

COMP 1842

Week 4 part 2 – CSS Selectors

Matt Prichard

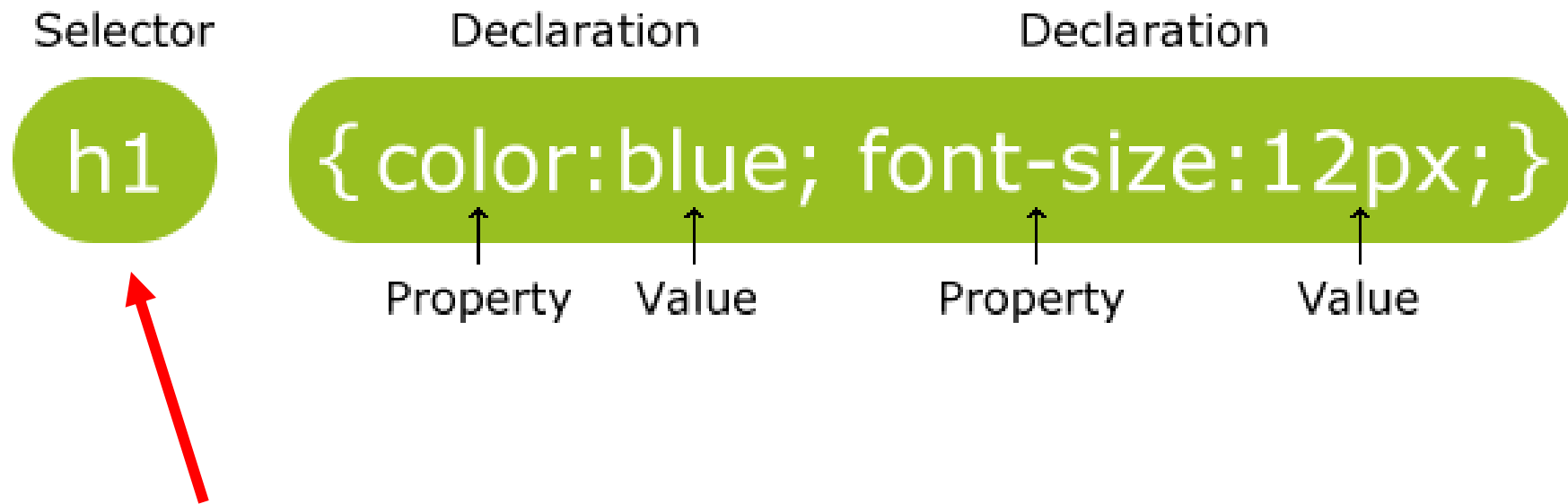
Introduction

The two hardest parts of beginning CSS are manipulating the box model without going mad and understanding CSS selectors.

The actual styling part is quite simple, picking fonts and colours etc.

<https://www.freecodecamp.org/news/css-selectors-cheat-sheet/>

Recap



The selector is how we target the element/s we wish to style

They start off simple but can become quite involved.

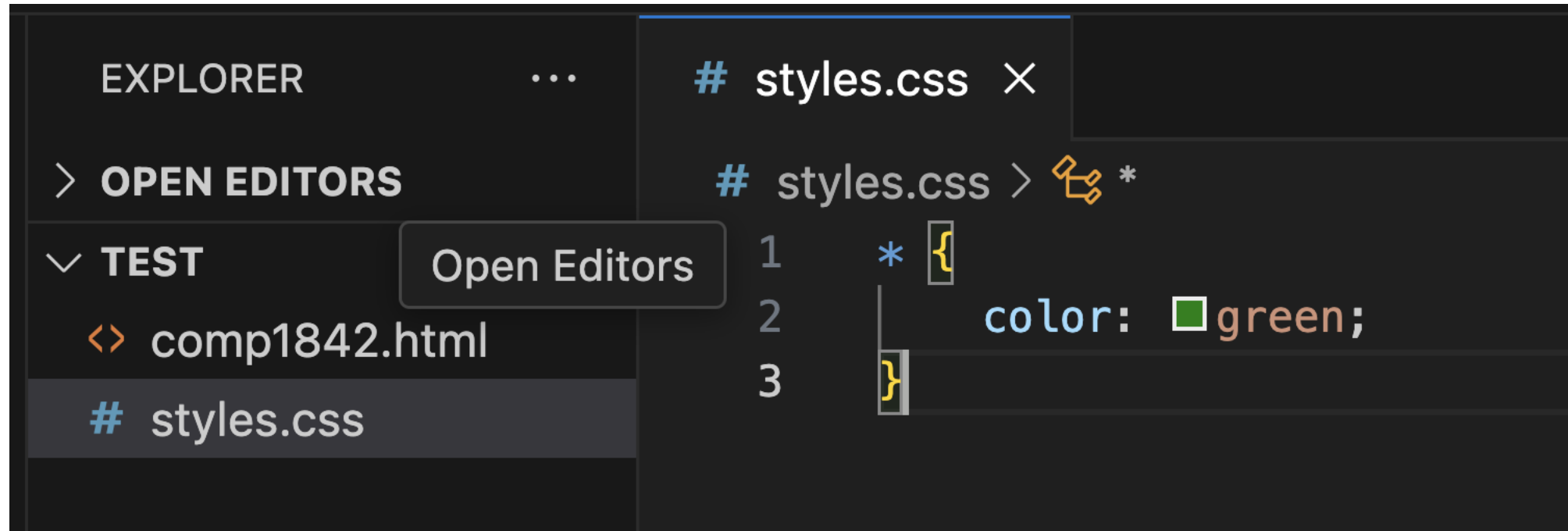
Universal selectors

```
/* Selects all elements */  
* {  
    color: green;  
}
```

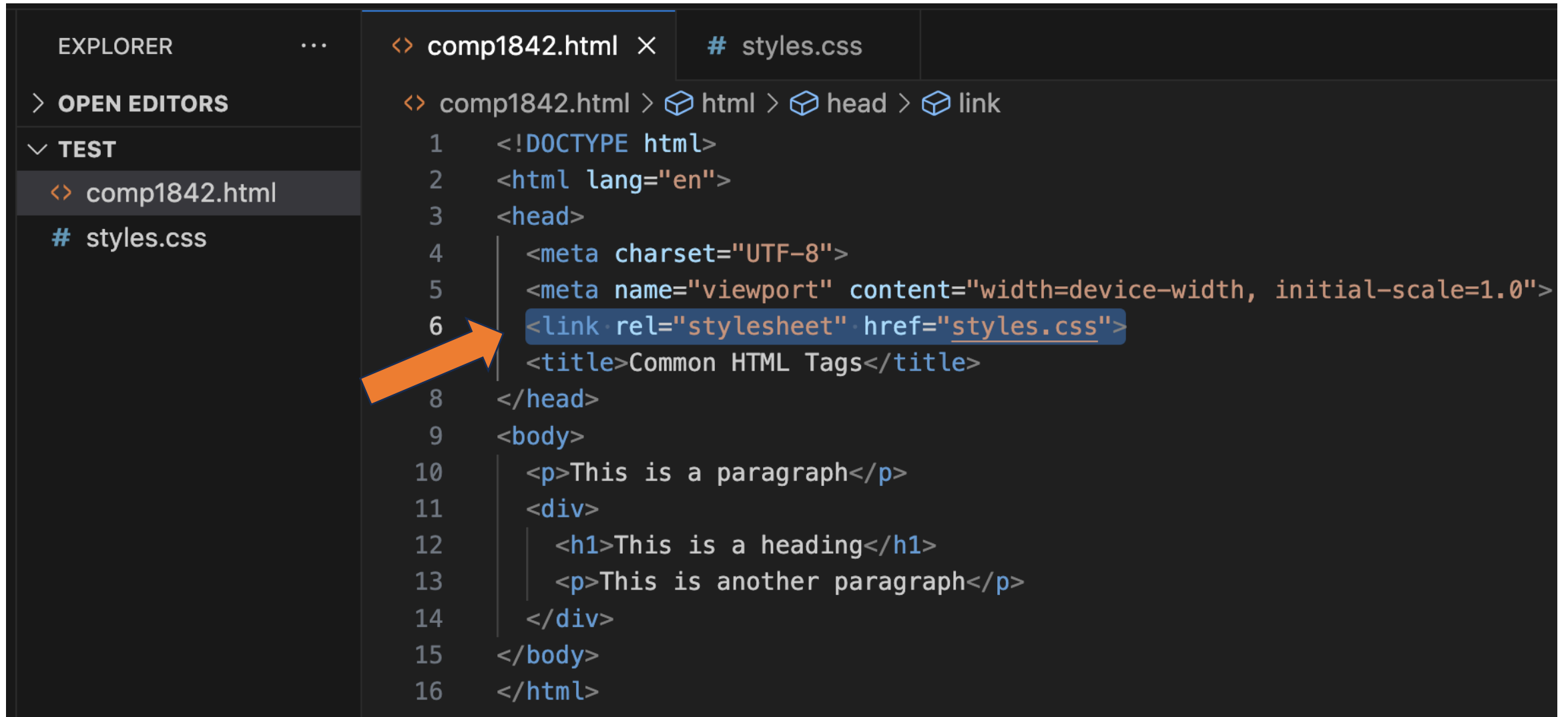
Select all elements inside <div> elements and set their background color to yellow:

```
div * {  
    background-color: yellow;  
}
```

Create css file



Insert external css



The screenshot shows the Visual Studio Code editor interface. On the left, the Explorer sidebar displays the file structure with 'comp1842.html' and 'styles.css' under a 'TEST' folder. The main editor area has two tabs: 'comp1842.html' and 'styles.css'. The 'comp1842.html' tab is active, showing an HTML document. The breadcrumb navigation at the top of the editor indicates the current position: 'comp1842.html > html > head > link'. The HTML code is as follows:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <link rel="stylesheet" href="styles.css">
7   <title>Common HTML Tags</title>
8 </head>
9 <body>
10  <p>This is a paragraph</p>
11  <div>
12    <h1>This is a heading</h1>
13    <p>This is another paragraph</p>
14  </div>
15 </body>
16 </html>
```

An orange arrow points to the line number 6, highlighting the newly added `<link rel="stylesheet" href="styles.css">` tag within the `<head>` section.

Result

- Ctrl + R: reload
- Ctrl + Shift + R: Hard reload

This is a paragraph

This is a heading

This is another paragraph

Element or tag selectors

```
/* All <a> elements. */  
a {  
    color: red;  
}
```

```
h1, h2, p {  
    text-align: right;  
    color: blue;  
}
```

CSS Element Selector, also called a Tag selector is a selector that selects all the HTML elements by a specified names on a web page. It is the most basic selector of all. Elements can be grouped.

styles.css ×

styles.css > h1

```
1  * {
2    color: green;
3  }
4
5  div * {
6    background-color: yellow;
7  }
8
9  a {
10   color: red;
11 }
12
13 h1, h2, p {
14   text-align: right;
15   color: blue;
16 }
```

<> comp1842.html ×

<> comp1842.html > html > body > h2

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1">
6    <link rel="stylesheet" href="styles.css">
7    <title>Common HTML Tags</title>
8  </head>
9  <body>
10   <p>This is a paragraph</p>
11   <div>
12     <h1>This is a heading</h1>
13     <p>This is another paragraph</p>
14   </div>
15   <a>This is a link</a>
16   <h1>This is h1</h1>
17   <h2>This is h2</h2>
18 </body>
19 </html>
```

This is a paragraph

This is a heading

This is another paragraph

This is a link

This is h1

This is h2

Class selectors

```
.red {  
  color: #f33;  
}  
  
.yellow-bg {  
  background: #ffa;  
}
```

```
<p class="red">This paragraph has red text.</p>  
<p class="red yellow-bg">  
  This paragraph has red text and a yellow background.  
</p>
```

Selects all elements with the 'red' class. The syntax is a dot in CSS and the word class in the html, again these can be combined.

```
.red {  
  color: ■ #f33  
}  
  
.yellow-bg {  
  background: ■ #ffa;  
}
```

```
16 <h1>This is h1</h1>  
17 <h2>This is h2</h2>  
18 <p class="red">This paragraph has red text</p>  
19 <p class="red yellow-bg">This paragraph has red text and yellow  
   background</p>  
20 </body>  
21 </html>
```

This paragraph has red text

This paragraph has red text and yellow background

Id selectors

```
#identified {  
  background-color: skyblue;  
}
```

This div has a special ID on it!
This is just a regular div.

```
<div id="identified">This div has a special ID on it!</div>  
<div>This is just a regular div.</div>
```

The id selector uses the id attribute of an HTML element to select a specific element. The id of an element is unique within a page, so the id selector is used to select one unique element!

```
.yellow-bg {  
  background: ■ #ffa;  
}  
  
#identifier {  
  background-color: ■ skyblue;  
}
```

```
16 <h1>This is h1</h1>  
17 <h2>This is h2</h2>  
18 <p class="red">This paragraph has red text</p>  
19 <p class="red yellow-bg">This paragraph has red text and yellow  
background</p>  
20 <div id="identifier">This div has a specific ID on it!</div>  
21 </body>  
22 </html>
```

This div has a specific ID on it!

Attribute selectors

The [attribute] selector is used to select elements with a specified attribute. Very useful for styling forms.

```
<style>
input[type=text] {
  width: 150px;
  display: block;
  margin-bottom: 10px;
  background-color: yellow;
}

input[type=button] {
  width: 120px;
  margin-left: 35px;
  display: block;
}
```

Firstname:

Peter

Lastname:

Griffin

Example Button

```
input[type="text"] {  
  width: 150px;  
  display: block;  
  margin-bottom: 10px;  
  background-color: yellow;  
}
```

```
input[type="button"] {  
  width: 120px;  
  margin-left: 35px;  
  display: block;  
}
```

```
20 <div id="identifier">This div has a specific ID on it!</div>  
21 <div>Username</div>  
22 <input type="text" placeholder="Enter your name">  
23 <div>Password</div>  
24 <input type="password" placeholder="Enter your password">  
25 <input type="button" value="Submit">  
26 </body>  
27 </html>
```

Username

Password

Combinators

Descendent combinator

The descendant selector matches all elements that are descendants of a specified element.

The following example selects all <p> elements inside <div> elements:

```
div p {  
    background-color: yellow;  
}
```

```
div p {  
  background-color: yellow;  
}
```

```
25 <input type="button" value="Submit">  
26 <div>Normal div</div>  
27 <p>Normal paragraph</p>  
28 <div>  
29   <p>p tag inside div tag</p>  
30 </div>  
31 </body>  
32 </html>
```

Normal div

Normal paragraph

p tag inside div tag

Descendent combinator

selects all elements that are descendant (or child) of the elements with div tag and 'row' class

```
div.row * {  
    background-color: yellow;  
}
```

```
<div class="row">  
  <p>Paragraph 1 in the div.</p>  
  <p>Paragraph 2 in the div.</p>  
  <section>section 3 in the div.</section>  
</div>
```

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div.

```
div.row * {
```

```
  background-color: plum;
```

```
}
```

```
30 </div>
```

```
31 <div class="row">
```

```
32   <p>Paragraph 1 in the div</p>
```

```
33   <p>Paragraph 2 in the div</p>
```

```
34   <section>Section 3 in the div</section>
```

```
35 </div>
```

```
36 </body>
```

Paragraph 1 in the div

Paragraph 2 in the div

Section 3 in the div

Child combinator

The child combinator (>) is placed between two CSS selectors. It matches only those elements that are the direct children of the first.

```
div > p{  
    background-color: yellow;  
}
```

```
<div>  
  <p>Paragraph 1 in the div.</p>  
  <p>Paragraph 2 in the div.</p>  
  <section><p>paragraph 3 in a section so not a child</p></section>  
  <p>paragraph 4 in the div</p>  
</div>  
<p>Paragraph 5. Not in a div.</p>
```

Paragraph 1 in the div.

Paragraph 2 in the div.

paragraph 3 in a section so not a child

paragraph 4 in the div

Paragraph 5. Not in a div.

```
background-color: yellow;
```

```
div.row * {
```

```
background-color: plum;
```

```
div > p {
```

```
background-color: skyblue;
```

```
}
```

```
31 <div class="row">
```

```
32   <p>Paragraph 1 in the div</p>
```

```
33   <p>Paragraph 2 in the div</p>
```

```
34   <section>Section 3 in the div</section>
```

```
35 </div>
```

```
36 <div>
```

```
37   <p>Paragraph 1</p>
```

```
38   <p>Paragraph 2</p>
```

```
39   <section>
```

```
40     <p>Paragraph 3</p>
```

```
41   </section>
```

```
42   <p>Paragraph 4</p>
```

```
43 </div>
```

```
44 <p>Paragraph 5</p>
```

```
45 </body>
```

Paragraph 1

Paragraph 2

Paragraph 3

Paragraph 4

Paragraph 5

Adjacent Sibling Selector (+)

It selects the element that is immediately preceded by the former element. Sibling elements must have the same parent element, and "adjacent" means "immediately following".

```
div + p {  
  background-color: yellow;  
}
```

```
15  <section>  
16    <div>  
17      <p>Paragraph 1 in the div.</p>  
18      <p>Paragraph 2 in the div.</p>  
19    </div>  
20    <p>Paragraph 3. After a div.</p>  
21    <p>Paragraph 4. After a div.</p>  
22    <div>  
23      <p>Paragraph 5 in the div.</p>  
24      <p>Paragraph 6 in the div.</p>  
25    </div>  
26    <p>Paragraph 7. After a div.</p>  
27    <p>Paragraph 8. After a div.</p>  
28  </section>
```

```
div + p {  
  background-color: orange;  
}
```

```
44 <p>Paragraph 5</p>  
45 <section>  
46   <div>  
47     <p>p1 in the div</p>  
48     <p>p2 in the div</p>  
49   </div>  
50   <p>p3 after a div</p>  
51   <p>p4 after a div</p>  
52   <div>  
53     <p>p5 in the div</p>  
54     <p>p6 in the div</p>  
55   </div>  
56   <p>p7 after a div</p>  
57   <p>p8 after a div</p>  
58 </section>  
59 </body>
```

p1 in the div

p2 in the div

p3 after a div

p4 after a div

p5 in the div

p6 in the div

p7 after a div

p8 after a div

General Sibling Selector (~)

The general sibling selector (~) selects all elements that are next siblings of a specified element. The following example selects all <p> elements that are next siblings of <div> elements:

```
div ~ p {  
  background-color: yellow;  
}
```

```
<body>  
  <p>Paragraph 1.</p>  
  <div>  
    <p>Paragraph 2.</p>  
  </div>  
  <p>Paragraph 3.</p>  
  <code>Some code.</code>  
  <p>Paragraph 4.</p>  
</body>
```

```
div ~ p {  
  background-color: red;  
}
```

```
58 </section>  
59 <p>p1</p>  
60 <div>  
61 |   <p>p2</p>  
62 </div>  
63 <p>p3</p>  
64 <code>Some code</code>  
65 <p>p4</p>  
66 </body>  
67 </html>
```

p2

p3

Some code

p4

Pseudo
selectors &
Pseudo
classes

:first-child

The :first-child selector allows you to target the first element immediately inside another element.

```
p:first-child {  
  font-size: 1.5em;  
}
```

```
<article>  
  <p>First paragraph...</p>  
  <p>Lorem ipsum...</p>  
  <p>Dolor sit amet...</p>  
  <p>Consectetur adipisicing...</p>  
</article>
```

```
p:first-child {  
  font-size: 1.5em;  
}
```

```
65 <p>p4</p>  
66 <article>  
67   <p>P1</p>  
68   <p>P2</p>  
69   <p>P3</p>  
70   <p>P4</p>  
71 </article>  
72 </body>
```

P1

P2

P3

P4

:last-child

The :last-child selector allows you to target the last element immediately inside another element.

```
p:last-child {  
  font-size: 0.75em;  
}
```

```
<article>  
  <p>First paragraph...</p>  
  <p>Lorem ipsum...</p>  
  <p>Dolor sit amet...</p>  
  <p>Consectetur adipisicing...</p>  
</article>
```

:nth-child()

Target the nth element immediately inside (or child of) another element.

Admits integers, even, odd, or formulas

```
/* Select the first list item */  
li:nth-child(1) { }  
  
/* Select the 5th list item */  
li:nth-child(5) { }  
  
/* Select every other list item starting with first */  
li:nth-child(odd) { }  
  
/* Select every 3rd list item starting with first */  
li:nth-child(3n - 2) { }  
  
/* Select every 3rd list item starting with 2nd */  
li:nth-child(3n - 1) { }
```

<https://css-tricks.com/almanac/selectors/n/nth-child/>

`a:not(.name)`

Selects all a elements that are not of the .name class

This can go on and on

Scary full list

<https://css-tricks.com/almanac/>

<https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>

Selectors

CSS selectors are patterns used to select elements in the DOM.

A

::before / ::after

:active

:any-link

:autofill

Properties

CSS properties style elements through their values.

A

accent-color

align-content

align-items

align-self

CSS Diner game

<https://flukeout.github.io/>

