "Tomato Soup",
3    duration: 30,
4    vegetarian: true,
5    author: {
6      name: "Maria",
7      country: "Brasil",
8    },
9    ingredients: ["Tomatoes", "Onions", "Garlic", "Olive oil", "Salt"],
10  };
```

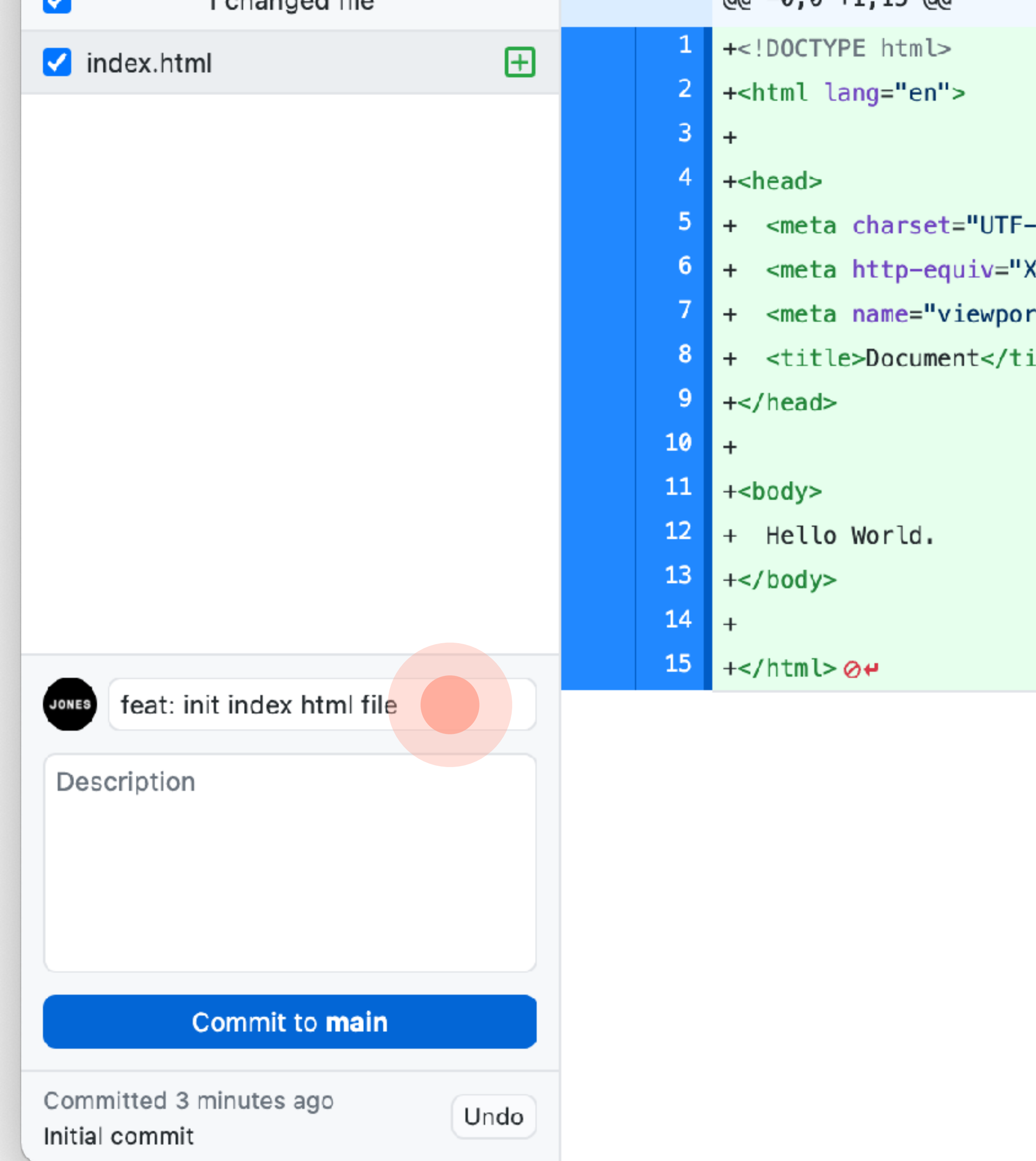
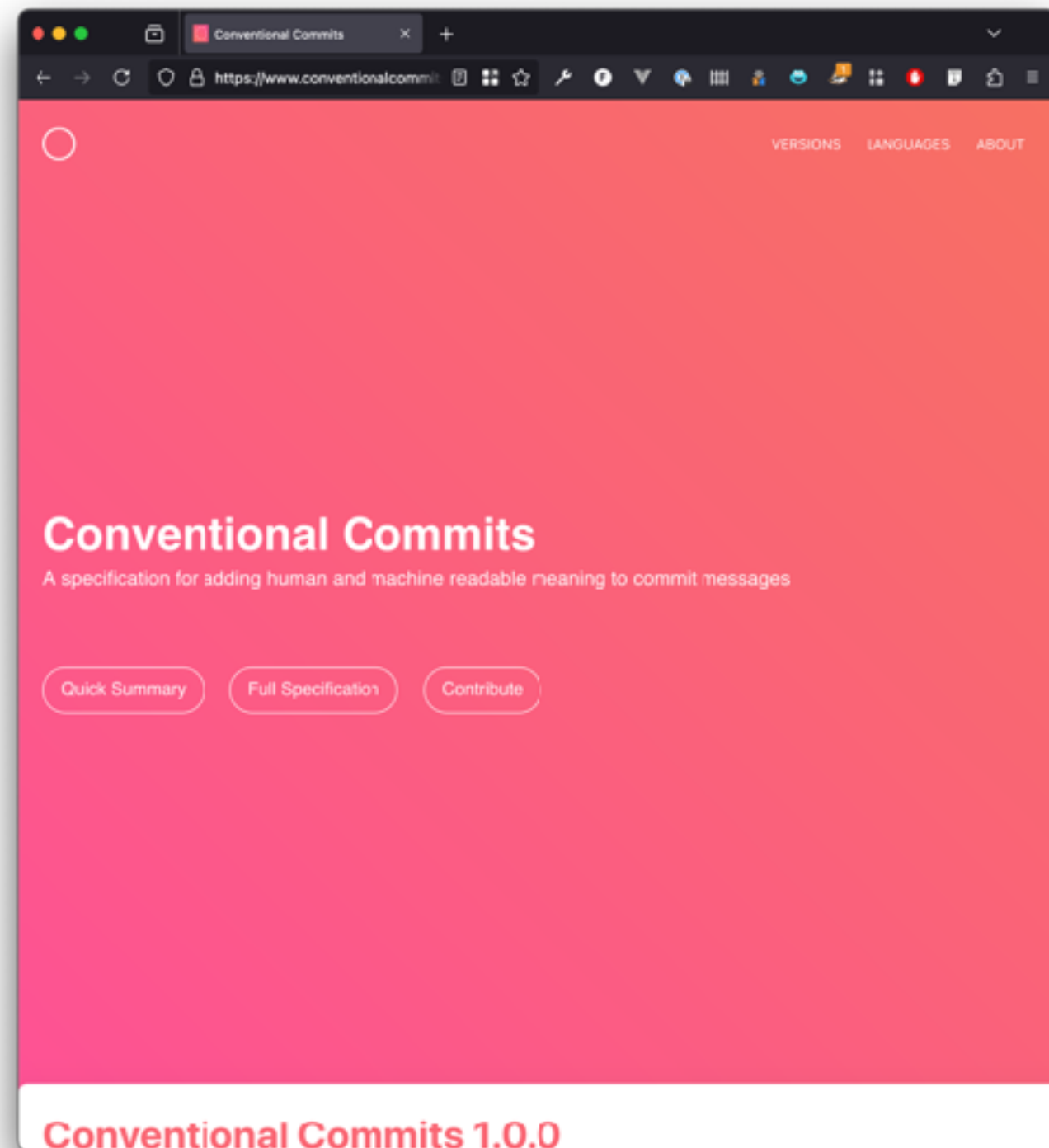
OBJECTS — EXERCISE 13

- Solve little object exercises
- Download files from:
<https://short.jonasscheiwiller.ch/ex13>

```
1 // =====
2 // EXERCISE: Objects
3 // =====
4 //
5 // Open the browser console to see your results.
6 // (Right-click the page → Inspect → Console)
7 //
8 // =====
9
10 const recipe = {
11   title: "Tomato Soup",
12   duration: 30,
13   vegetarian: true,
14   author: {
15     name: "Maria",
16     country: "Brasil",
17   },
18   ingredients: ["Tomatoes", "Onions", "Garlic", "Olive oil", "Salt"],
19 };
20
21 // TODO 1: log the entire object
22 console.log(/* ??? */);
23
24 // TODO 2: log just the title
25 console.log(/* ??? */);
26
27 // TODO 3: log the author's name
28 console.log(/* ??? */);
29
30 // TODO 4: log how many ingredients the recipe has
31 console.log(/* ??? */);
32
33 // TODO 5: loop over the ingredients and log each one
34 /* ??? */
35
```

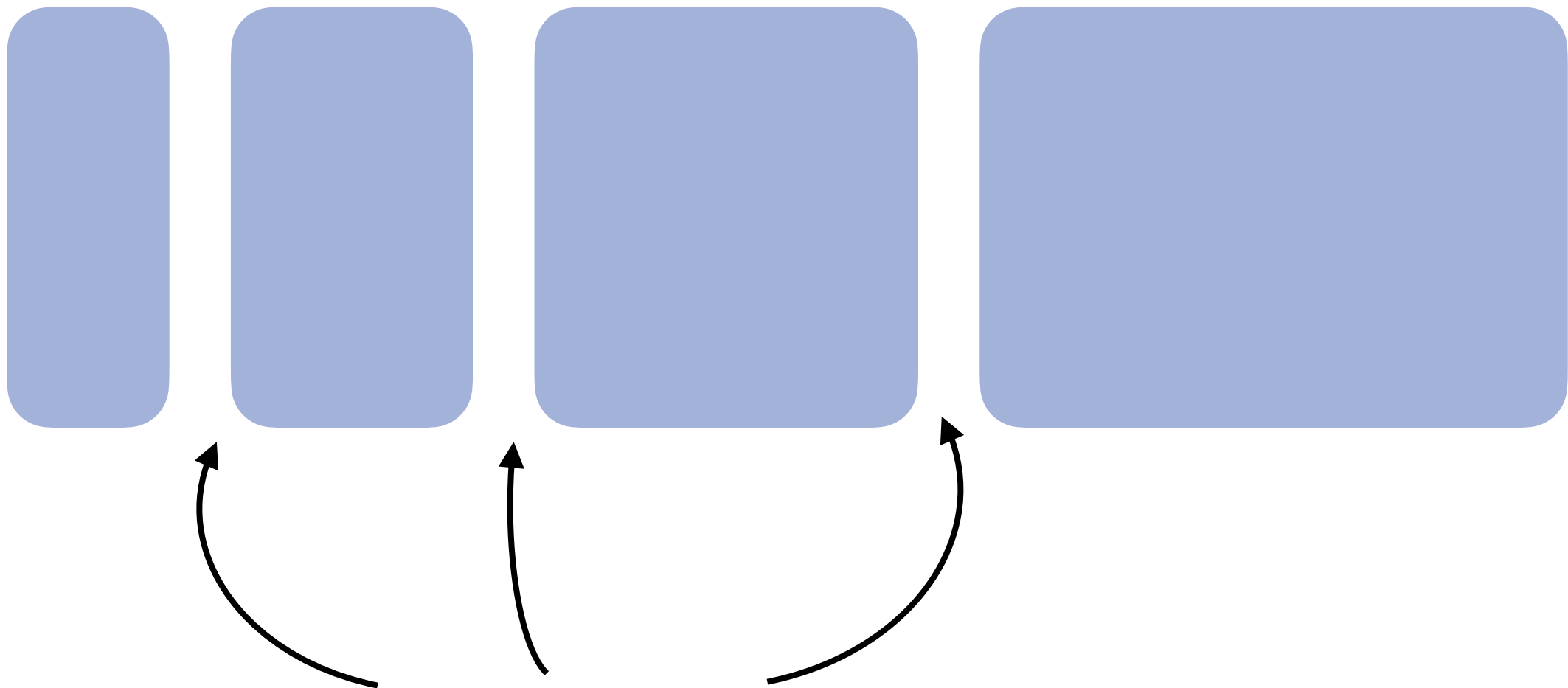
COMMIT MESSAGE

- Describe your changes and commit them.
 - ⚠ Description is rarely used
- You might want to start using some commit guidelines:
<https://www.conventionalcommits.org/en/v1.0.0/>



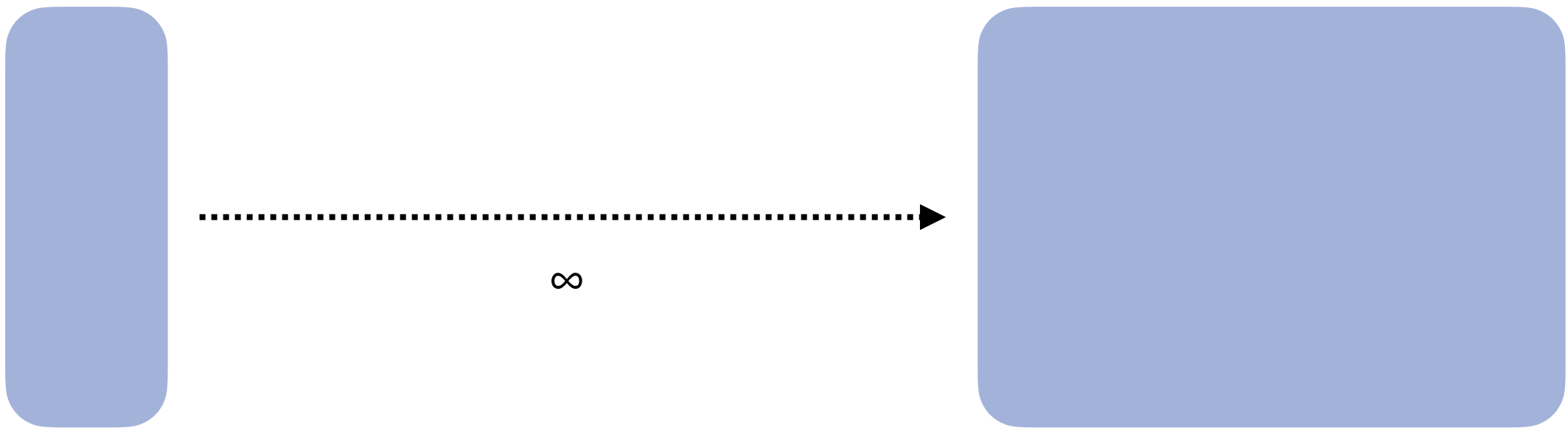
RESPONSIVE LAYOUTS

Adaptive Layout



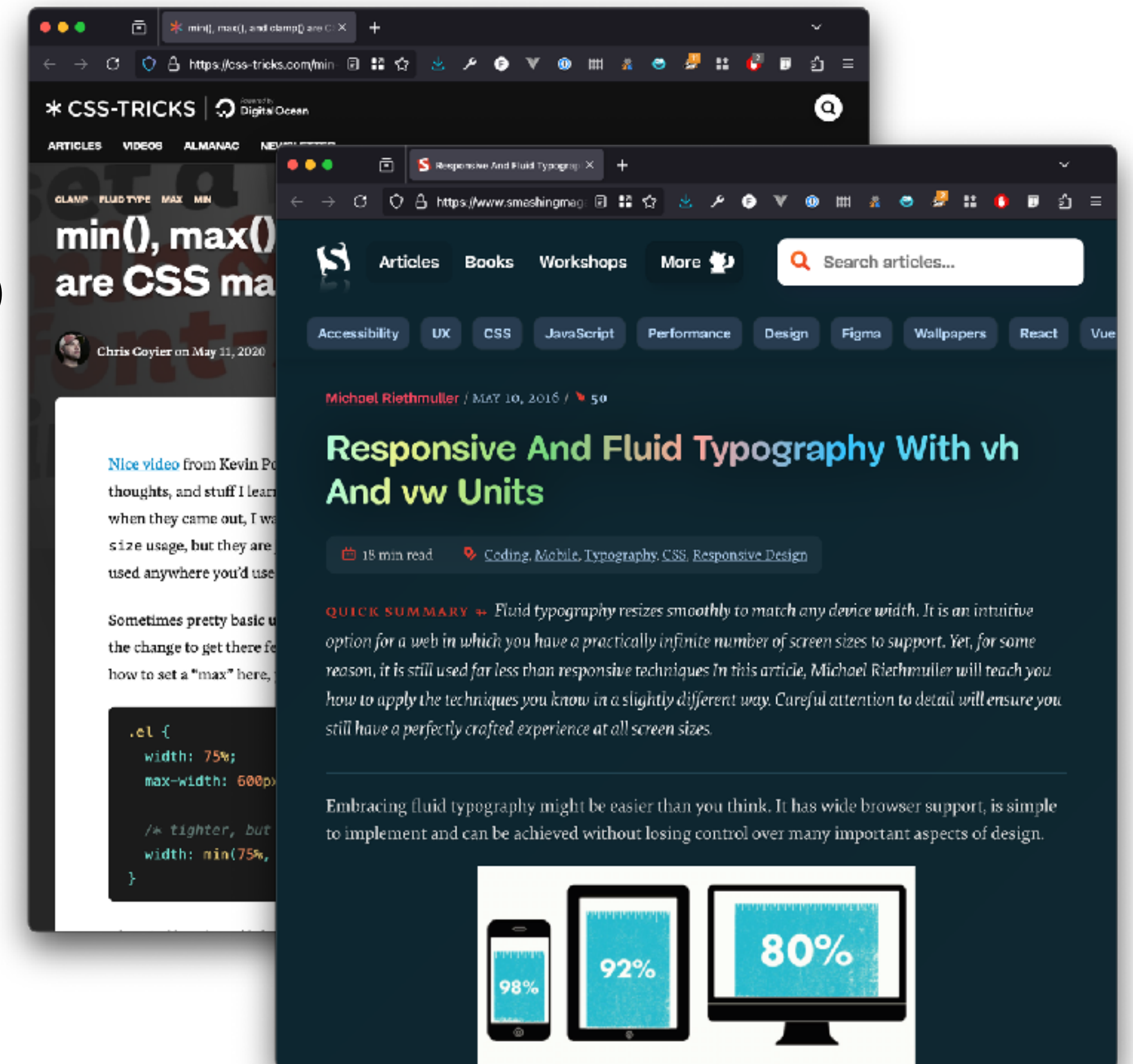
Breakpoints

Liquid Layout

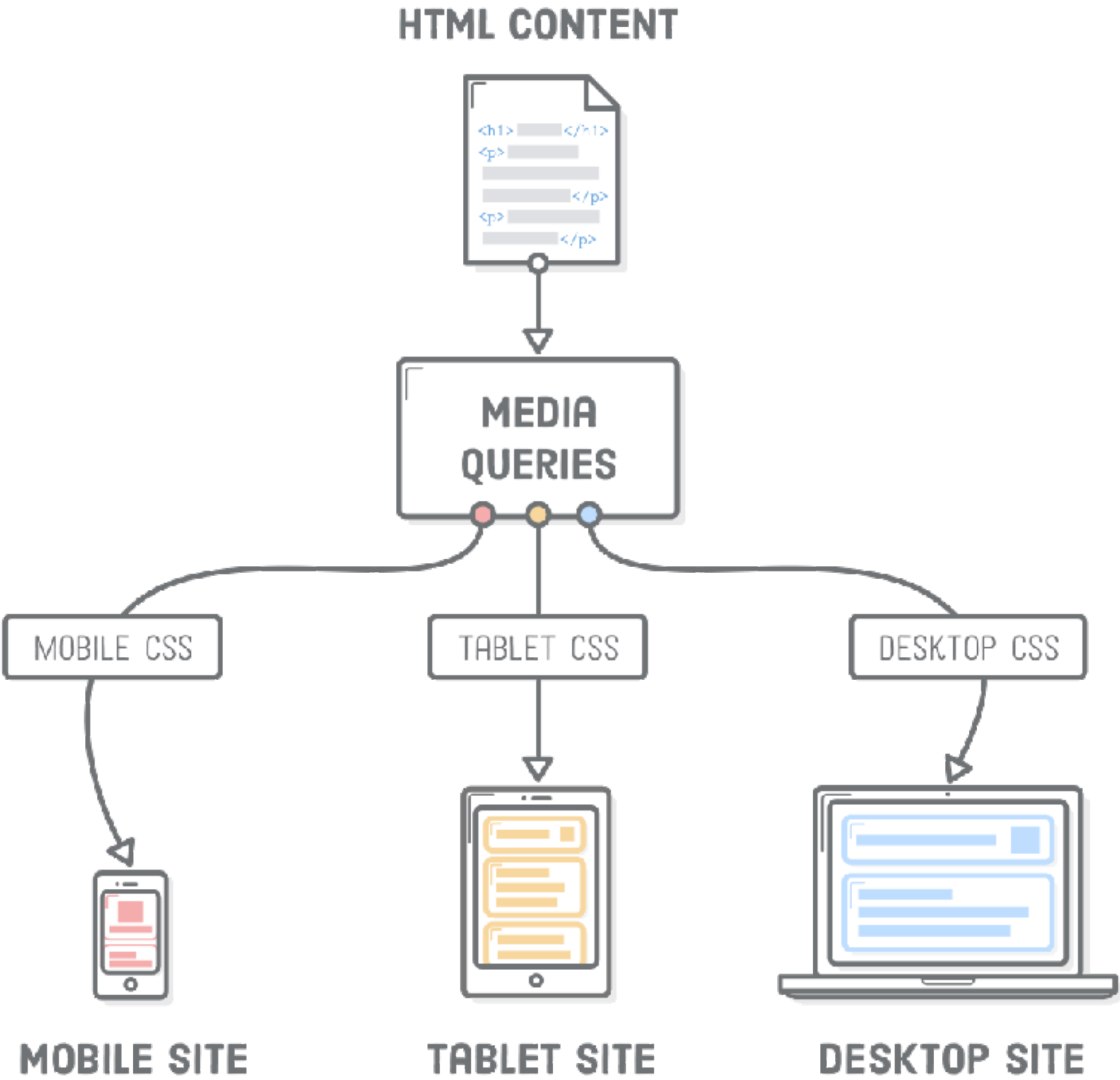


LIQUID LAYOUTS

- Build your layout in percentages (%)
- Often not enough
→ hint: liquid layouts with min and max values
- Article about Fluid Typography
(<https://www.smashingmagazine.com/2016/05/fluid-typography/>)
- Article about min, max and clamp
(<https://css-tricks.com/min-max-and-clamp-are-css-magic/>)



MEDIA QUERIES IN ACTION



BREAKPOINTS → MEDIAQUERIES

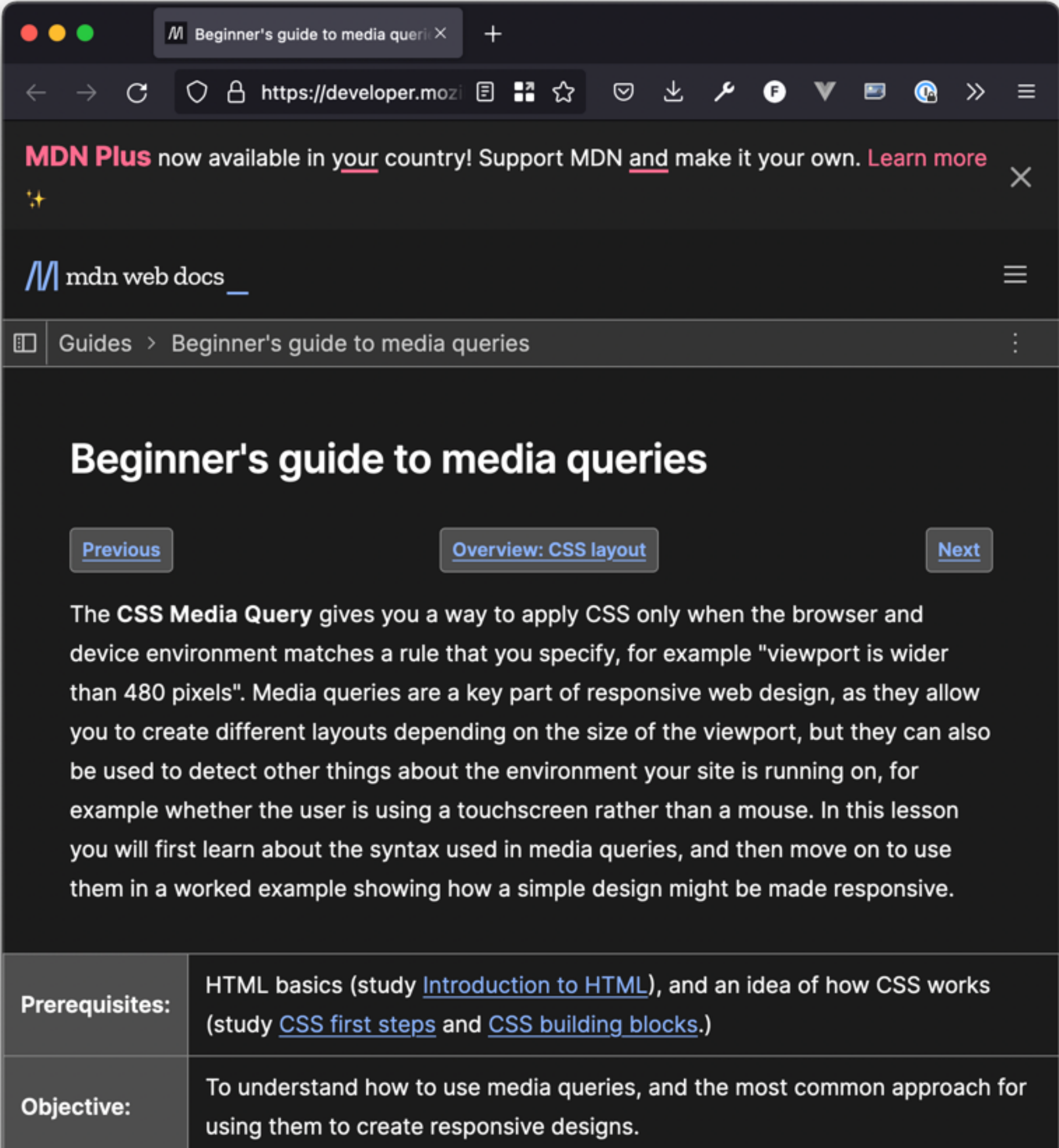
- Mobile First vs. Desktop First



```
1  /* use your own selectors, of course! .class is a placeholder */
2  .class {
3      /* mobile styles */
4  }
5
6  @media only screen and (min-width: 768px) {
7      .class {
8          /* tablet styles */
9      }
10 }
11
12 @media only screen and (min-width: 1280px) {
13     .class {
14         /* desktop styles */
15     }
16 }
```

BREAKPOINTS → MEDIAQUERIES

- https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Media_queries



MEDIA QUERIES — EXERCISE 13B

- Solve little exercises about media queries
- Download files from:
<https://short.jonasscheiwiller.ch/ex13b>

RESPONSIVE IMAGES

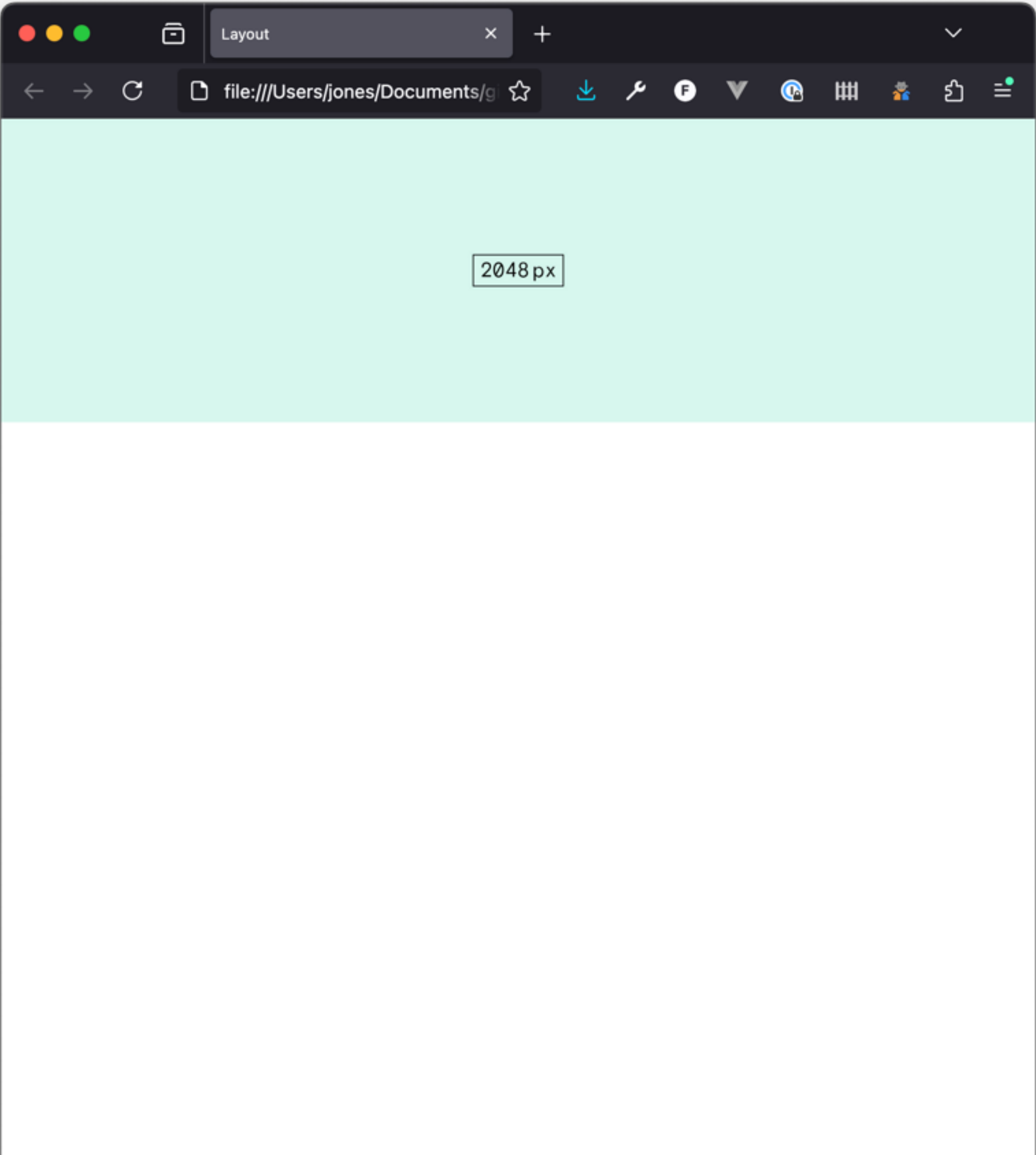


```
1 
```

Helpful:

<https://www.smashingmagazine.com/2014/05/responsive-images-done-right-guide-picture-srcset/>

RESPONSIVE IMAGES



FUNCTIONS → DECLARATION

```
1  // Function declaration
2  function greet1() {
3      console.log("Hello, world!");
4  }
5
6  // Function expression
7  const greet2 = function() {
8      console.log("Hello, world!");
9  };
10
11 // Arrow function expression
12 const greet3 = () => {
13     console.log("Hello, world!");
14 };
15
16 // Method declaration in an object
17 const obj = {
18     greet4: function() {
19         console.log("Hello, world!");
20     }
21 };
22
23 // ES6 shorthand method definition
24 const obj2 = {
25     greet5() {
26         console.log("Hello, world!");
27     }
28 };
29
30 // Method using function constructor
31 const greet6 = new Function('console.log("Hello, world!");');
```

FUNCTIONS → PARAMETERS



```
1  function sum(a, b) {  
2      return a + b;  
3  }  
4  
5  console.log(sum(5, 3)); // → 8  
6  console.log(sum(10, 7)); // → 17
```

FUNCTIONS — EXERCISE 14

- Solve little function exercises
- Download files from:
<https://short.jonasscheiwiller.ch/ex14>

```
1 // =====
2 // EXERCISE: Functions & Parameters
3 // =====
4 //
5 // Open the browser console to see your results.
6 // (Right-click the page → Inspect → Console)
7 //
8 // =====
9
10 // Formats a price with a currency symbol
11 function formatPrice(amount, currency) {
12   console.log(`${currency} ${amount.toFixed(2)}`);
13 }
14
15 formatPrice(9.9, "CHF"); // → CHF 9.90
16 formatPrice(4.5, "EUR"); // → EUR 4.50
17
18 // =====
19
20 // TODO 1: call formatPrice() with a price of your choice
21
22 // TODO 2: define a function called discount
23 //         it takes two parameters: price and percent
24 //         it should log the reduced price
25 //         EXAMPLE: discount(100, 20) → "Price after discount: CHF 80.00"
26
27 // TODO 3: call discount() with two different examples
28
```