

PICTURE



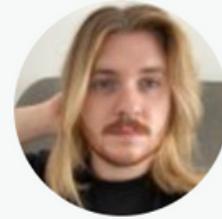


**Ben**

@cinemashoebox · [Follow](#)



the people in the background here do not have legs. are  
metalegs only granted at a certain level of seniority. are  
legs a right or a privilege, mark



**gordon**

@hetnonsense · [Follow](#)

The fish planning to leave the primordial soup:

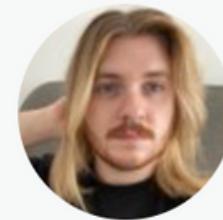


**Ben**

@cinemashoebox · [Follow](#)



the people in the background here do not have legs. are  
metalegs only granted at a certain level of seniority. are  
legs a right or a privilege, mark

**gordon**@hetnonsense · [Follow](#)

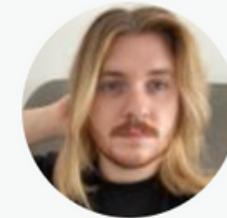
The fish planning to leave the primordial soup:

**Smooth Dunk**@SmoothDunk · [Follow](#)**Meta**

We have added legs because new technology allowed us to.  
It is not because I just remembered humans have legs.

**Ben**@cinemashoebox · [Follow](#)

the people in the background here do not have legs. are  
metalegs only granted at a certain level of seniority. are  
legs a right or a privilege, mark



**gordon**

@hetnonsense · [Follow](#)

The fish planning to leave the primordial soup:



That feeling when the stock is down 50% year-over-year.

Screenshot: Meta / Facebook / Kotaku



**Smooth Dunk**

@SmoothDunk · [Follow](#)



 **Meta**

We have added legs because new technology allowed us to.  
It is not because I just remembered humans have legs.



**Ben**

@cinemashoebox · [Follow](#)



the people in the background here do not have legs. are  
metalegs only granted at a certain level of seniority. are  
legs a right or a privilege, mark

We appreciate your  
presence!

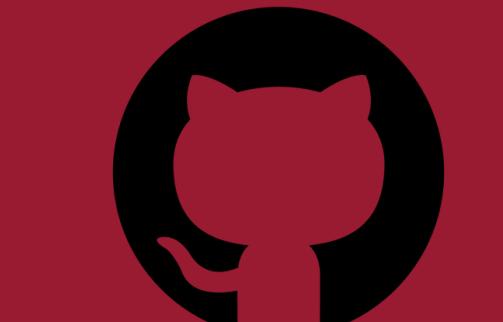
# WHAT'S CICSOFT?

- Interest-oriented CICS organization
- Environment to enhance their technical skills
- Members learn developing applications

## WHY?

Our aim is to empower students by delivering the means to build incredible apps and realize their ideas through code.

# LAST TIME



**GitHub**

**HTML**



**CSS**



- Git/GitHub:
  - Control and track changes to files
- HTML:
  - Structure layout for websites
- CSS:
  - Apply styles to websites

# THIS TIME

- More CSS:
  - Apply styles to websites
- JavaScript for the Web:
  - Add logic for websites
- "HTML/CSS/JS":
  - Build interactive websites



BACK TO CSS{}

# CSS

## Introducing Flexboxes

### Sample flexbox example

#### First article

Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

#### Second article

Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

#### Third article

Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

Cray food truck brunch, XOXO +1 keffiyeh pickled chambray waistcoat ennui. Organic small batch paleo 8-bit. Intelligentsia umami wayfarers pickled, asymmetrical kombucha letterpress kitsch leggings cold-pressed squid chartreuse put a bird on it. Listicle pickled man bun cornhole heirloom art party.

# CSS Flexboxes

Resource : [css-tricks guide to flexboxes](#)

# CSS

## Think like Flexboxes

<div>

</div>

# CSS

## Think like Flexboxes

<div>

</div>

Container!

# CSS

## Think like Flexboxes

<div>

</div>

Container!



The screenshot shows a web browser window with the title "The <div> Tag". The URL bar contains "https://www.w3docs.com". The main content area displays the text "The <div> Tag" in large, bold, dark blue font. Below it, a green box contains the explanatory text: "We use the <div> tag to group two paragraphs for applying background to the text, and to add color to this word we place it within <span> tag." At the bottom of this green box, another message reads: "Pay attention, that the <div> tag is a block-level element, so a line break is placed before and after it." The browser interface includes standard window controls (red, yellow, green circles) and navigation buttons.

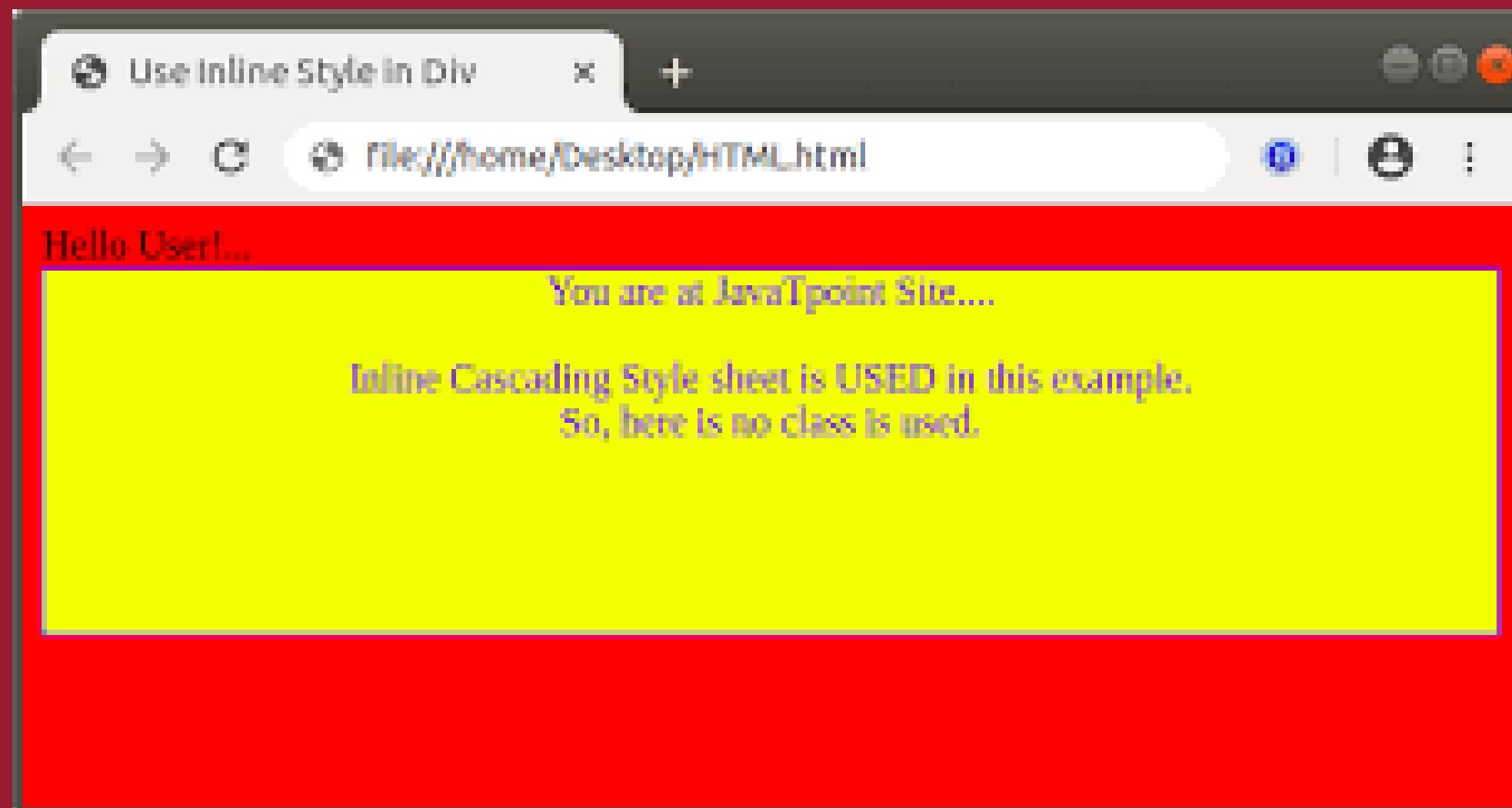
**The <div> Tag**

We use the <div> tag to group two paragraphs for applying background to the text, and to add color to this word we place it within <span> tag.

Pay attention, that the <div> tag is a block-level element, so a line break is placed before and after it.

# CSS

## Think like Flexboxes



A screenshot of a web browser window titled "The <div> Tag". The address bar shows "https://www.w3docs.com". The page content is as follows:

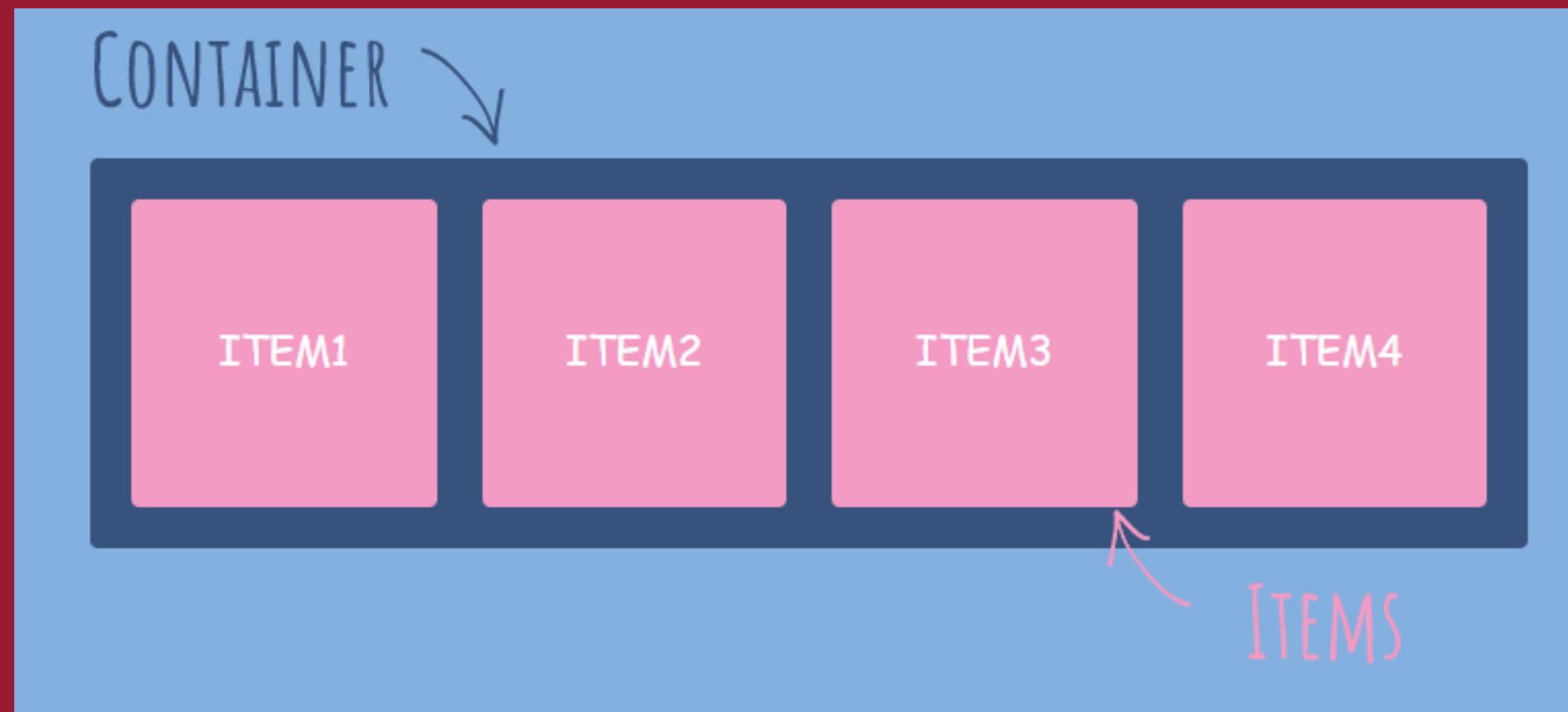
## The <div> Tag

We use the <div> tag to group two paragraphs for applying background to the text, and to add color to this word we place it within <span> tag.

Pay attention, that the <div> tag is a block-level element, so a line break is placed before and after it.

# CSS

## Think like Flexboxes



The screenshot shows a web browser window titled "The <div> Tag" with the URL "https://www.w3docs.com". The main content area displays the following text:

## The <div> Tag

We use the <div> tag to group two paragraphs for applying background to the text, and to add color to this word we place it within <span> tag.

Pay attention, that the <div> tag is a block-level element, so a line break is placed before and after it.

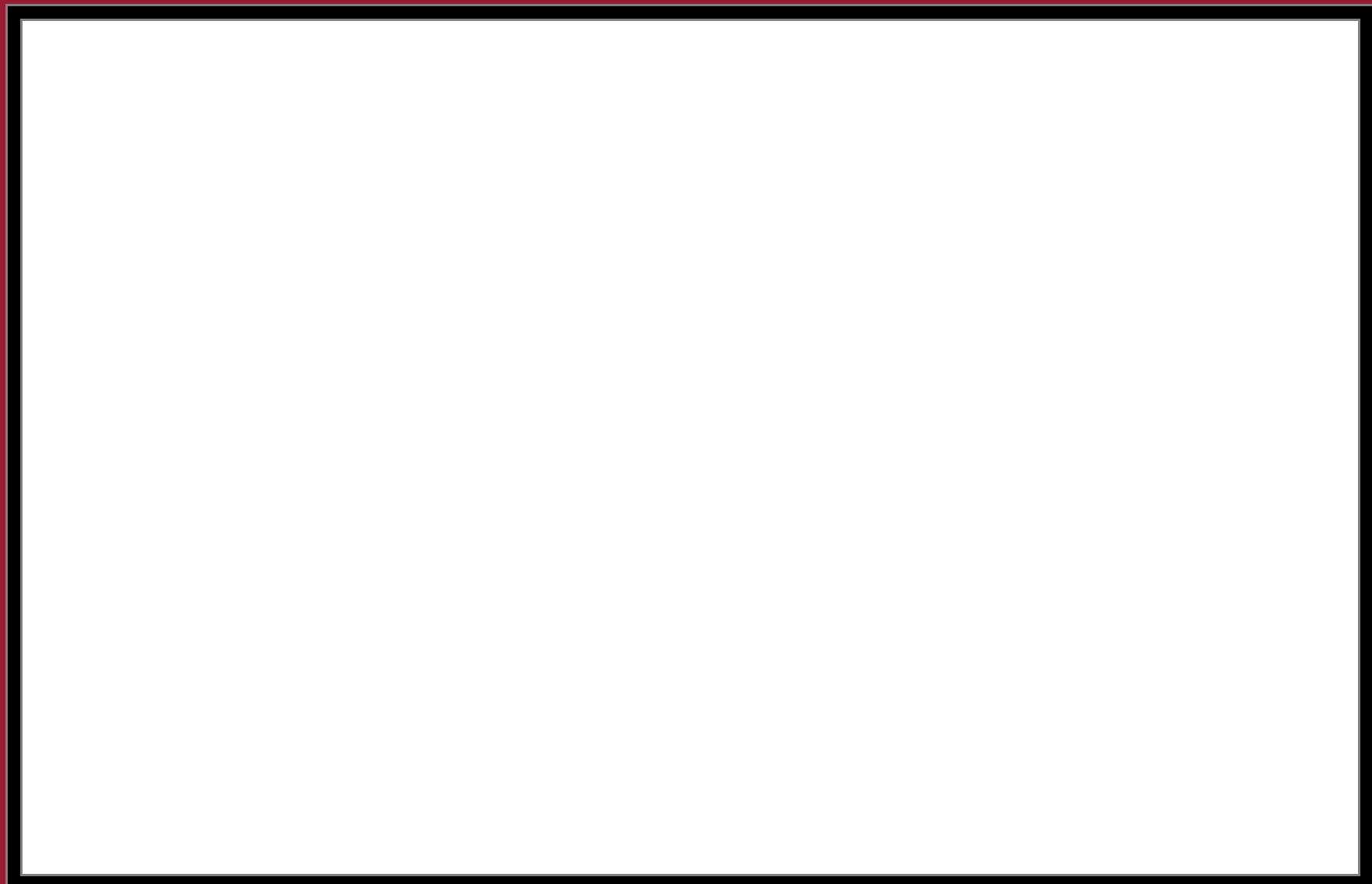
# CSS

## How do we build this?

Sample flexbox example							
First article	Second article	Third article	Fourth article	Fifth article	Sixth article	Seventh article	Eighth article
Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth photo booth	Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth photo booth	Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth photo booth	Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth photo booth	Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth photo booth	Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth photo booth	Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth photo booth	Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth photo booth

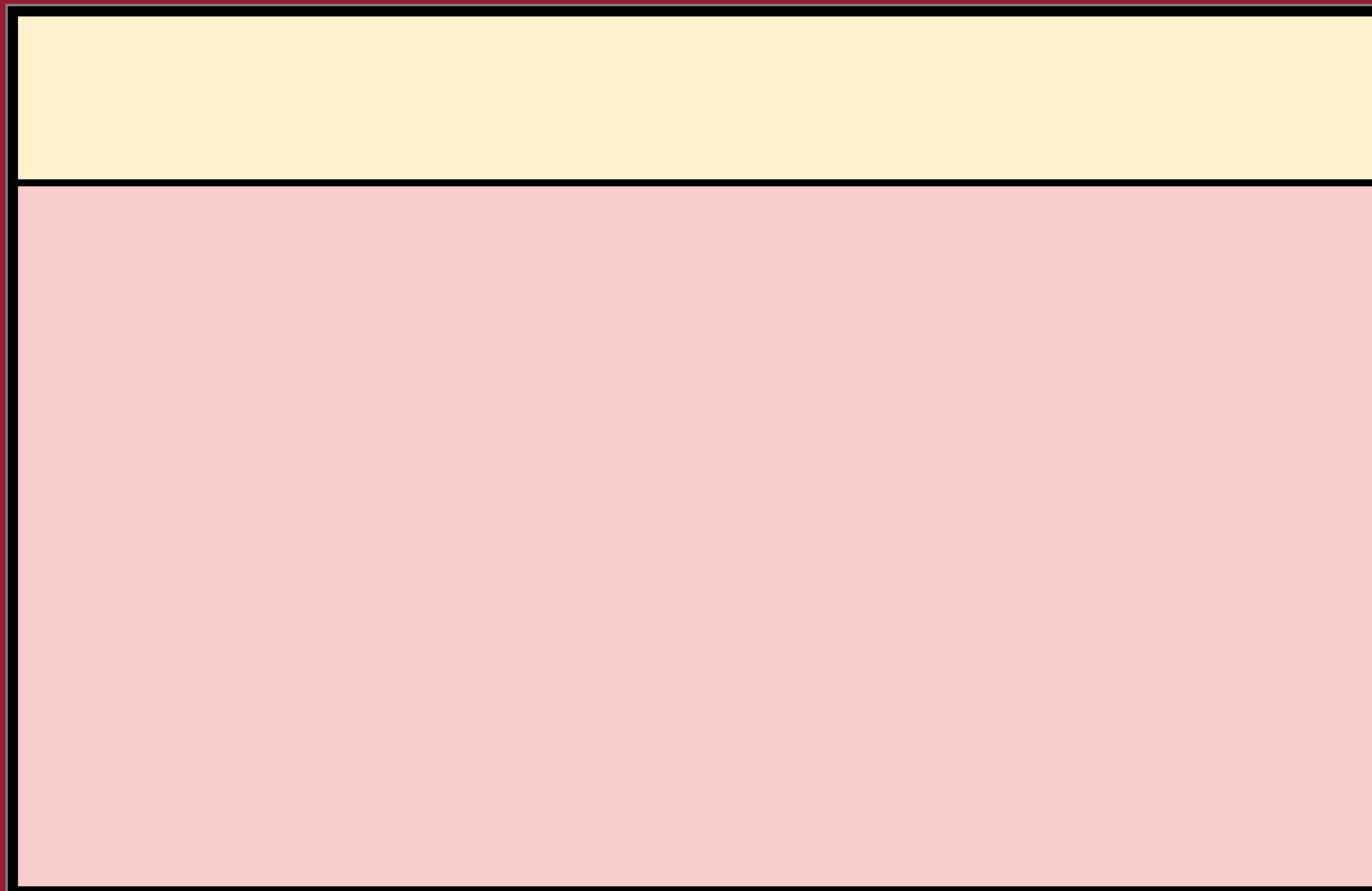
# CSS

## Think like Flexboxes



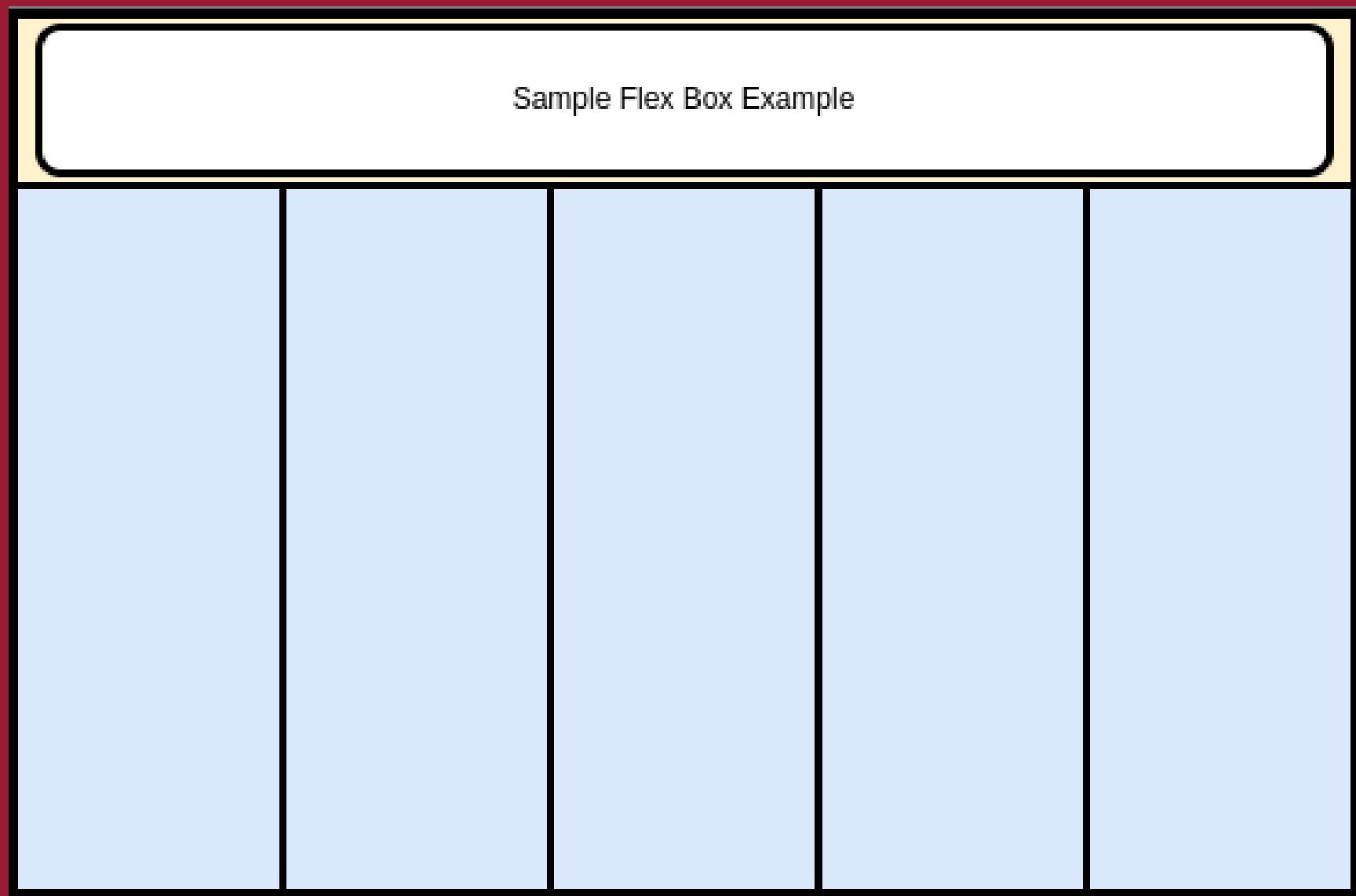
# CSS

## Think like Flexboxes



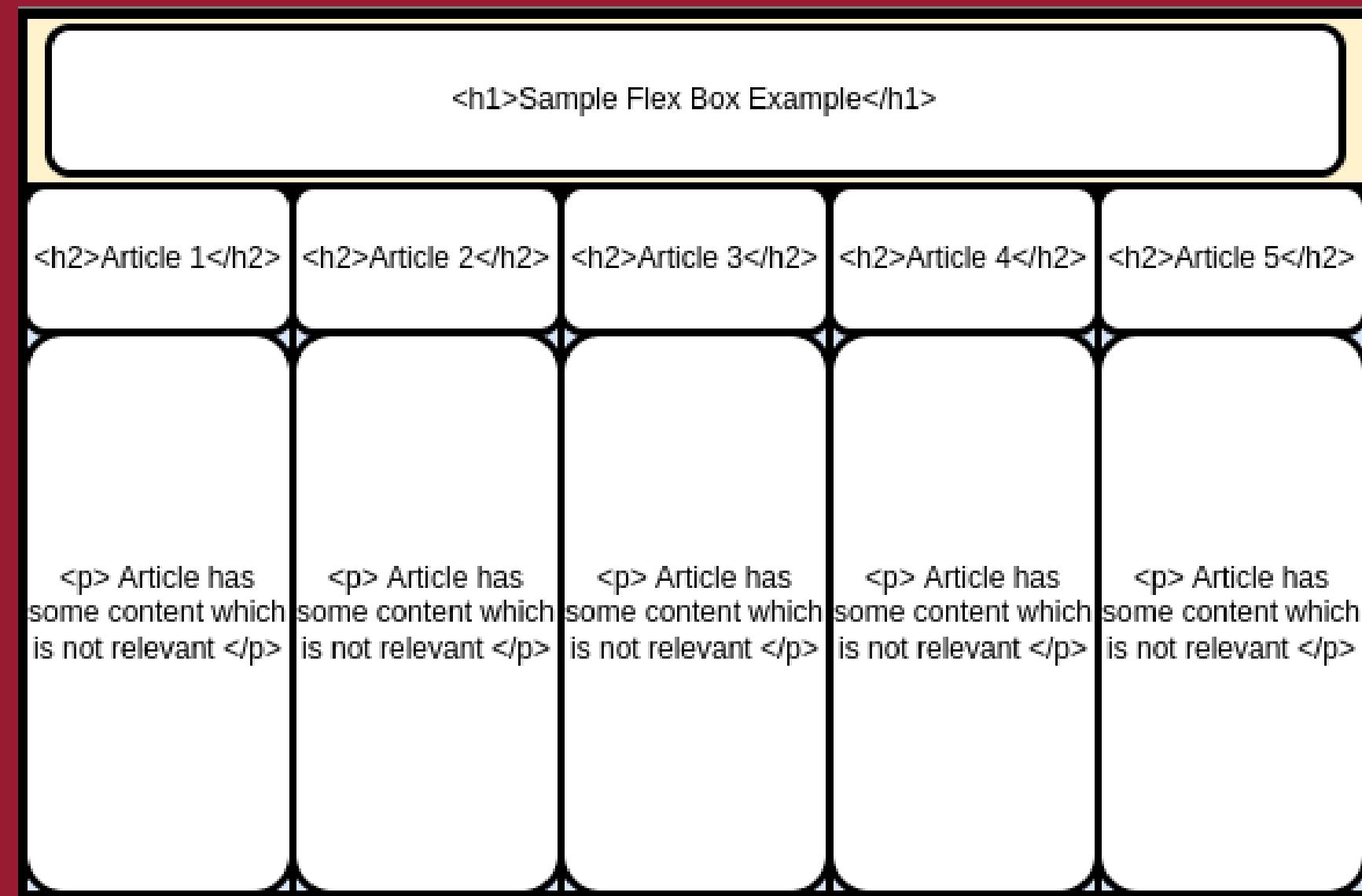
# CSS

## Think like Flexboxes



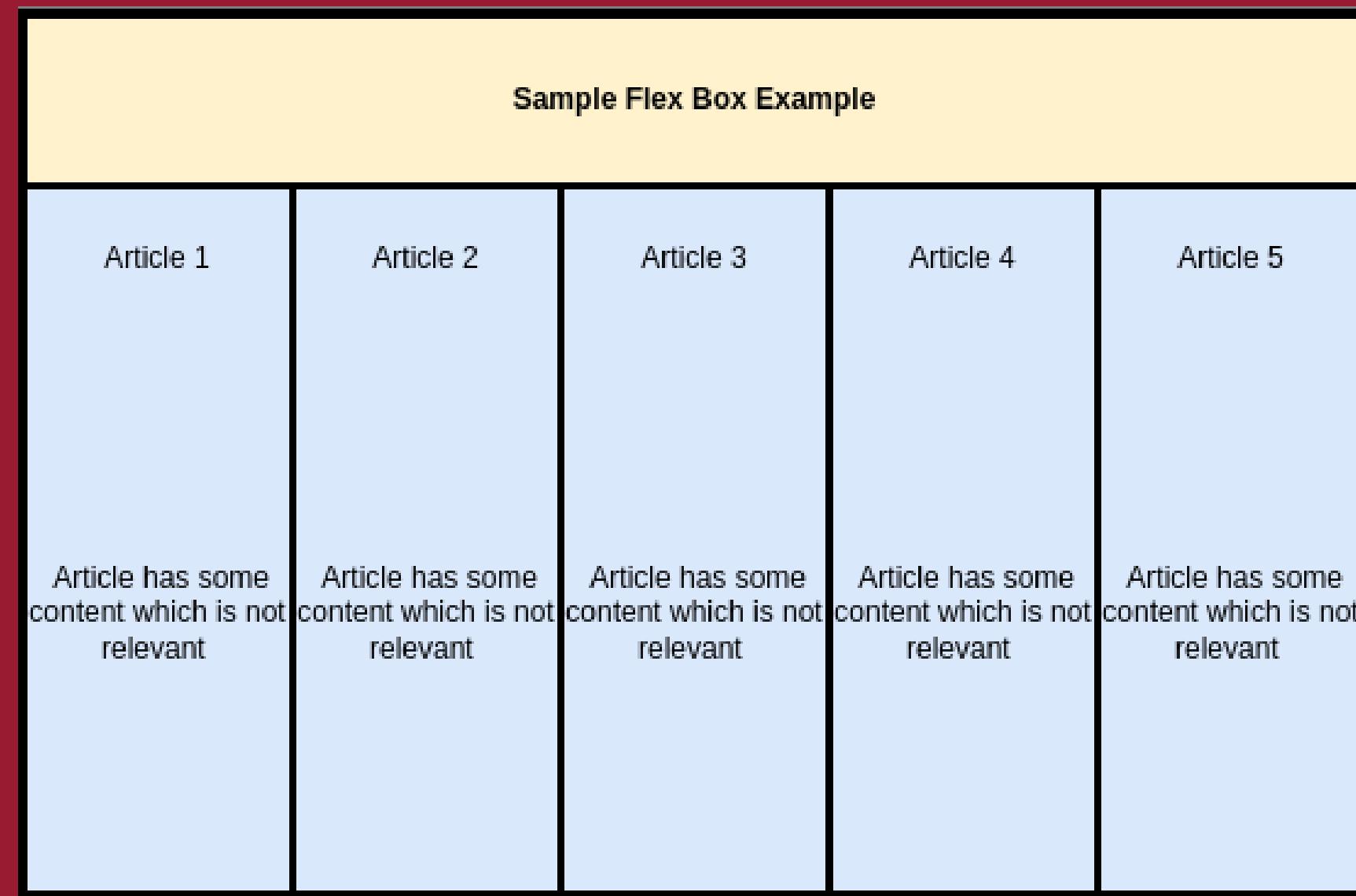
# CSS

## Think like Flexboxes



# CSS

## Think like Flexboxes



# CSS

## Think like Flexboxes

Sample Flex Box Example				
Article 1	Article 2	Article 3	Article 4	Article 5
Article has some content which is not relevant	Article has some content which is not relevant	Article has some content which is not relevant	Article has some content which is not relevant	Article has some content which is not relevant

Sample flexbox example							
First article	Second article	Third article	Fourth article	Fifth article	Sixth article	Seventh article	Eighth article
Tacos actually microdosing, pour-over semiotics banjo chickarrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth.	Tacos actually microdosing, pour-over semiotics banjo chickarrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth.	Tacos actually microdosing, pour-over semiotics banjo chickarrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth.	Tacos actually microdosing, pour-over semiotics banjo chickarrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth.	Tacos actually microdosing, pour-over semiotics banjo chickarrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth.	Tacos actually microdosing, pour-over semiotics banjo chickarrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth.	Tacos actually microdosing, pour-over semiotics banjo chickarrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth.	Tacos actually microdosing, pour-over semiotics banjo chickarrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth.

# JAVASCRIPT

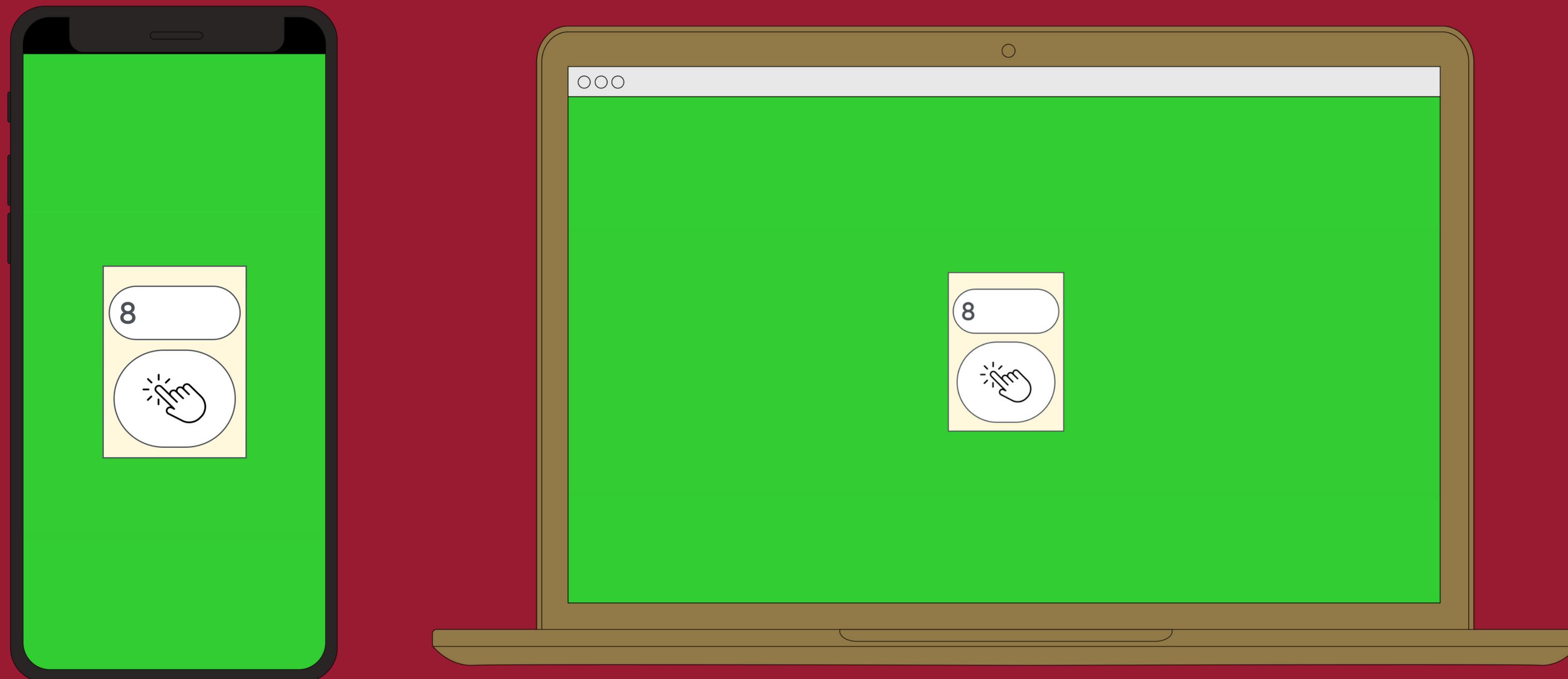
# HTML + CSS + JS

## Coding



# HTML + CSS + JS

## Coding



# HTML + CSS + JS

## Connecting CSS Sheet

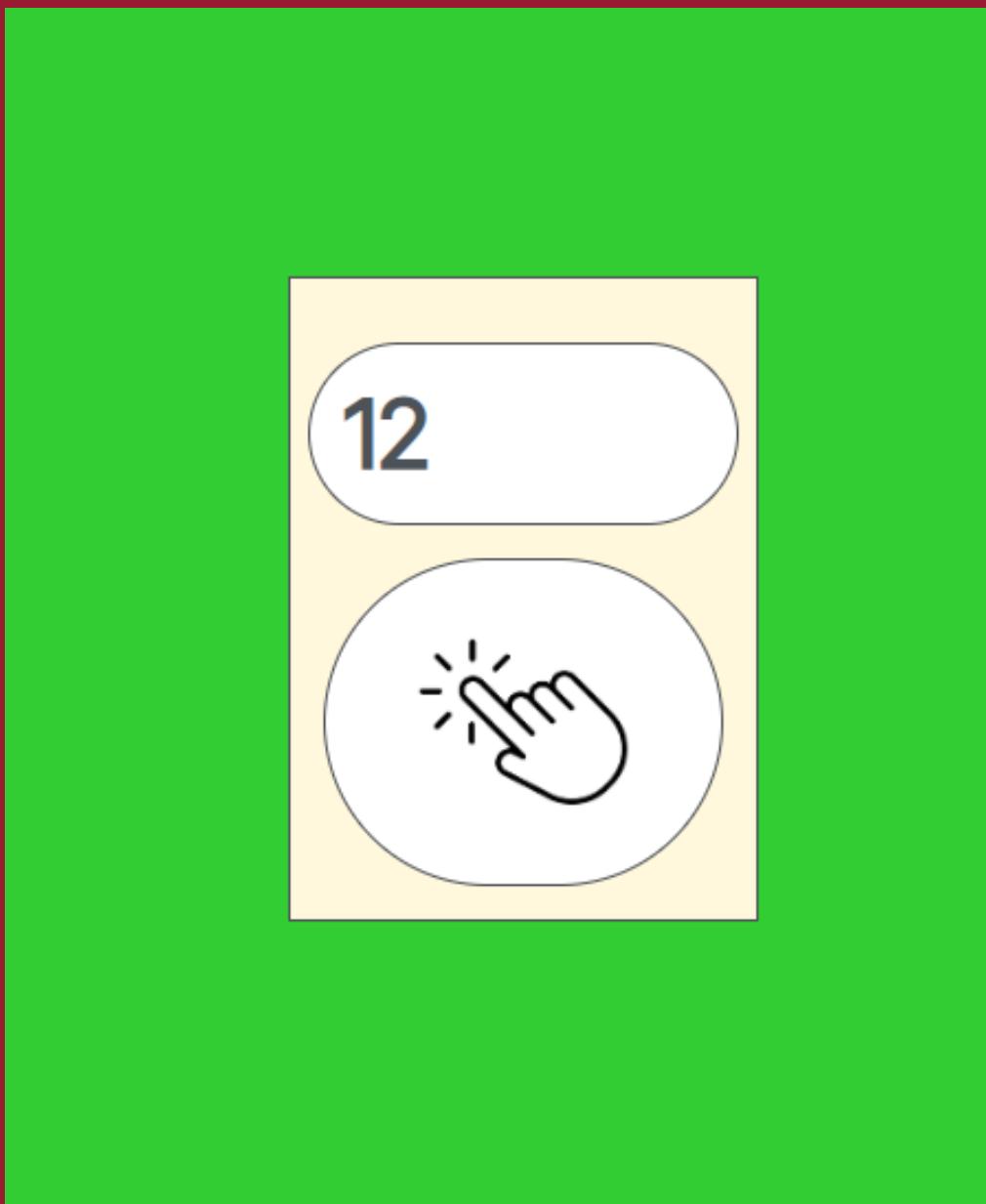


```
● ● ●  
1 <head>  
2   <meta charset="UTF-8" />  
3   <meta http-equiv="X-UA-Compatible" content="IE=edge" />  
4   <meta name="viewport" content="width=device-width, initial-scale=1.0" />  
5   <title>CICSoft Web Clicker</title>  
6   <link rel="stylesheet" href="styles.css" />  
7 </head>
```

A code editor window showing the beginning of an HTML file. It includes meta tags for character encoding, compatibility, and viewport settings, followed by a title and a link to a CSS file named 'styles.css'. A white arrow points from the text 'Connecting CSS Sheet' in the slide title down to the line containing the 'link' tag.

# HTML + CSS + JS

## Body

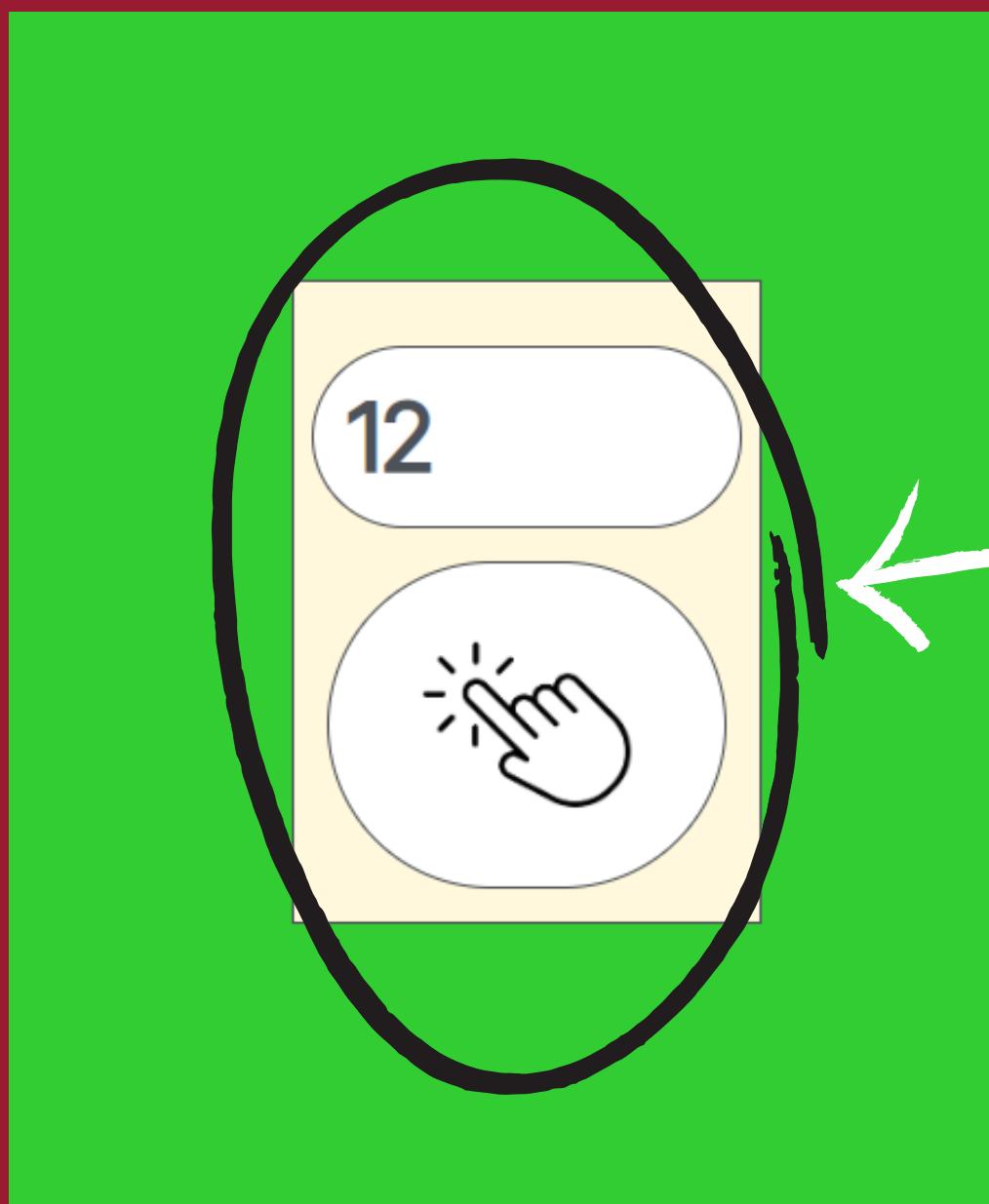


```
● ● ●  
1 <body>  
2   <div id="clicker-body">  
3     <div id="clicker-main">  
4       <div id="clicker-display" class="round" onclick="clear_click()">  
5         <span id="total">0</span>  
6       </div>  
7         
13     </div>  
14   </div>  
15   <script src="index.js"></script>  
16 </body>
```

A code editor window showing the HTML and JavaScript code for the clicker game. The code is written in a monospaced font. A white arrow points from the bottom right towards the closing tag of the script block.

# HTML + CSS + JS

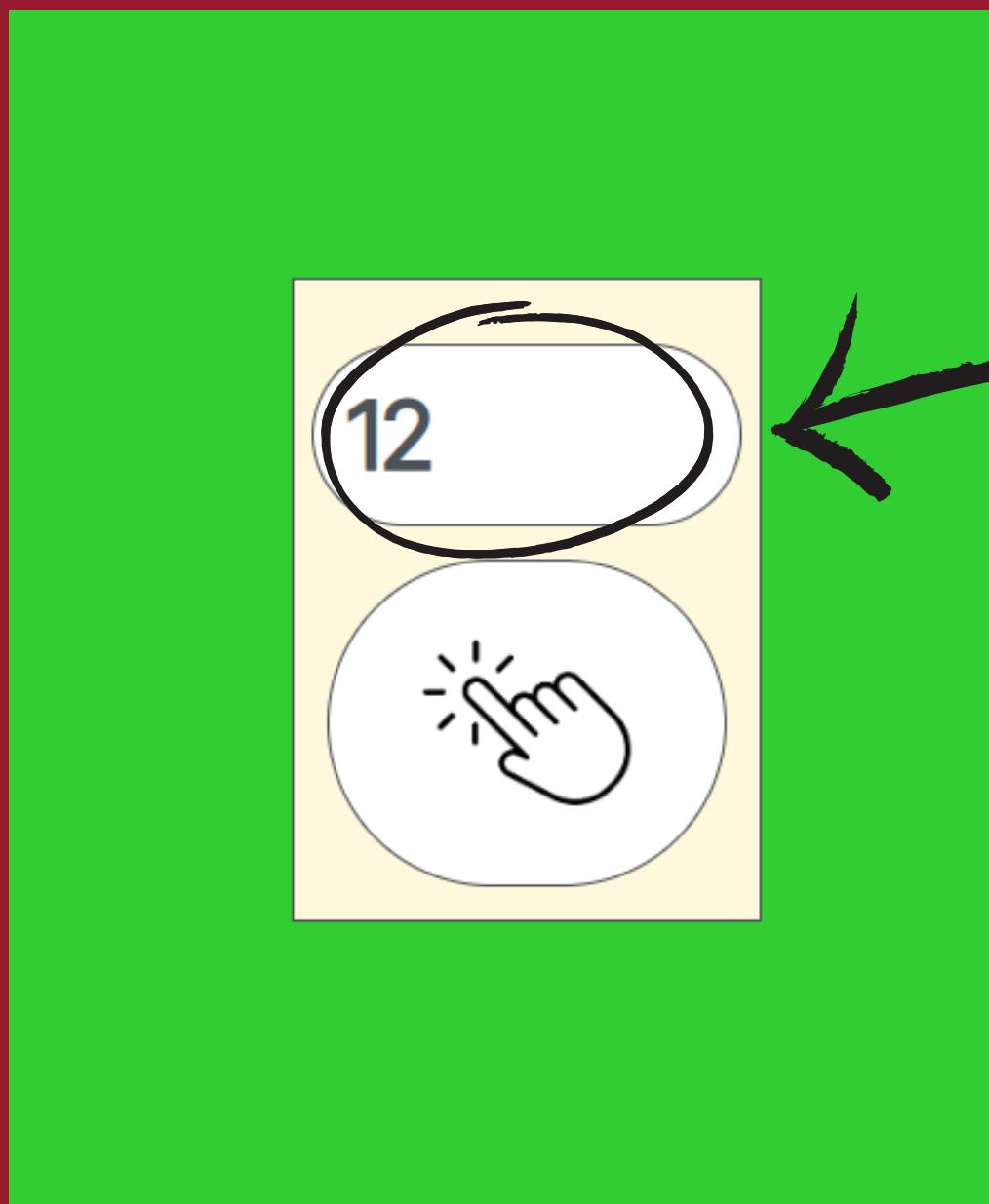
## Clicker Body



```
<body>
  <div id="clicker-body">
    <div id="clicker-main">
      <div id="clicker-dis...
        <span id="total">
        </div>
        
      </div>
    </div>
    <script src="index.js"></scr...
  </body>
```

# HTML + CSS + JS

## Clicker Display



```
<body>
  <div id="clicker-body">
    <div id="clicker-main">
      <div id="clicker-display" class="round">
        <span id="total">0</span>
      </div>
      
    </div>
  </div>
  <script src="index.js"></script>
</body>
```

A code editor window displaying the HTML and CSS for the clicker application. The code includes a body section with a clicker-body container. Inside, there's a clicker-main container holding a clicker-display element (a rounded div) with a total span (containing '0') and a click button (an img element with a round class, a premade URL for an image, a width of 150 pixels, and an onclick event that calls increment\_click). A script tag at the bottom points to index.js. A large white arrow points from the 'clicker-display' span in the code to the '12' value in the screenshot, indicating the connection between the code and the user interface.

# HTML + CSS + JS

## Clicker Display

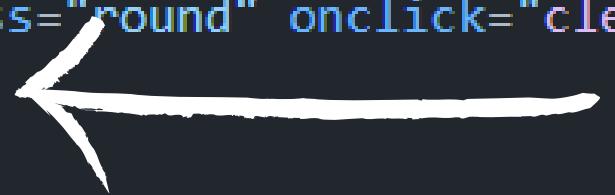
```
<body>
  <div id="clicker-body">
    <div id="clicker-main">
      <div id="clicker-display" class="round" onclick="clear_click()">
        <span id="total">0</span>
      </div>
      
    </div>
  </div>
  <script src="index.js"></script>
</body>
```

What's happening here?

# HTML + CSS + JS

## Clicker Display

```
<body>
  <div id="clicker-body">
    <div id="clicker-main">
      <div id="clicker-display" class="round" onclick="clear_click()">
        <span id="total">0</span>
      </div>
      
    </div>
  </div>
  <script src="index.js"></script>
</body>
```



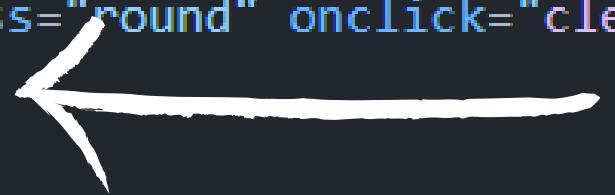
initial display

change this to 5 and see what happens?

# HTML + CSS + JS

## Clicker Display

```
<body>
  <div id="clicker-body">
    <div id="clicker-main">
      <div id="clicker-display" class="round" onclick="clear_click()">
        <span id="total">0</span>
      </div>
      
    </div>
  </div>
  <script src="index.js"></script>
</body>
```



initial display

changing this will affect the loading display  
not the functionality

# HTML + CSS + JS

## Clicker Display

```
<body>
  <div id="clicker-body">
    <div id="clicker-main">
      <div id="clicker-display" class="round" onclick="clear_click()">
        <span id="total">0</span>
      </div>
      
    </div>
  </div>
  <script src="index.js"></script>
</body>
```



Let's see what id, class, and onclick are doing in the div...

# HTML + CSS + JS

## Clicker Display

```
<body>
  <div id="clicker-body">
    <div id="clicker-main">
      <div id="clicker-display" class="round" onclick="clear_click()">
        <span id="total">0</span>
      </div>
      
    </div>
  </div>
  <script src="index.js"></script>
</body>
```

We already talked about id and class in the previous section.

Class is to group tags and apply effects to them and id is to define them uniquely

# HTML + CSS + JS

## Clicker Display

```
<body>
  <div id="clicker-body">
    <div id="clicker-main">
      <div id="clicker-display" class="round" onclick="clear_click()">
        <span id="total">0</span>
      </div>
      
    </div>
  </div>
  <script src="index.js"></script>
</body>
```



Before we look at  
onclick, let's focus on  
styling

# HTML + CSS + JS

## Body Styles



```
1 @import url("https://fonts.googleapis.com/css2?family=Inter+Tight&display=swap");
```



```
1 body {  
2     font-family: "Inter Tight";  
3     background-color: limegreen;  
4     margin: 0px;  
5 }
```



# HTML + CSS + JS

## Clicker Container

```
● ● ●  
1 #clicker-body {  
2   display: flex;  
3   width: 100vw;  
4   height: 100vh;  
5   justify-content: center;  
6   align-items: center;  
7 }
```



# HTML + CSS + JS

## Clicker Body

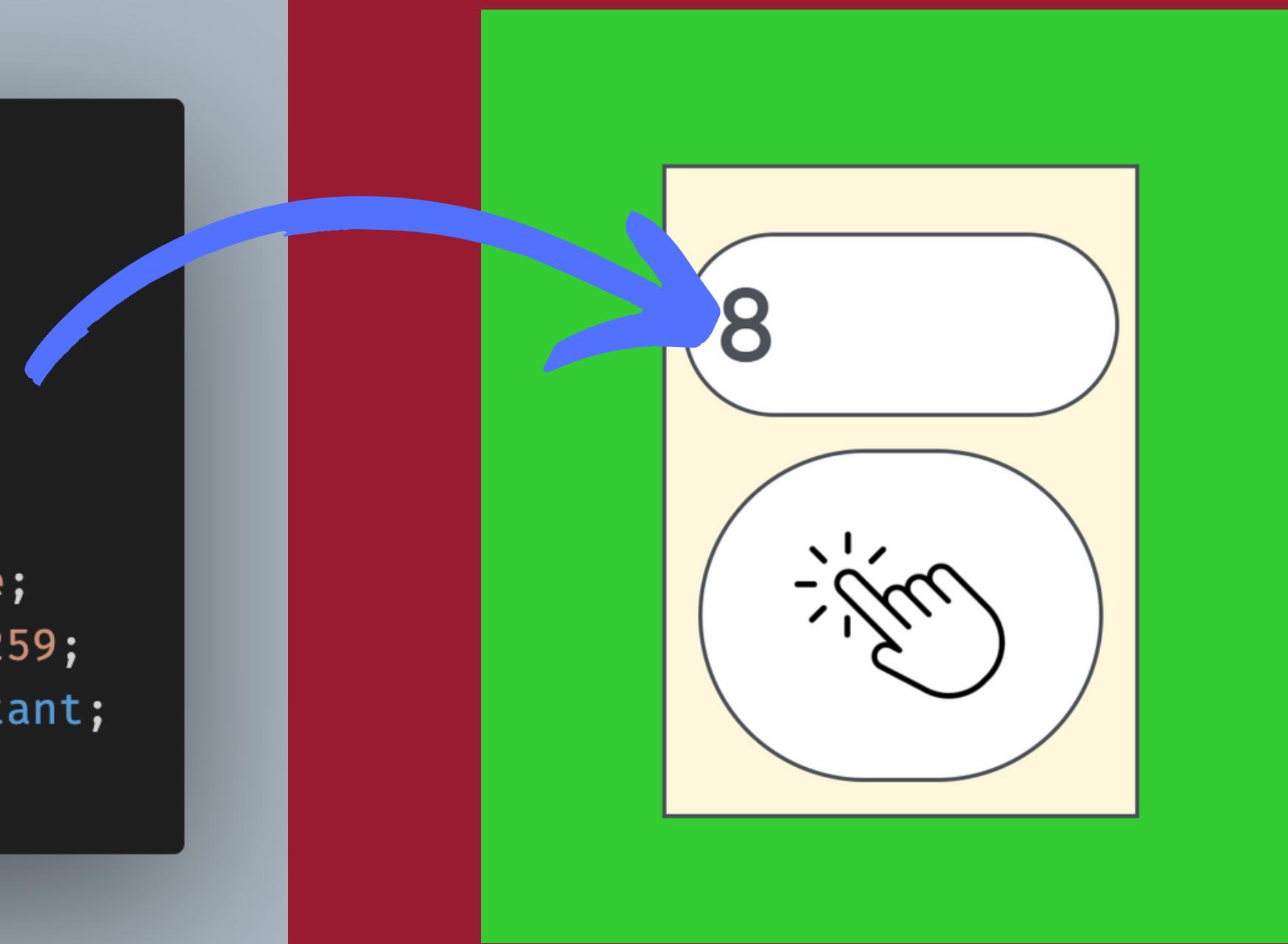
```
● ● ●  
1 #clicker-main{  
2   display: flex;  
3   flex-direction: column;  
4   justify-content: center;  
5   align-items: center;  
6   padding: 1em;  
7   background-color: cornsilk;  
8   border: 2px solid #4d5259;  
9 }
```



# HTML + CSS + JS

## Clicker Display Panel

```
● ● ●  
1 #clicker-display {  
2   display: flex;  
3   width: 90%;  
4   padding: 0.5rem 1rem;  
5   margin: 1rem 0rem;  
6   background-color: white;  
7   border: 2px solid #4d5259;  
8   font-size: 3rem !important;  
9 }
```



# HTML + CSS + JS

## Clicker Rounded Components

The image shows a screenshot of a Clicker game interface. On the left, there is a black sidebar with three colored dots (red, yellow, green) at the top. Below the dots is a block of CSS code. On the right, there is a green background area containing a yellow rectangular button with a white rounded rectangle inside it. The number '8' is displayed on the button. Below the button is a circular icon with a hand cursor pointing at it. Three blue curved arrows point from the CSS code on the left to the yellow button and the circular icon on the right, highlighting specific styling rules.

```
1 .round {  
2     font-size: 12px;  
3     color: #4d5259;  
4     line-height: 1.5;  
5     font-weight: bold;  
6     padding: 0.5em 2em;  
7     background: #ffffff;  
8     border: 2px solid #4d5259;  
9     border-radius: 100px;  
10 }  
11  
12 .round:hover {  
13     cursor: pointer;  
14     box-shadow: 4px 4px 0 0 #4d5259;  
15 }
```

# HTML + CSS + JS

## Clicker Display

```
<body>
  <div id="clicker-body">
    <div id="clicker-main">
      <div id="clicker-display" class="round" onclick="clear_click()">
        <span id="total">0</span>
      </div>
      
    </div>
  </div>
  <script src="index.js"></script>
</body>
```



Now, let's look at  
onclick

# HTML + CSS + JS

## onclick

```
<body>
  <div id="clicker-body">
    <div id="clicker-main">
      <div id="clicker-display" class="round" onclick="clear_click()">
        <span id="total">0</span>
      </div>
      
    </div>
  </div>
  <script src="index.js"></script>
</body>
```

# HTML + CSS + JS

## onclick

```
<body>
  <div id="clicker-body">
    <div id="clicker-main">
      <div id="clicker-display" class="round" onclick="clear_click()">
        <span id="total">0</span>
      </div>
      
    </div>
  </div>
  <script src="index.js"></script>
</body>
```

when the element is clicked, we execute the clear\_click() function

# HTML + CSS + JS

## onclick

```
<body>
  <div id="clicker-body">
    <div id="clicker-main">
      <div id="clicker-display" class="round" onclick="clear_click()">
        <span id="total">0</span>
      </div>
      
    </div>
  </div>
  <script src="index.js"></script>
</body>
```

what is clear click and  
where do we define it?

# HTML + CSS + JS

## onclick

```
<body>
  <div id="clicker-body">
    <div id="clicker-main">
      <div id="clicker-display" class="round" onclick="clear_click()">
        <span id="total">0</span>
      </div>
      
    </div>
  </div>
  <script src="index.js"></script>
</body>
```



in a js file linked here

# HTML + CSS + JS

## our JS file

```
count = 0;
increment = 1;

function increment_click() {
    count += increment;
    document.getElementById("total").innerHTML = `${count}`;
}

function clear_click() {
    count = 0;
    document.getElementById("total").innerHTML = `${count}`;
}
```

# HTML + CSS + JS

## our JS file

```
function clear_click() {  
    count = 0;  
    document.getElementById("total").innerHTML = `${count}`;  
}
```

our function sets the count variable to 0 and assigns the total element in the HTML

file to be 0

```
<div id="clicker-display" class="round" onclick="clear_click()">  
    <span id="total">0</span>  
</div>
```

# HTML + CSS + JS

## our JS file

```
function clear_click() {  
    count = 0;  
    document.getElementById("total").innerHTML = `${count}`;  
}
```

What happens if I change  
count to 100?

```
<div id="clicker-display" class="round" onclick="clear_click()">  
    <span id="total">0</span>  
</div>
```

# HTML + CSS + JS

## our JS file

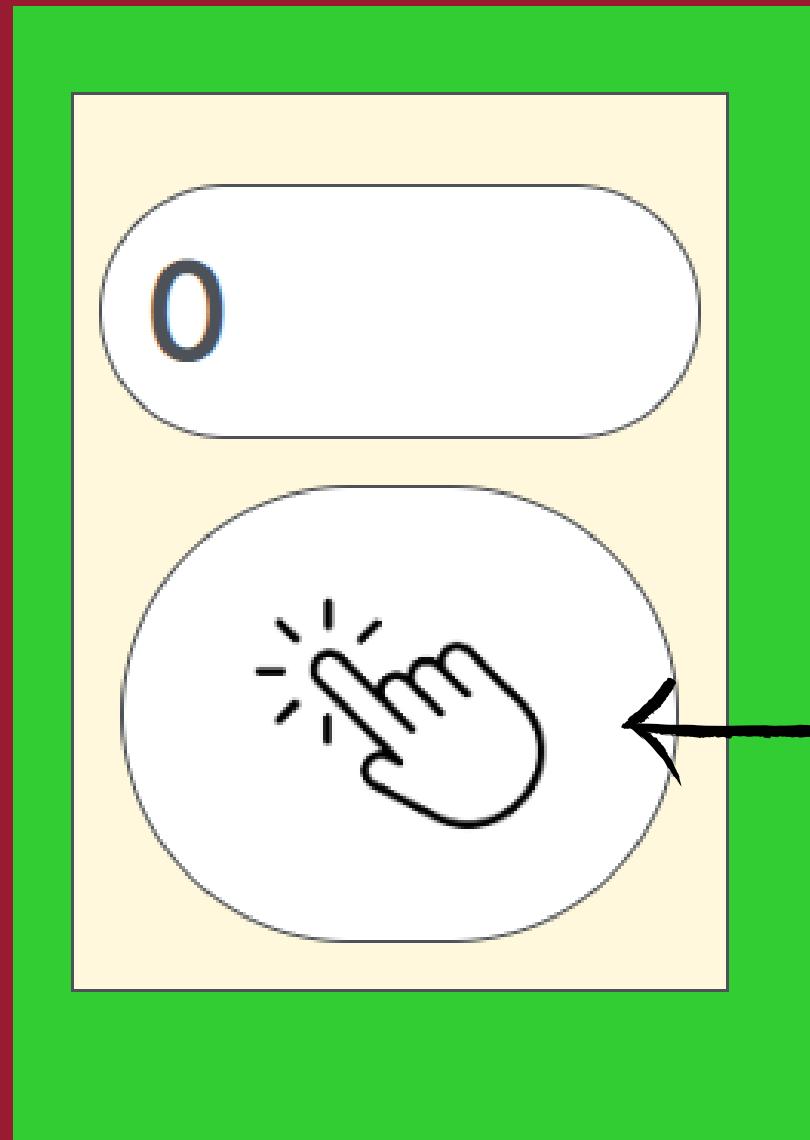
```
function clear_click() {  
    count = 0;  
    document.getElementById("total").innerHTML = `${count}`;  
}
```

What happens if I change  
count to count+100?

```
<div id="clicker-display" class="round" onclick="clear_click()">  
    <span id="total">0</span>  
</div>
```

# HTML + CSS + JS

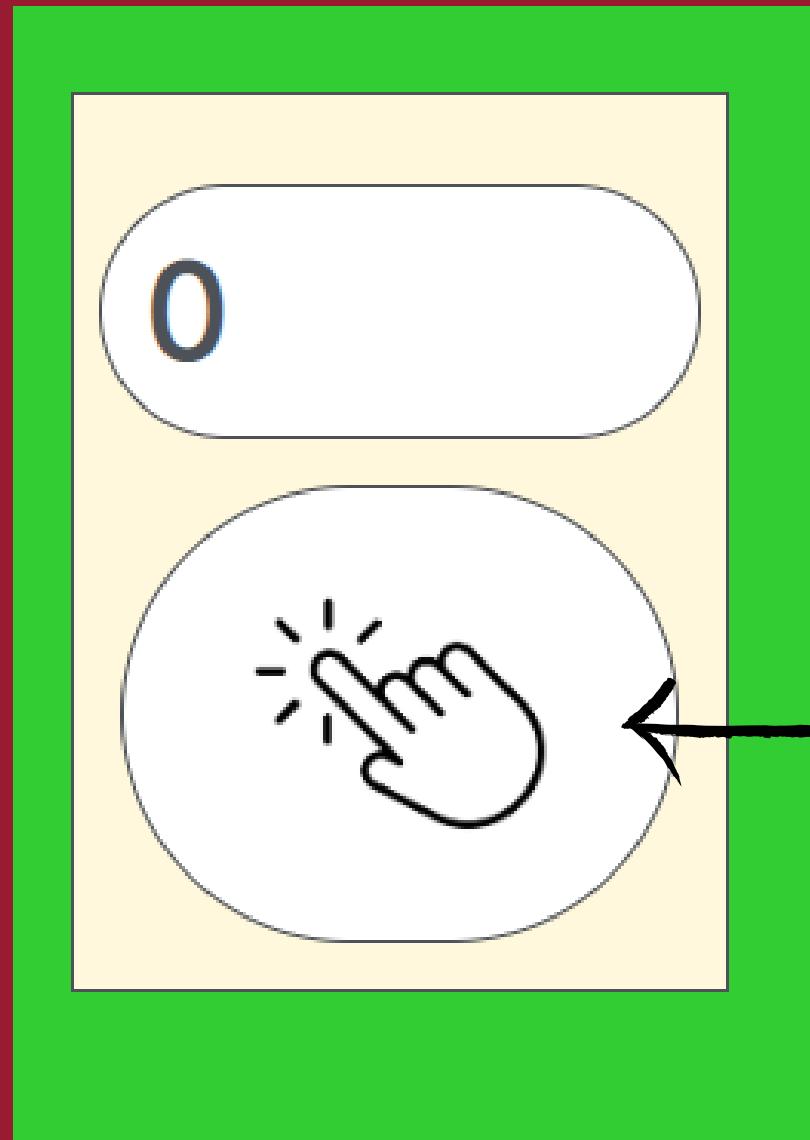
## Button



```
<div id="clicker-body">
  <div id="clicker-main">
    <div id="clicker-display" class="round" onclick="clear_click()">
      <span id="total">0</span>
    </div>
    
  </div>
</div>
<script src="index.js"></script>
```

# HTML + CSS + JS

## Button



```
<div id="clicker-body">
  <div id="clicker-main">
    <div id="clicker-display" class="round" onclick="clear_click()">
      <span id="total">0</span>
    </div>
    
  </div>
</div>
<script src="index.js"></script>
```

The button is our image tag

# HTML + CSS + JS

## Button

```
<div id="clicker-body">
  <div id="clicker-main">
    <div id="clicker-display" class="round" onclick="clear_click()">
      <span id="total">0</span>
    </div>
    
  </div>
</div>
<script src="index.js"></script>
```

We know what class, src and width are doing.

We also know onclick will make the element clickable.

But, what does increment\_click() do?

# HTML + CSS + JS

## Our JS file

```
count = 0;
increment = 1;

function increment_click() {
    count += increment;
    document.getElementById("total").innerHTML = `${count}`;
}
```

Count initially is 0 and increment is 1.  
Our increment\_click() function simply adds increment to count  
and assigns it to total's display.

# HTML + CSS + JS

## Our JS file

```
count = 0;
increment = 1;

function increment_click() {
    count += increment;
    document.getElementById("total").innerHTML = `${count}`;
}
```

what if we changed the increment to 50?

# YOUR TURN NOW

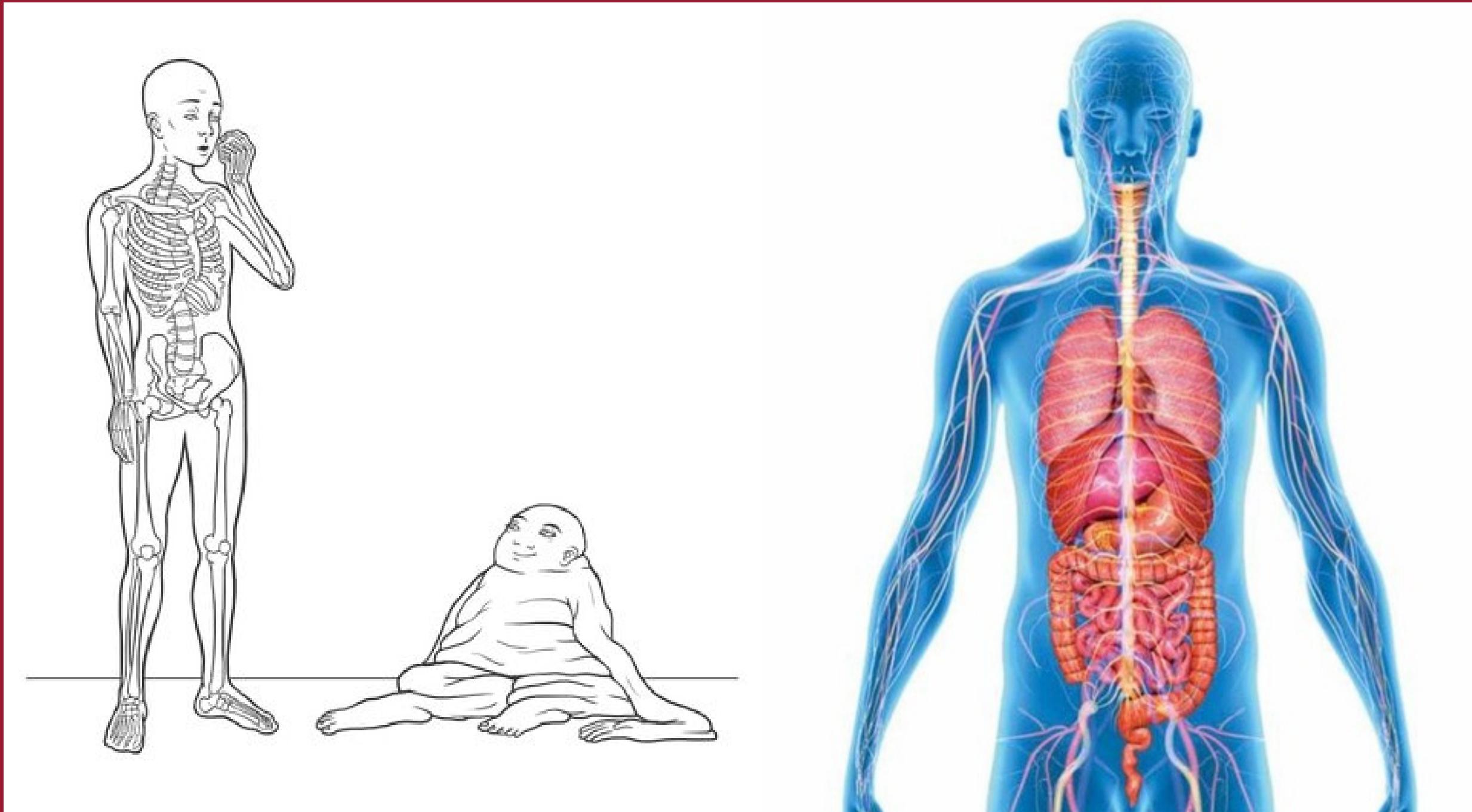
Add a button to reduce the count by increment's value.

i.e. increment = 10 and count = 100, the button's 2 clicks will display 80



# RECAP

# RECAP



HTML

CSS

JAVASCRIPT

# NEXT TIME

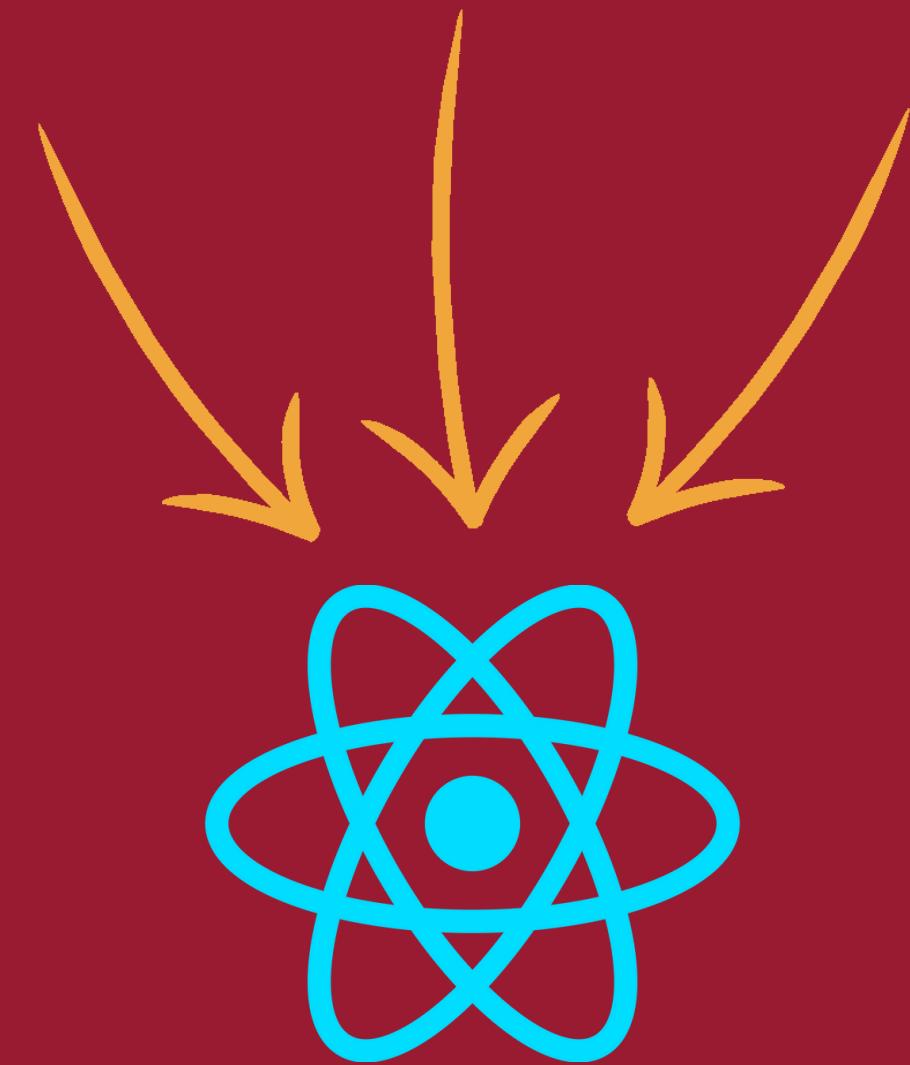
HTML



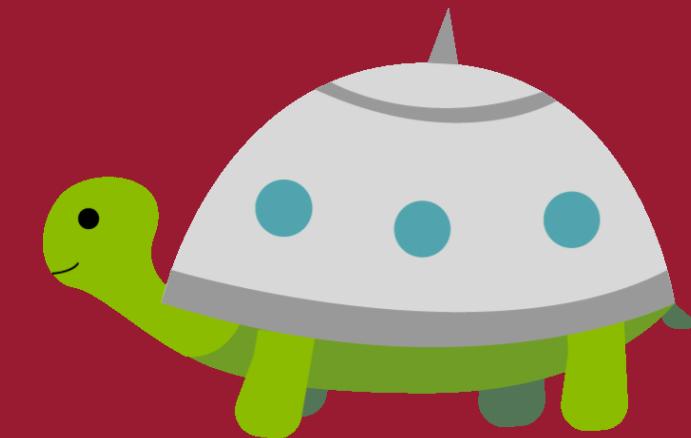
JS



CSS



ReDUCE  
ReUSE  
ReCYCLE



DEVELOPMENT  
TIME





Thank you for joining  
us today!