

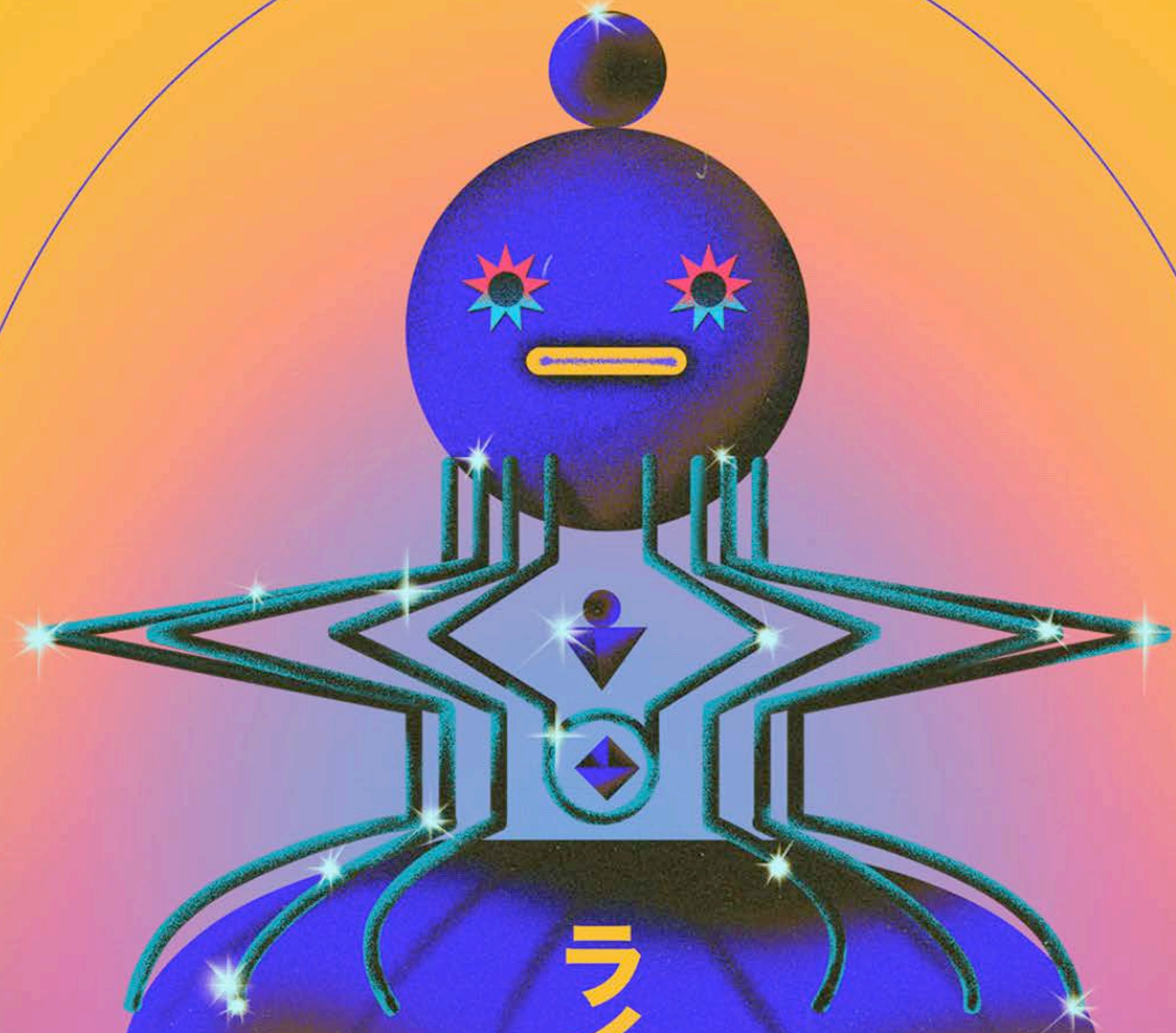
24:03

Bits & Atoms II

CSS

名ウキア保年ヤ
んまス権黄クイメワ伊

ちべ通以ハシコ東
カ万通ヤミ成ニハス
製貯親4海賢だ文説あっ。



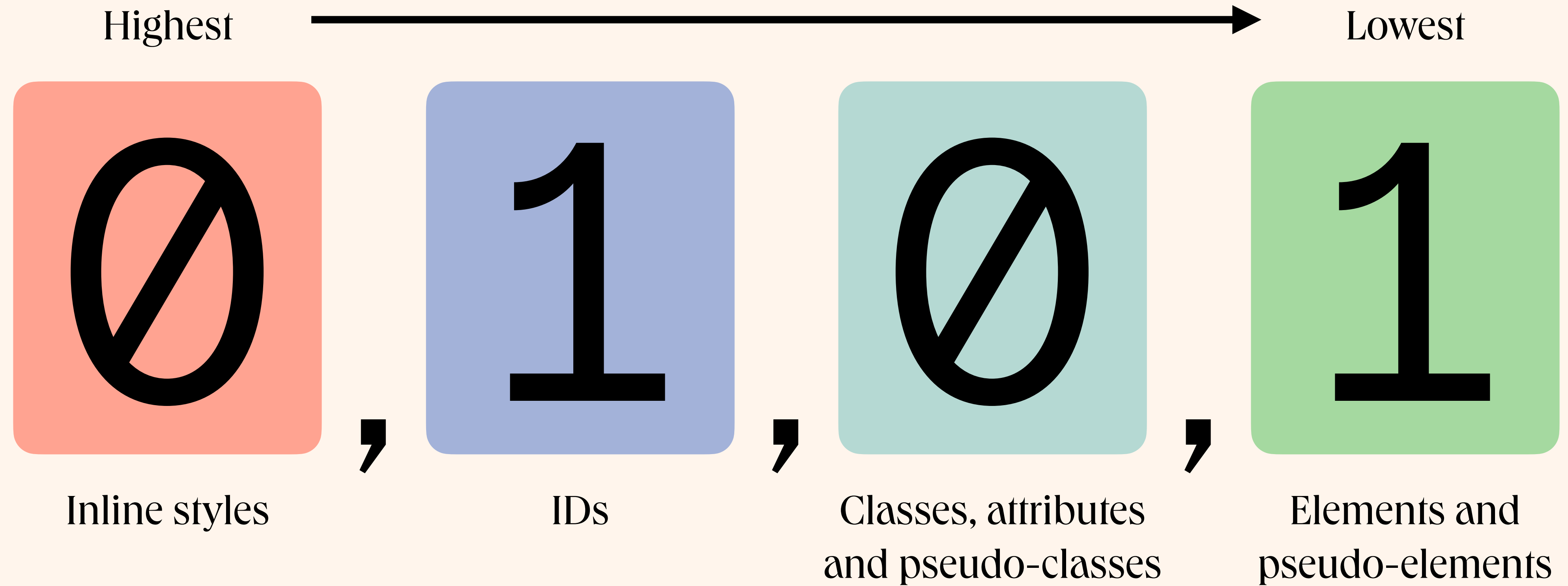
ライ イ ブ

Hands-on: style.css



```
1 * {  
2   margin: 0;  
3 }  
4  
5 .class {  
6   font-size: 30px;  
7   margin: 10px;  
8 }
```

CSS *Specificity*



CSS Specificity

Specificity

Example

0, 1, 0, 0

`#my-id {...}`

0, 0, 2, 0

`.class1 .class2 {...}`

0, 0, 1, 1

`p.class1 {...}`

0, 1, 1, 0

`#my-id.class1 {...}`

0, 0, 1, 0

`.class {...}`

CSS Specificity



a

1 x element selector

Sith: **0, 0, 1**



p a

2 x element selectors

Sith: **0, 0, 2**



.whatever

1 x class selector

Sith: **0, 1, 0**



a.whatever

1 x element selector
1 x class selector

Sith: **0, 1, 1**



p a.whatever

2 x element selectors
1 x class selector

Sith: **0, 1, 2**



.whatever .whatever

2 x class selectors

Sith: **0, 2, 0**



p.whatever a.whatever

2 x element selectors
2 x class selectors

Sith: **0, 2, 2**



#whatever

1 x id selector

Sith: **1, 0, 0**



a#whatever

1 x element selector
1 x id selector

Sith: **1, 0, 1**



.whatever a#whatever

1 x element selectors
1 x class selector
1 x id selector

Sith: **1, 1, 1**



.whatever .whatever #whatever

2 x class selectors
1 x id selector

Sith: **1, 2, 0**



#whatever #whatever

2 x id selectors

Sith: **2, 0, 0**

CSS *Specificity*



!important

Quiz: Red or Blue?

```
1 <!doctype html>
2
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <meta name="description" content="">
7   <title>Template</title>
8 </head>
9
10 <body>
11   <div id="main">
12     <section class="section" id="first-section">
13       <div class="wrapper title-wrapper">
14         <h1 class="heading">Salut 🙌</h1>
15         <ul class="list">
16           <li class="list-item">List item 1</li>
17           <li class="list-item">List item 2</li>
18           <li class="list-item">List item 3</li>
19         </ul>
20       </div>
21     </section>
22   </div>
23 </body>
24
25 </html>
```

```
1 h1.heading {
2   color: red;
3 }
4
5 .wrapper .heading {
6   color: blue;
7 }
8
```


Quiz: Red or Blue?

```
1 <!doctype html>
2
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <meta name="description" content="">
7   <title>Template</title>
8 </head>
9
10 <body>
11   <div id="main">
12     <section class="section" id="first-section">
13       <div class="wrapper title-wrapper">
14         <h1 class="heading">Salut 🙌</h1>
15         <ul class="list">
16           <li class="list-item">List item 1</li>
17           <li class="list-item">List item 2</li>
18           <li class="list-item">List item 3</li>
19         </ul>
20       </div>
21     </section>
22   </div>
23 </body>
24
25 </html>
```

```
1  div .heading {
2    color: red;
3  }
4
5  h1.heading {
6    color: blue;
7  }
8
```


Quiz: Red or Blue?

```
1 <!doctype html>
2
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <meta name="description" content="">
7   <title>Template</title>
8 </head>
9
10 <body>
11   <div id="main">
12     <section class="section" id="first-section">
13       <div class="wrapper title-wrapper">
14         <h1 class="heading">Salut 🙌</h1>
15         <ul class="list">
16           <li class="list-item">List item 1</li>
17           <li class="list-item">List item 2</li>
18           <li class="list-item">List item 3</li>
19         </ul>
20       </div>
21     </section>
22   </div>
23 </body>
24
25 </html>
```

```
1 section .heading {
2   color: red;
3 }
4
5 h1 .heading {
6   color: blue;
7 }
```

Quiz: Red or Blue?

```
1 <!doctype html>
2
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <meta name="description" content="">
7   <title>Template</title>
8 </head>
9
10 <body>
11   <div id="main">
12     <section class="section" id="first-section">
13       <div class="wrapper title-wrapper">
14         <h1 class="heading">Salut 🙌</h1>
15         <ul class="list">
16           <li class="list-item">List item 1</li>
17           <li class="list-item">List item 2</li>
18           <li class="list-item">List item 3</li>
19         </ul>
20       </div>
21     </section>
22   </div>
23 </body>
24
25 </html>
```

```
1 #first-section .heading {
2   color: red;
3 }
4
5 .section div .heading {
6   color: blue;
7 }
```

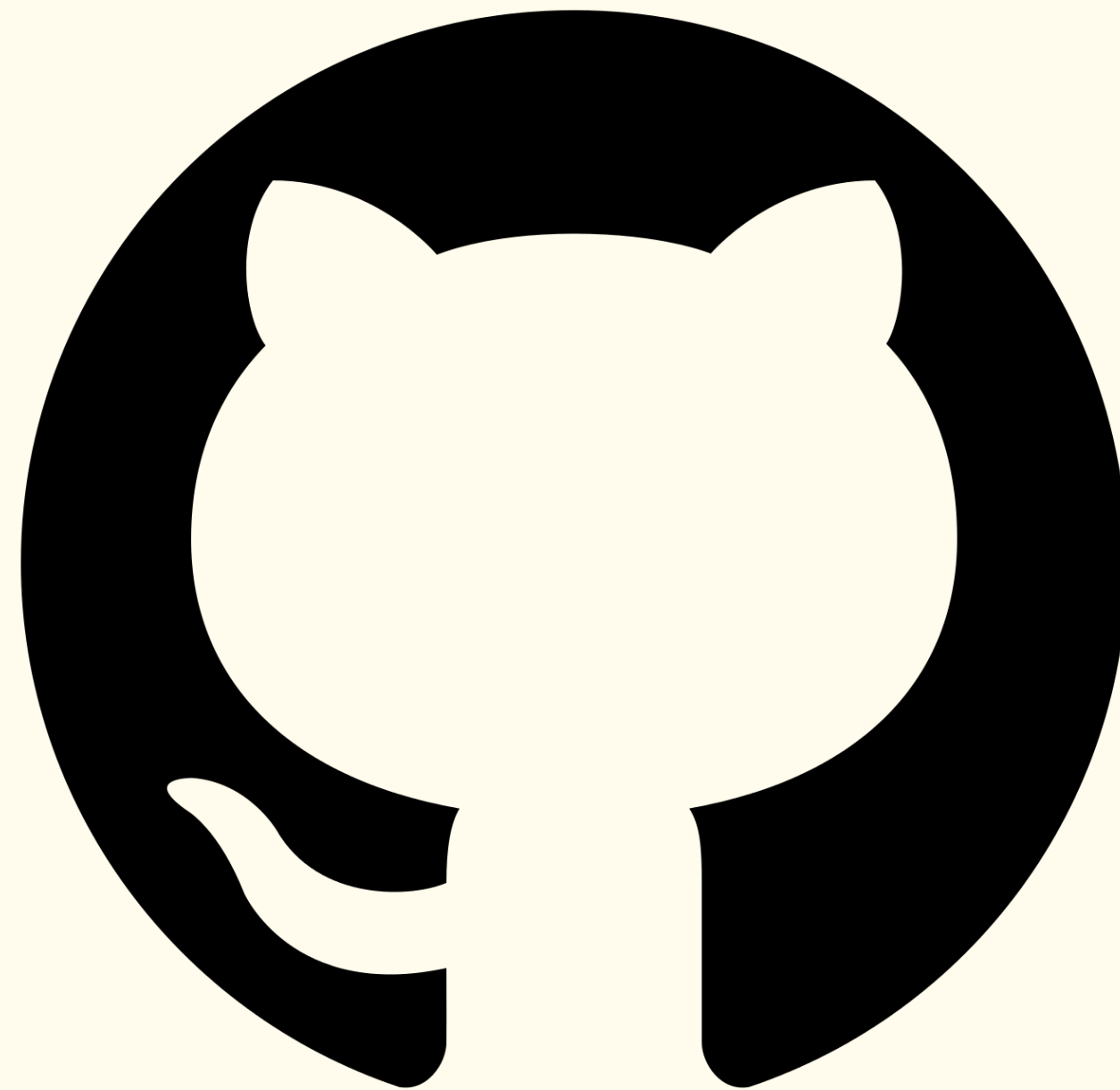

Quiz: Red or Blue?

```
1 <!doctype html>
2
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <meta name="description" content="">
7   <title>Template</title>
8 </head>
9
10 <body>
11   <div id="main">
12     <section class="section" id="first-section">
13       <div class="wrapper title-wrapper">
14         <h1 class="heading">Salut 🙌</h1>
15         <ul class="list">
16           <li class="list-item">List item 1</li>
17           <li class="list-item">List item 2</li>
18           <li class="list-item">List item 3</li>
19         </ul>
20       </div>
21     </section>
22   </div>
23 </body>
24
25 </html>
```

```
1  div #first-section li {
2    color: red;
3  }
4
5  #main ul li.list-item {
6    color: blue;
7  }
```

Bits & Atoms II

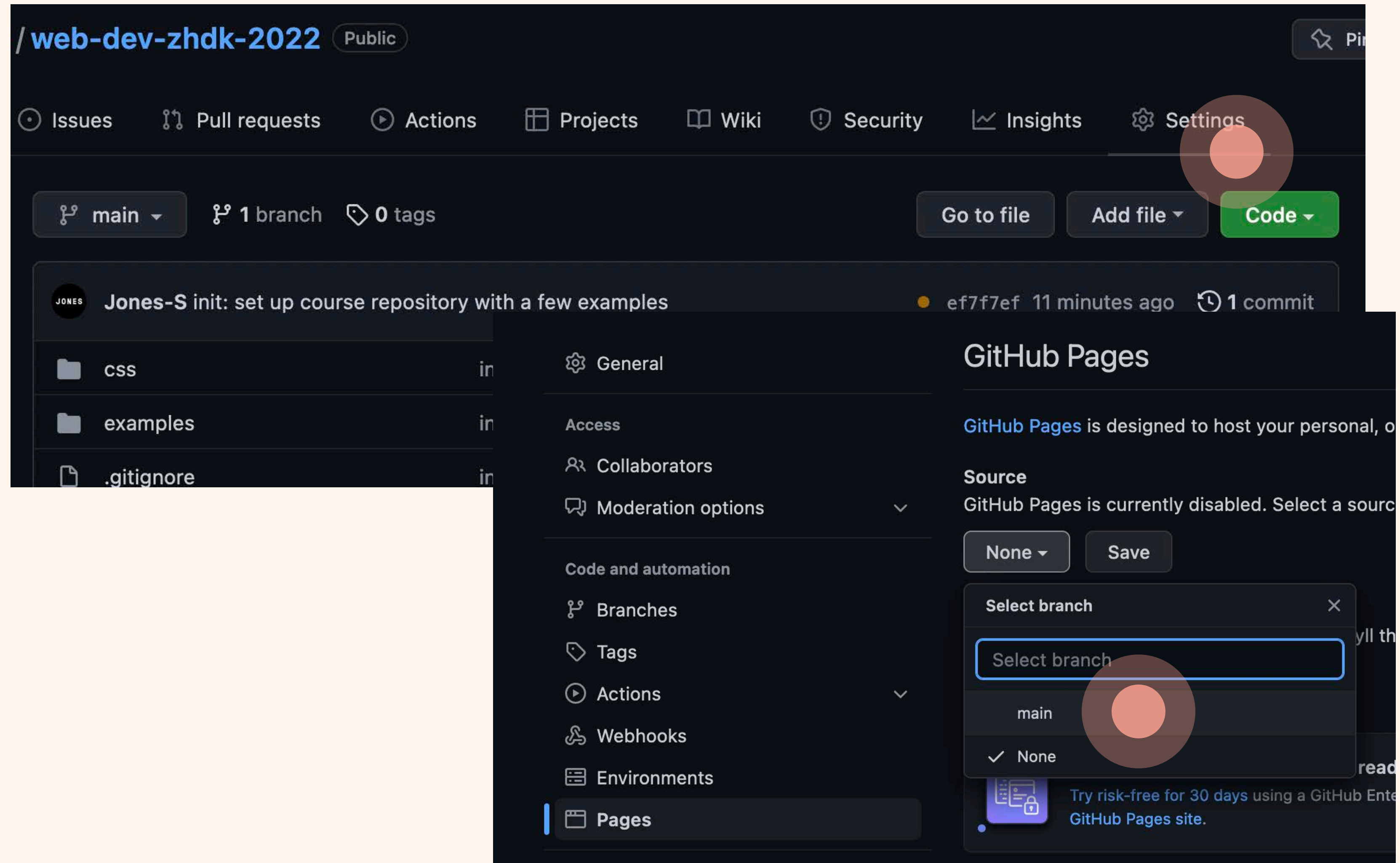
GitHub!



Hosting a website on Github

Github Repository
→ Settings
→ Pages

Select source
branch (main) and
folder. Done.



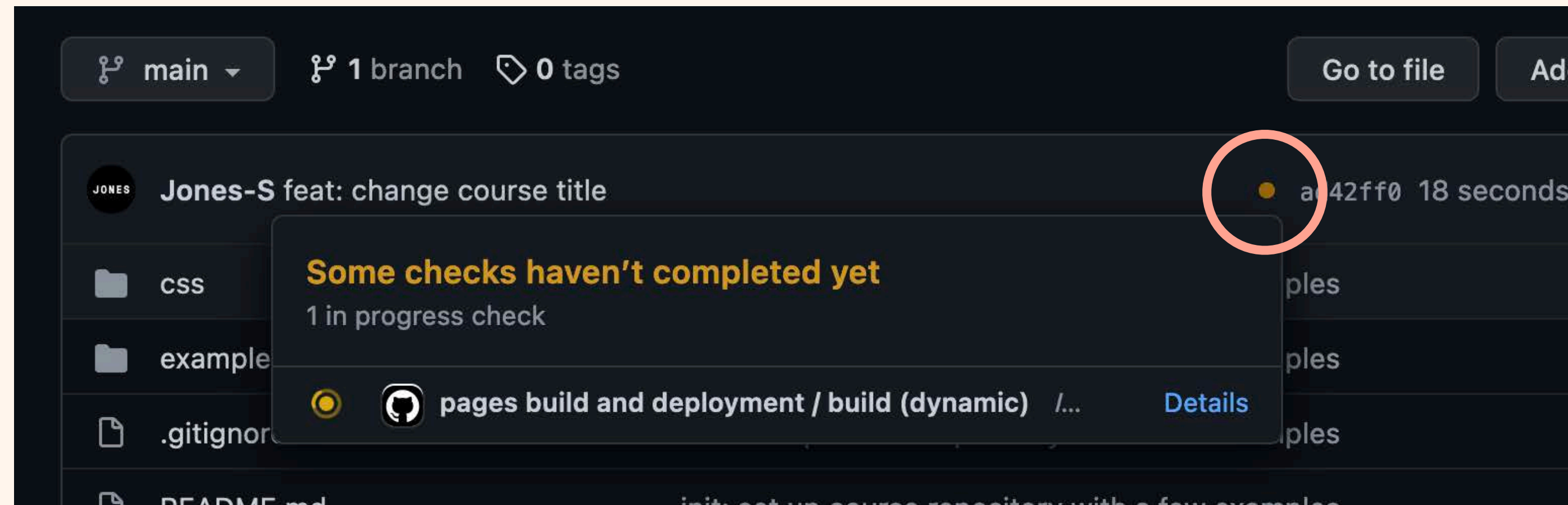
Hosting a website on Github

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/>



Using Github to go back?

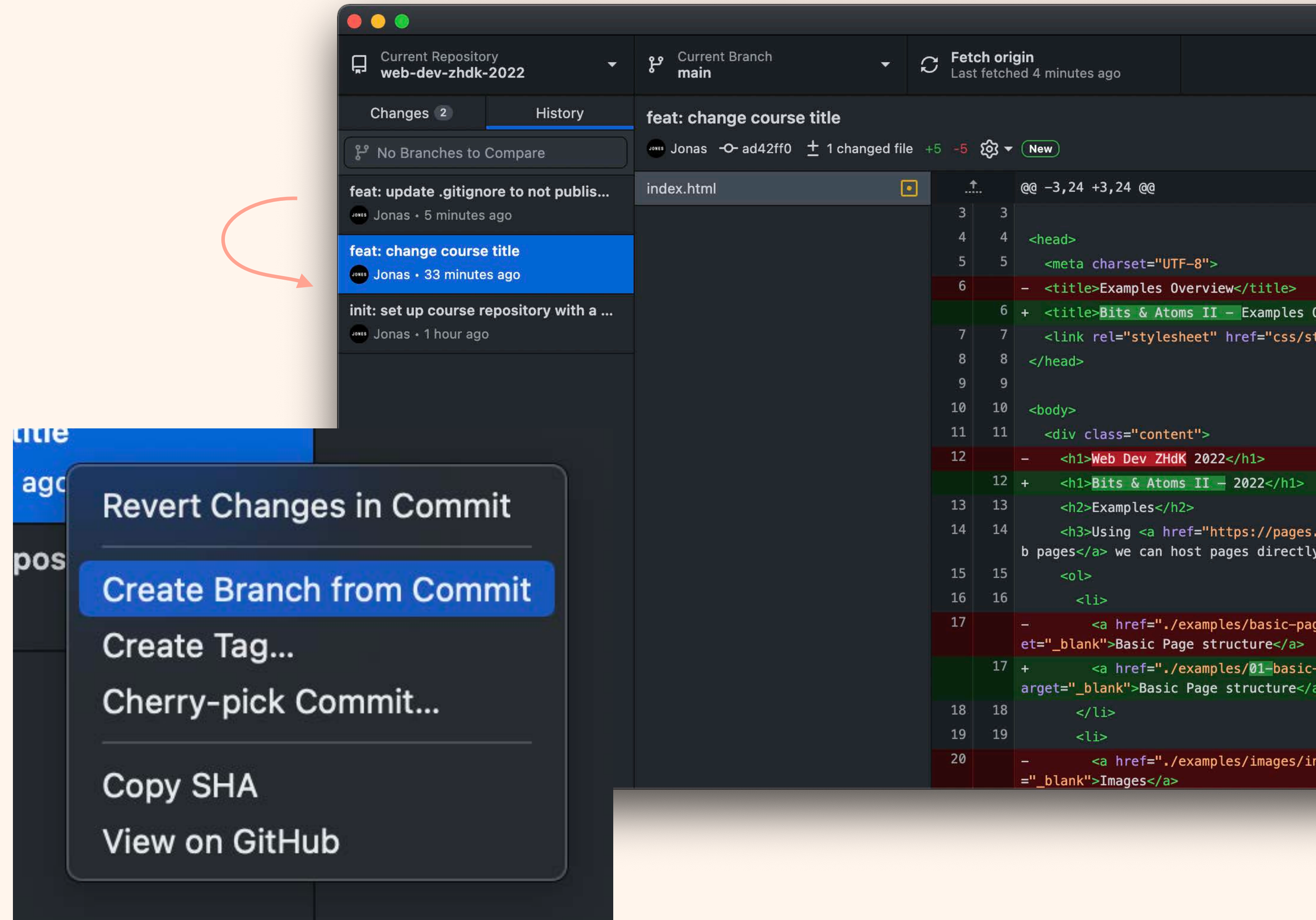
Github Desktop can only switch between branches. 🙄

But:

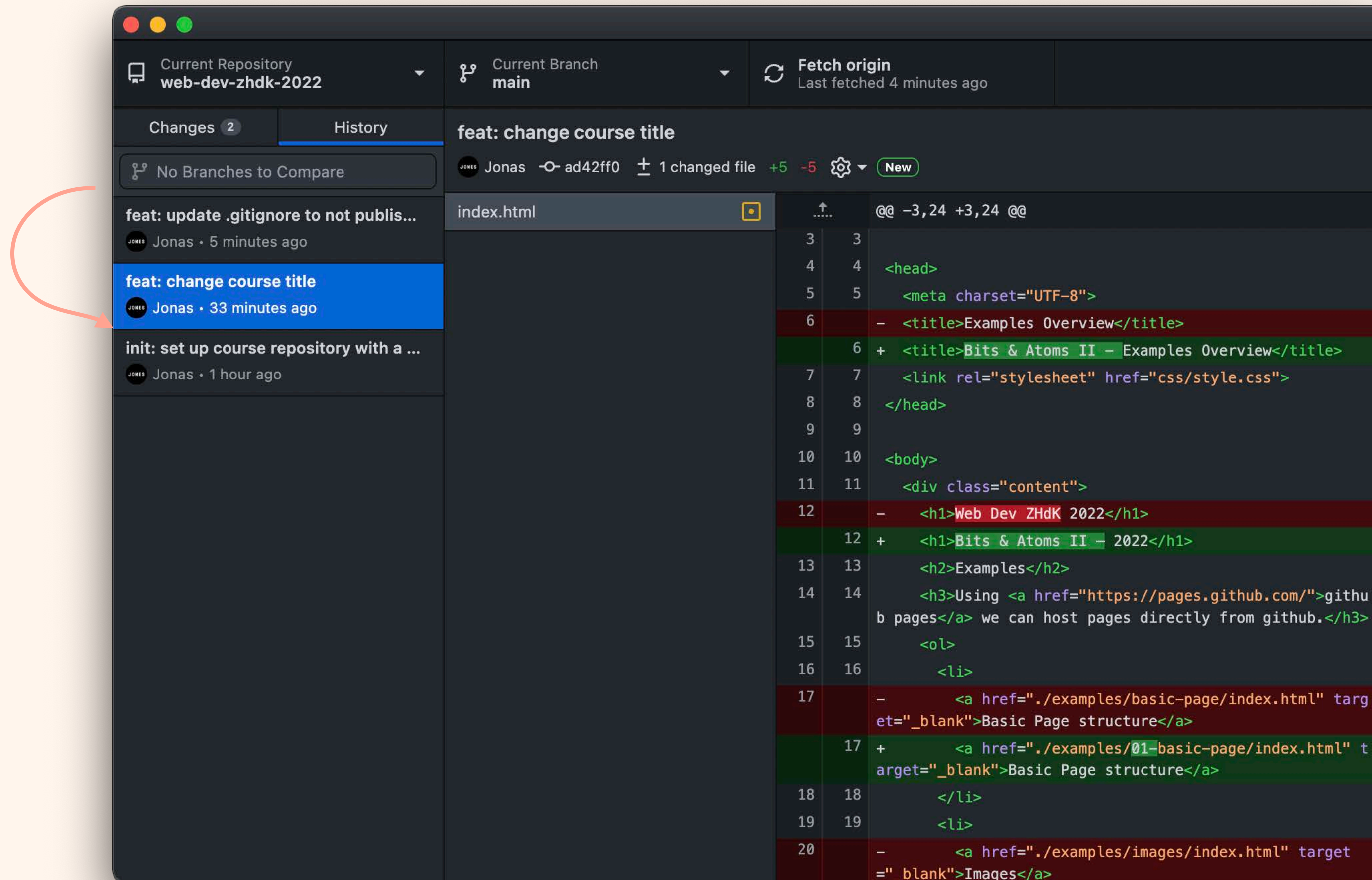
Right click

→ Create Branch from Commit

does the trick



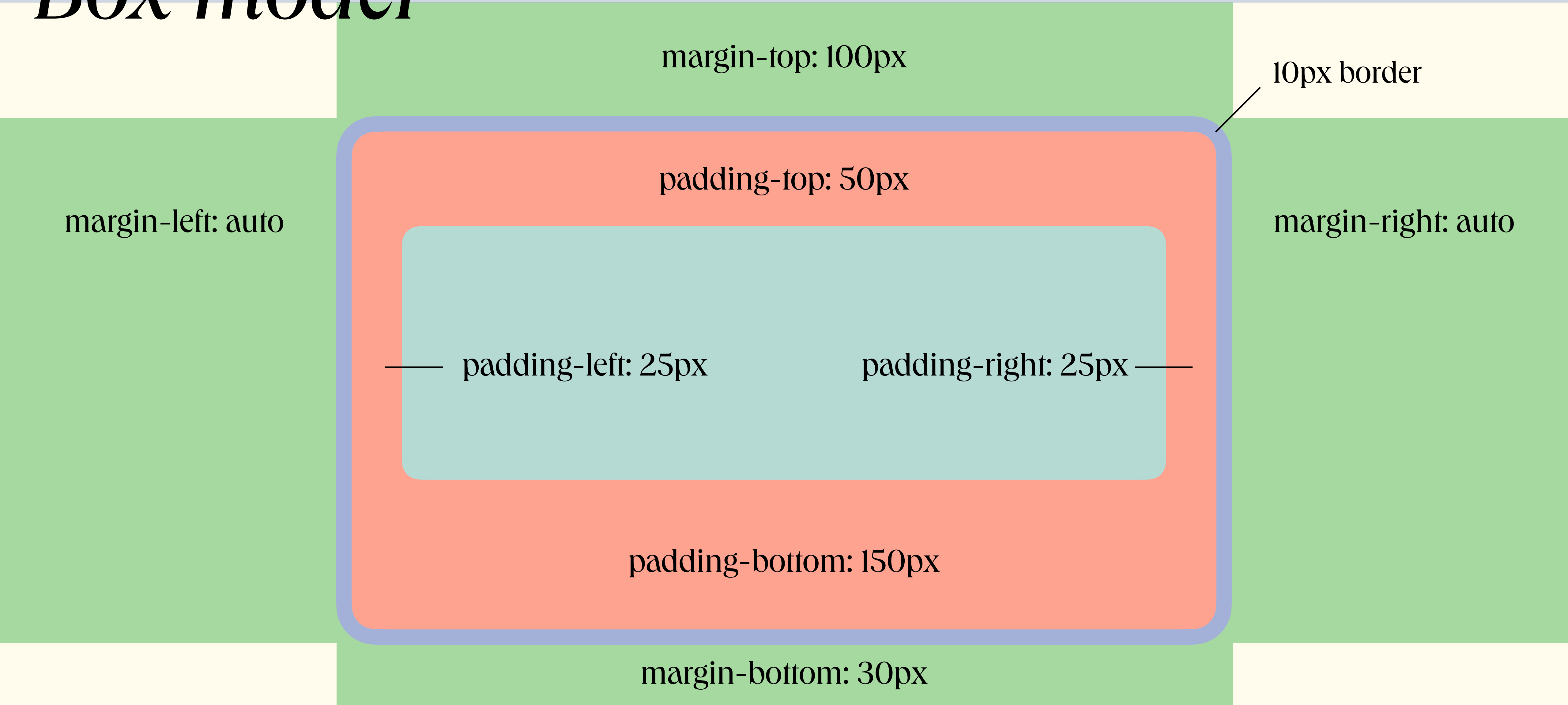
Demo: Switching branches and commits



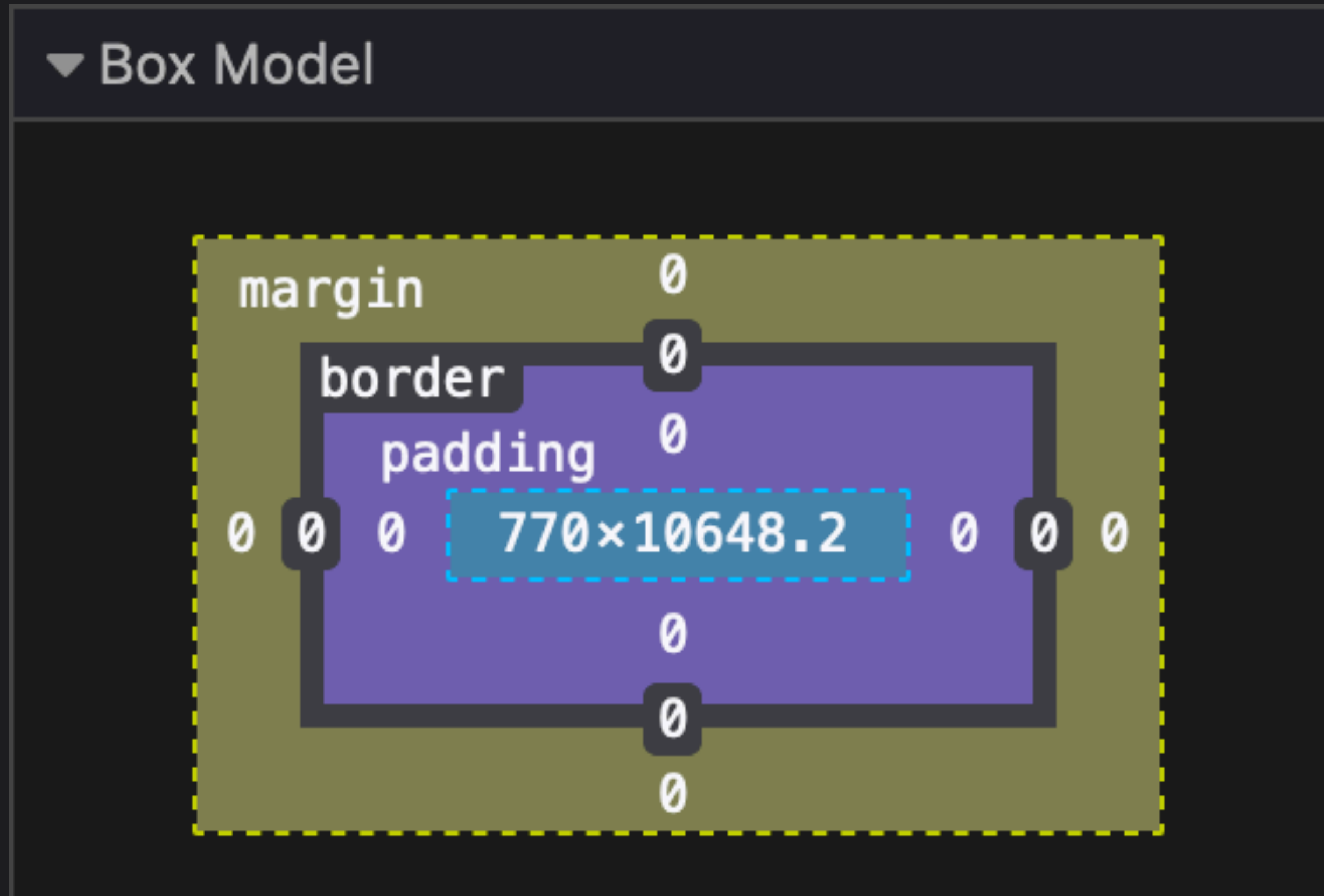
Divs and the box model

- The box-model
 - Padding, margin and border
- Display
- Positioning
 - absolute, fixed, relative, static

Box model



Box model



box-sizing property

Include padding and border in the element's total width and height.



```
1 * {  
2   box-sizing: border-box;  
3 }
```


Display

Most used
display values.



```
1 display: block;  
2 display: inline;  
3 display: inline-block;  
4 display: flex;  
5 display: inline-flex;  
6 display: grid;  
7 display: inline-grid;
```

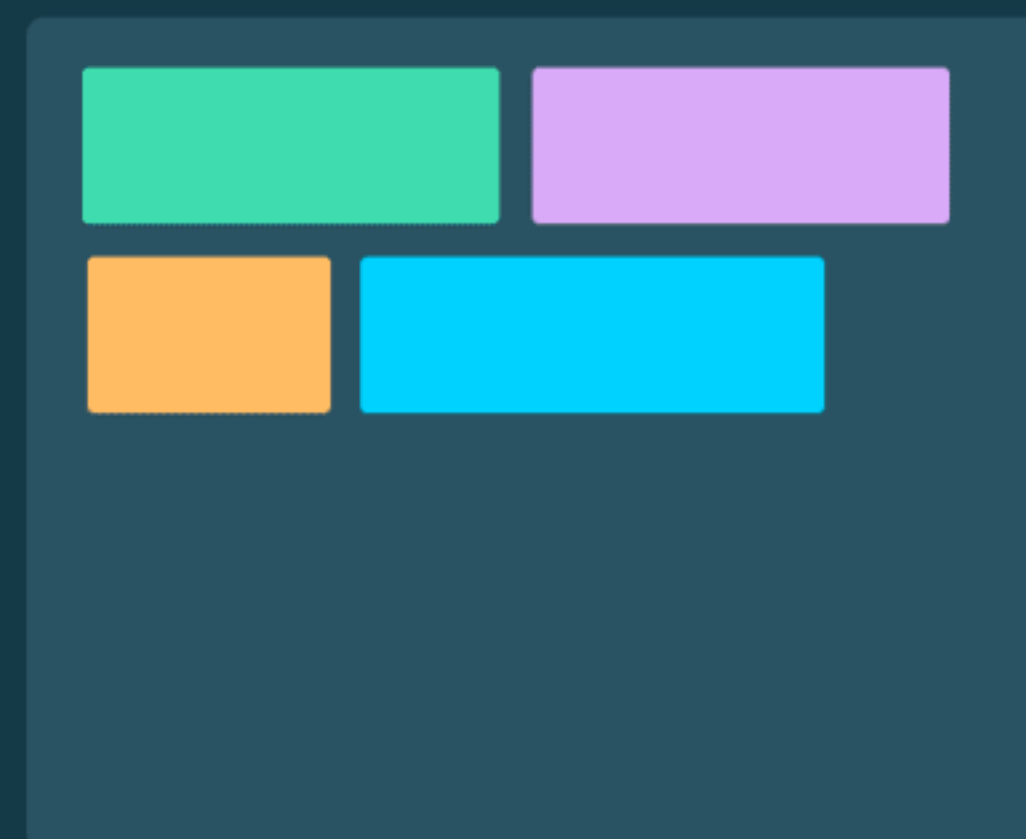
Inline vs. block

BLOCK VS INLINE



display: block;

Block elements stack, regardless of their width.



display: inline;

Inline elements flow from one line to the next.

Flex and Grid

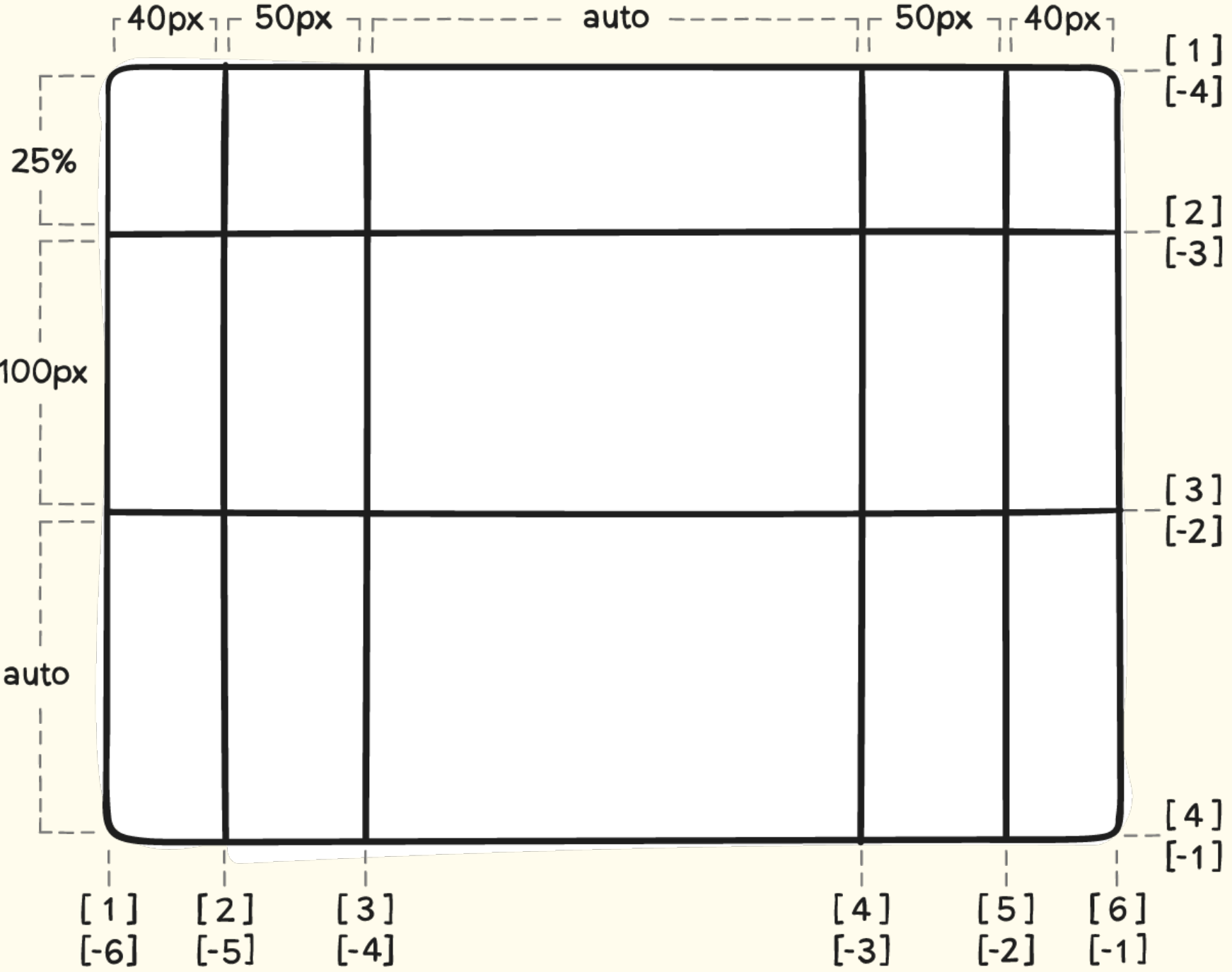
Flexbox



“FLEX CONTAINER”



“FLEX ITEMS”



CSS Grid

Flex and Grid

Flexbox

A screenshot of a web browser displaying a guide to Flexbox properties. The page is titled "Flexbox properties" and is divided into two main sections: "Properties for the Parent (flex container)" and "Properties for the Children (flex items)".

Properties for the Parent (flex container)

- display**: This defines a flex container; inline or block depending on the given value. It enables a flex context for all its direct children.

```
.container {  
  display: flex; /* or inline-flex */  
}
```

Note that CSS columns have no effect on a flex container.

Properties for the Children (flex items)

- order**: Illustrates how the order property affects the visual sequence of items. Examples shown include: [1, 1, 1, 2, 3], [-1, 1, 2, 5, -], and [2, 2, 99].

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

CSS Grid

A screenshot of a web browser displaying a complete guide to CSS Grid. The page includes definitions for track-size and line-name, CSS code examples for grid-template-columns and grid-template-rows, and a diagram illustrating grid lines and item placement.

- <track-size>** - can be a length, a percentage, or a fraction of the free space in the grid (using the `fr` unit)
- <line-name>** - an arbitrary name of your choosing

```
.container {  
  grid-template-columns: ... ...;  
  /* e.g.  
  1fr 1fr  
  minmax(10px, 1fr) 3fr  
  repeat(5, 1fr)  
  50px auto 100px 1fr  
  */  
  grid-template-rows: ... ...;  
  /* e.g.  
  min-content 1fr min-content  
  100px 1fr max-content  
  */  
}
```

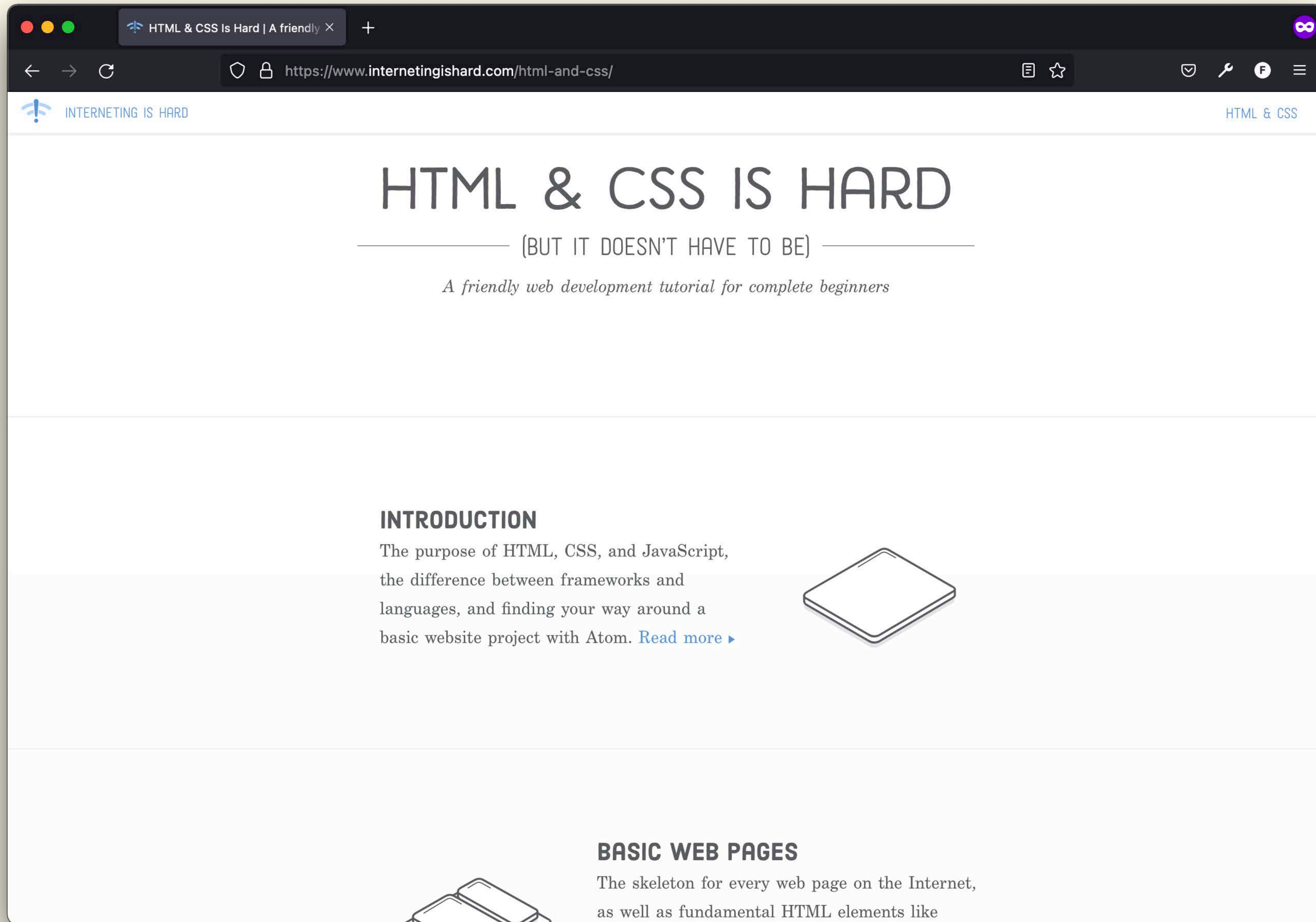
Grid lines are automatically assigned positive numbers from these assignments (-1 being an alternate for the very last row).

Examples:

```
.item-a {  
  grid-column-start: 2;  
  grid-column-end: five;  
  grid-row-start: row1-start;  
  grid-row-end: 3;  
}
```

```
.item-b {  
  grid-column-start: 1;  
  grid-column-end: span col4-start;  
  grid-row-start: 2;  
  grid-row-end: span 2;  
}
```

<https://css-tricks.com/snippets/css/complete-guide-grid/>



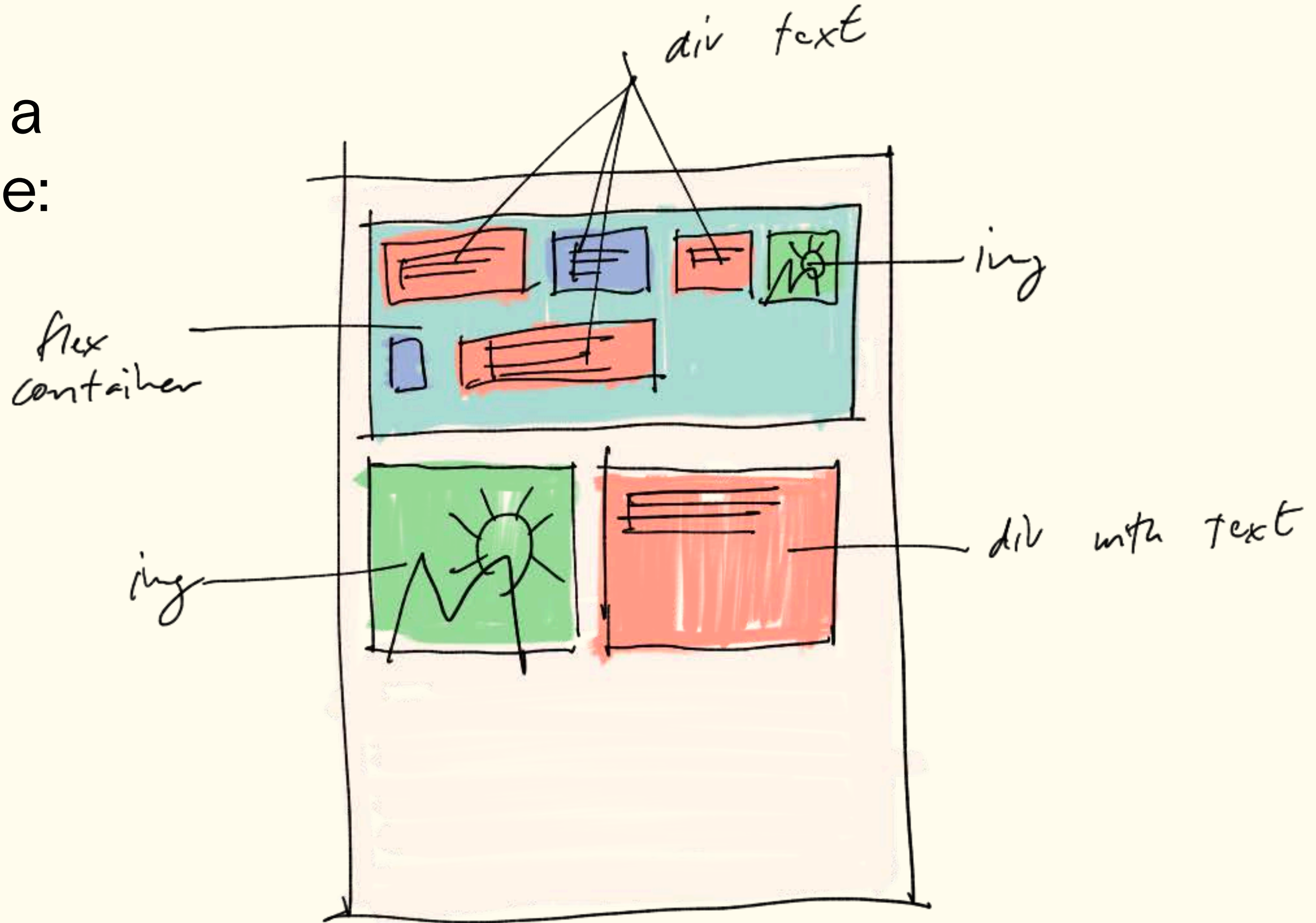
Valuable resource:

<https://www.internetingishard.com/html-and-css/>

Hands-on

Start with a rough sketch of a layout very similar to this one:

Try to setup the HTML and style and position your elements.

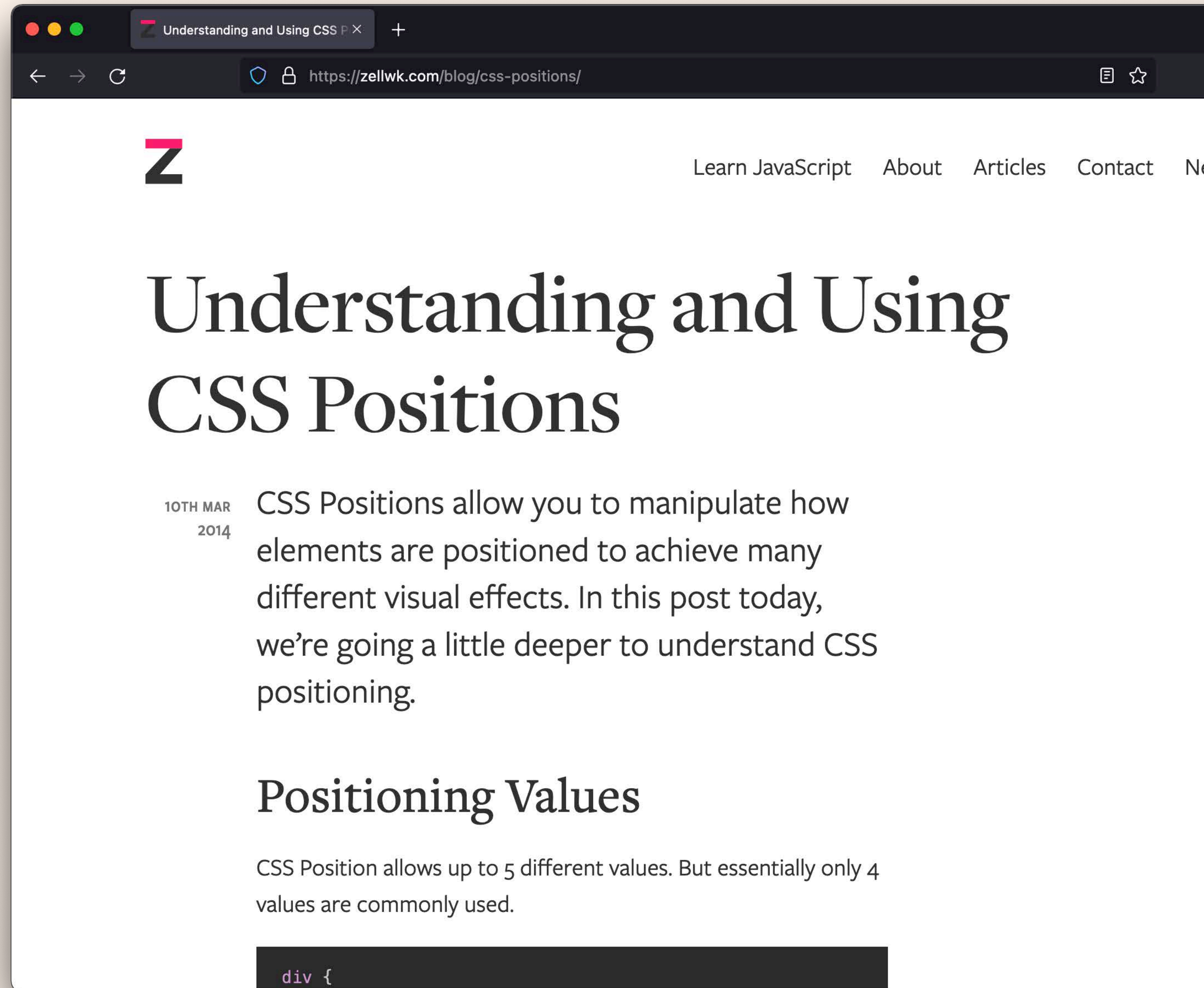


Positioning

<https://zellwk.com/blog/css-positions/>



```
1 position: static; /* default */  
2 position: relative;  
3 position: absolute;  
4 position: fixed;  
5 position: inherit; /* Not very common */
```



Course repository

Github pages → examples

<https://jones-s.github.io/web-dev-zhdk-2022/>

Github repo

<https://github.com/Jones-S/web-dev-zhdk-2022/>