Lecture_6

Dr.sarah Mohamed

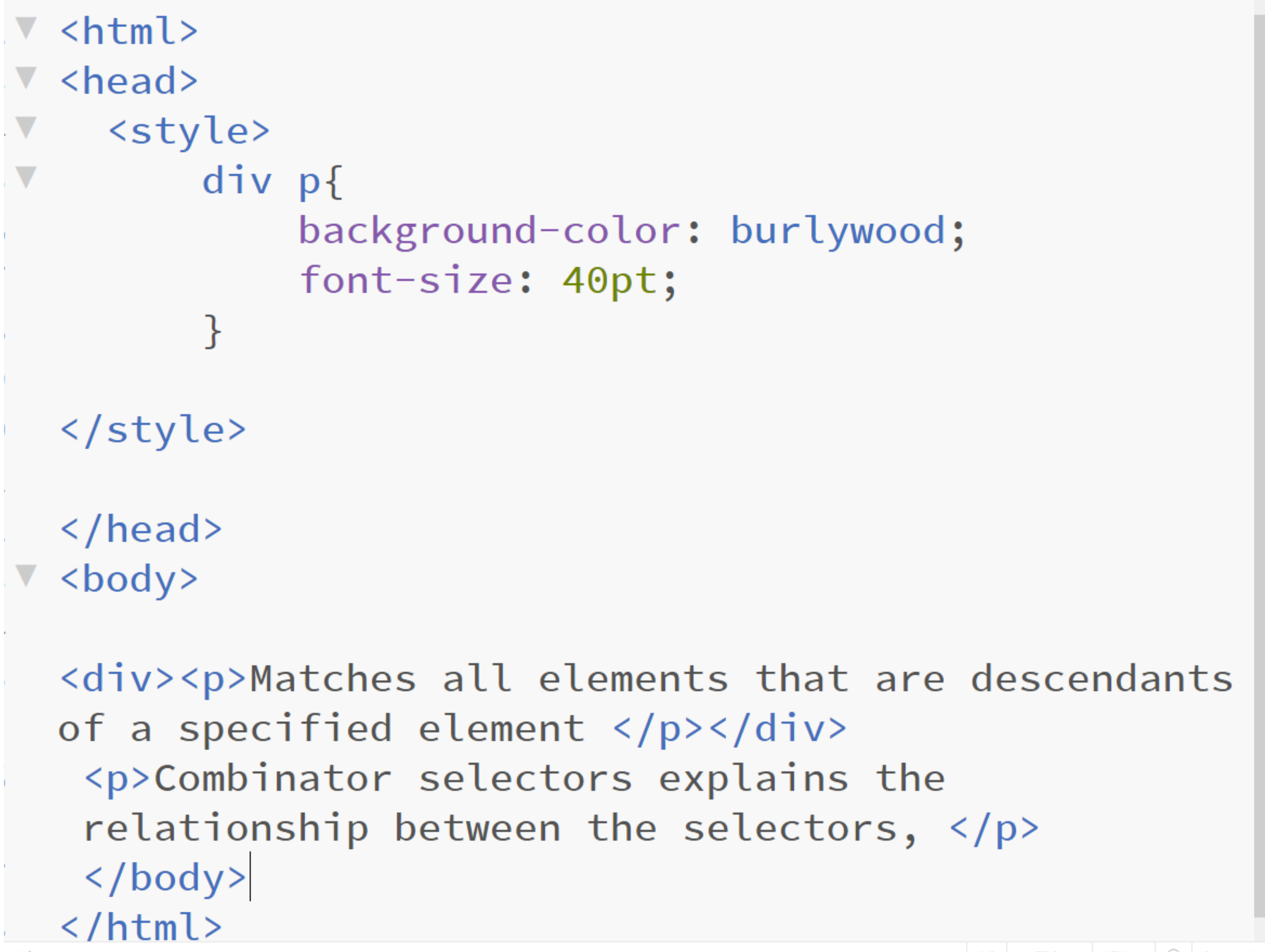
CSS Selectors

Combinator selectors

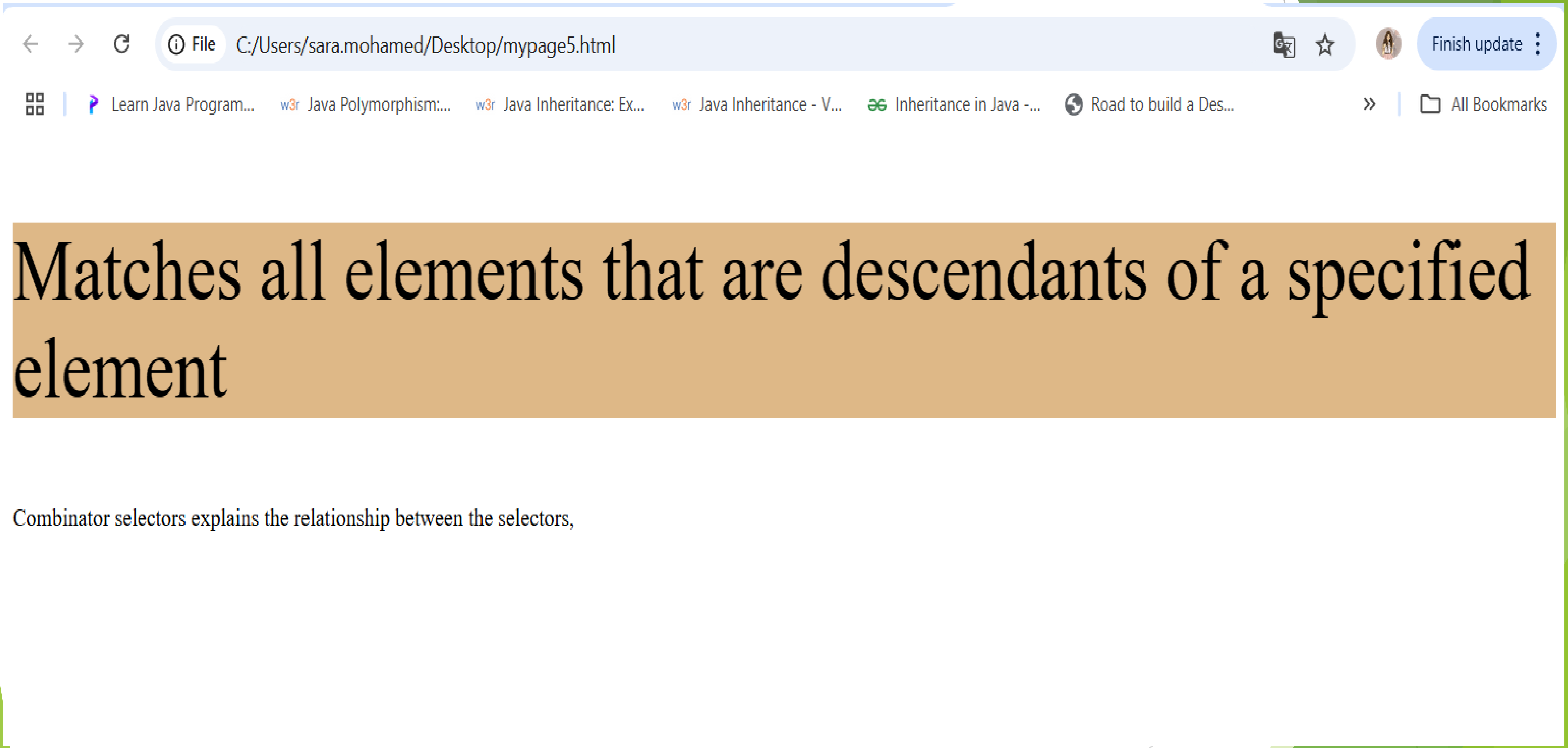
- Combinator selectors explains the relationship between the selectors, it can contain more than one simple selector, it have four types:
  - ❑ descendant selector
  - ❑ child selector (>)
  - ❑ adjacent sibling selector (+)
  - ❑ general sibling selector (~)

descendant selector

- Matches all elements that are descendants of a specified element .
- **For example** : The **div p{}** selector in CSS is called a **descendant selector** — it selects **all** `<p>` **elements that are inside a** `<div>`, no matter how deeply nested they are.



```
<html>
<head>
  <style>
      div p{
          background-color: burlywood;
          font-size: 40pt;
      }

</style>

</head>
<body>

<div><p>Matches all elements that are descendants of a specified element </p></div>
 <p>Combinator selectors explains the relationship between the selectors, </p>
 </body>
</html>
```

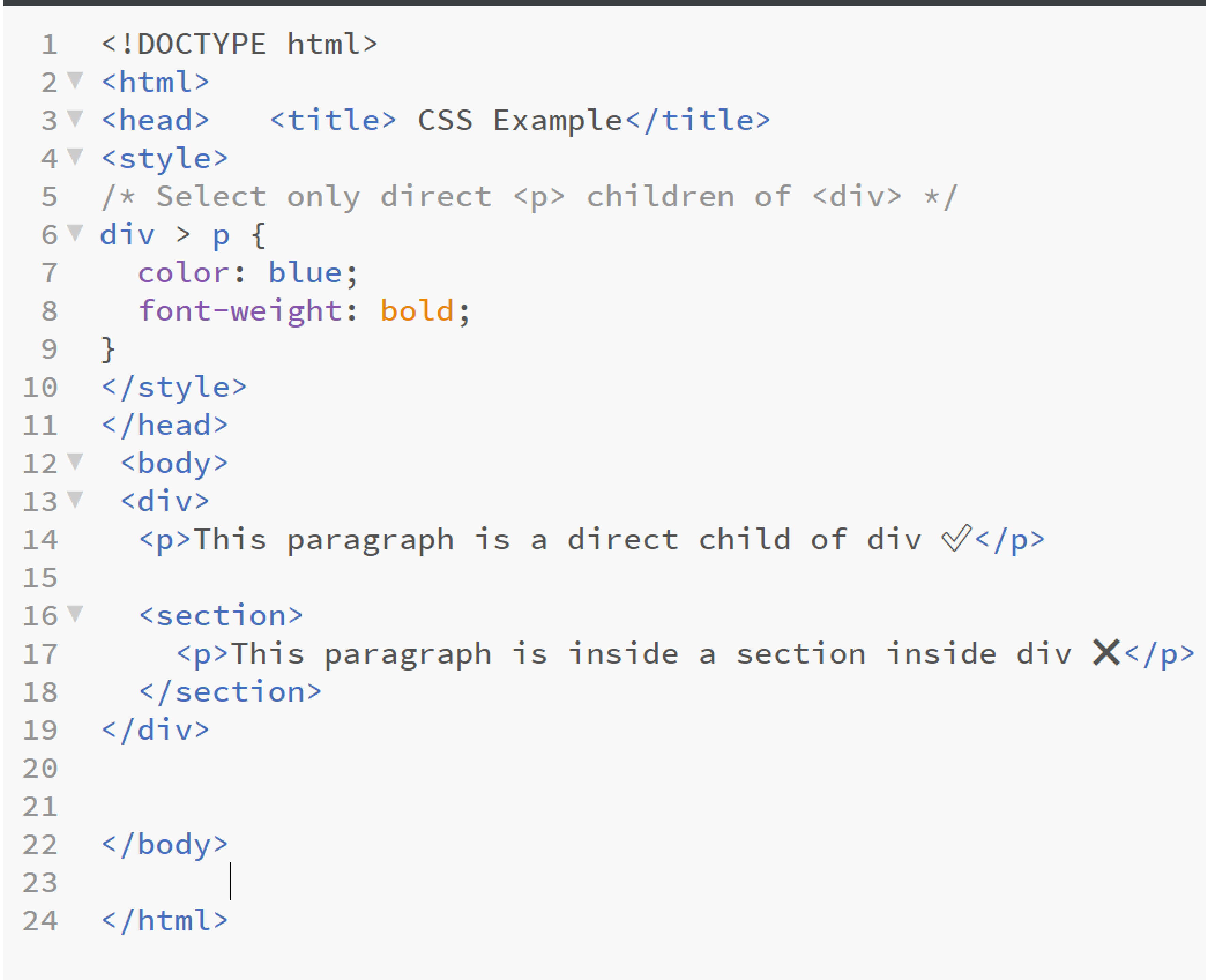


Matches all elements that are descendants of a specified element

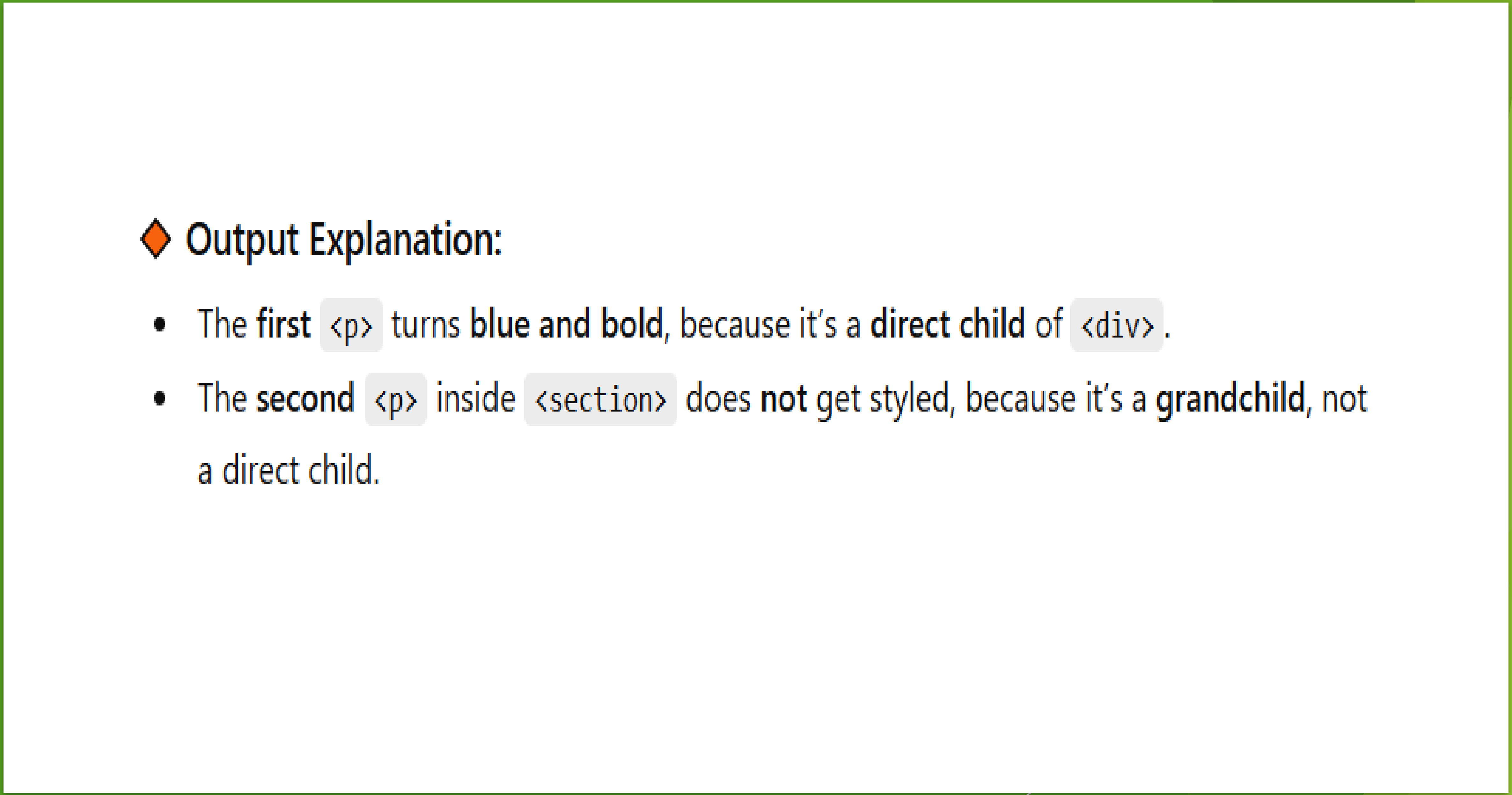
Combinator selectors explains the relationship between the selectors,

child selector (>)

- The **child selector (>)** in CSS is used to select **only the direct children** of an element — not deeper descendants.



```html
<!DOCTYPE html>
<html>
<head>    <title> CSS Example</title>
<style>
/* Select only direct <p> children of <div> */
div > p {
  color: blue;
  font-weight: bold;
}
</style>
</head>
 <body>
 <div>
  <p>This paragraph is a direct child of div ✅</p>

  <section>
    <p>This paragraph is inside a section inside div ❌</p>
  </section>
</div>


</body>

</html>
```

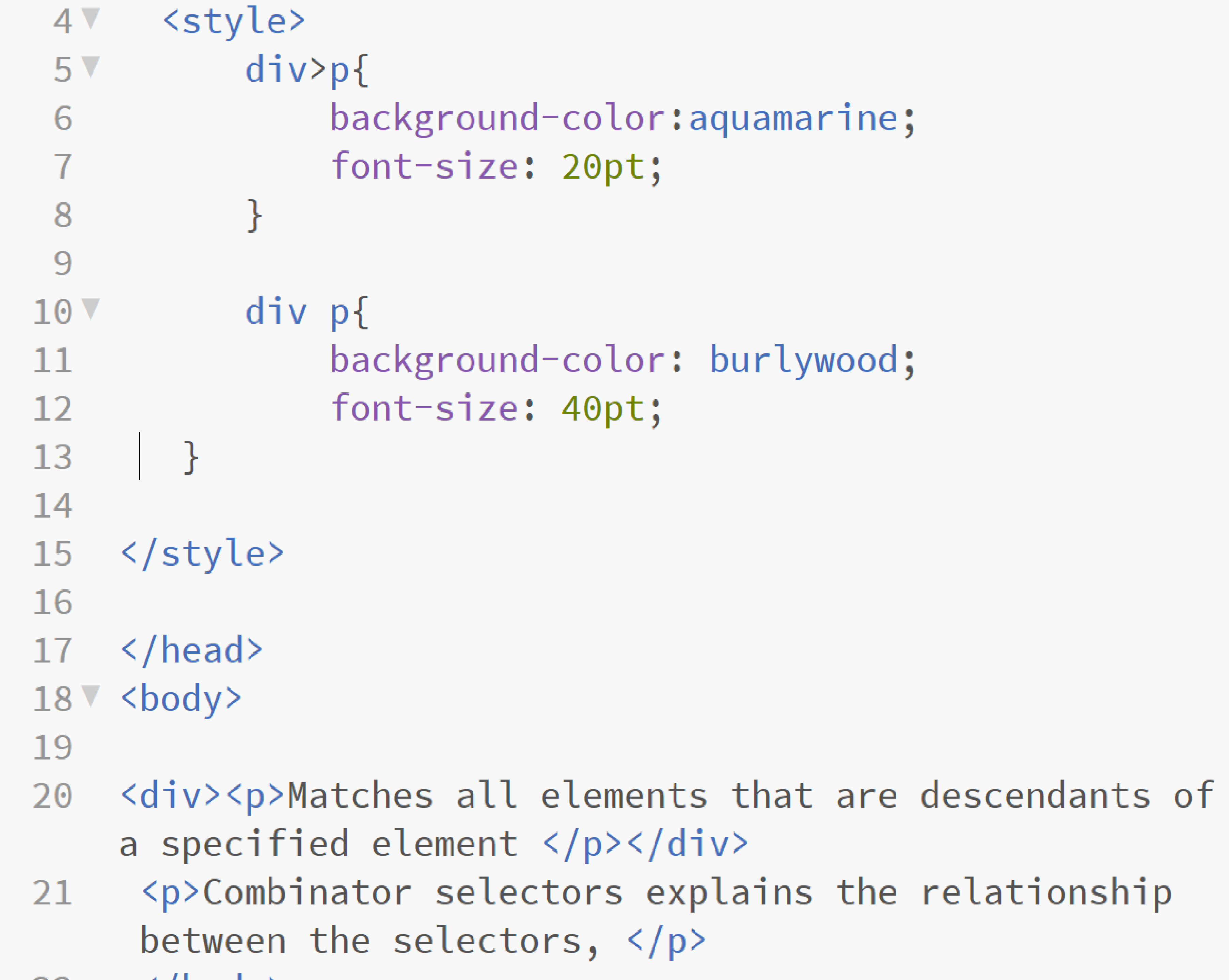


🔶 Output Explanation:

- The **first** `<p>` turns **blue and bold**, because it's a **direct child** of `<div>`.
- The **second** `<p>` inside `<section>` does **not** get styled, because it's a **grandchild**, not a direct child.

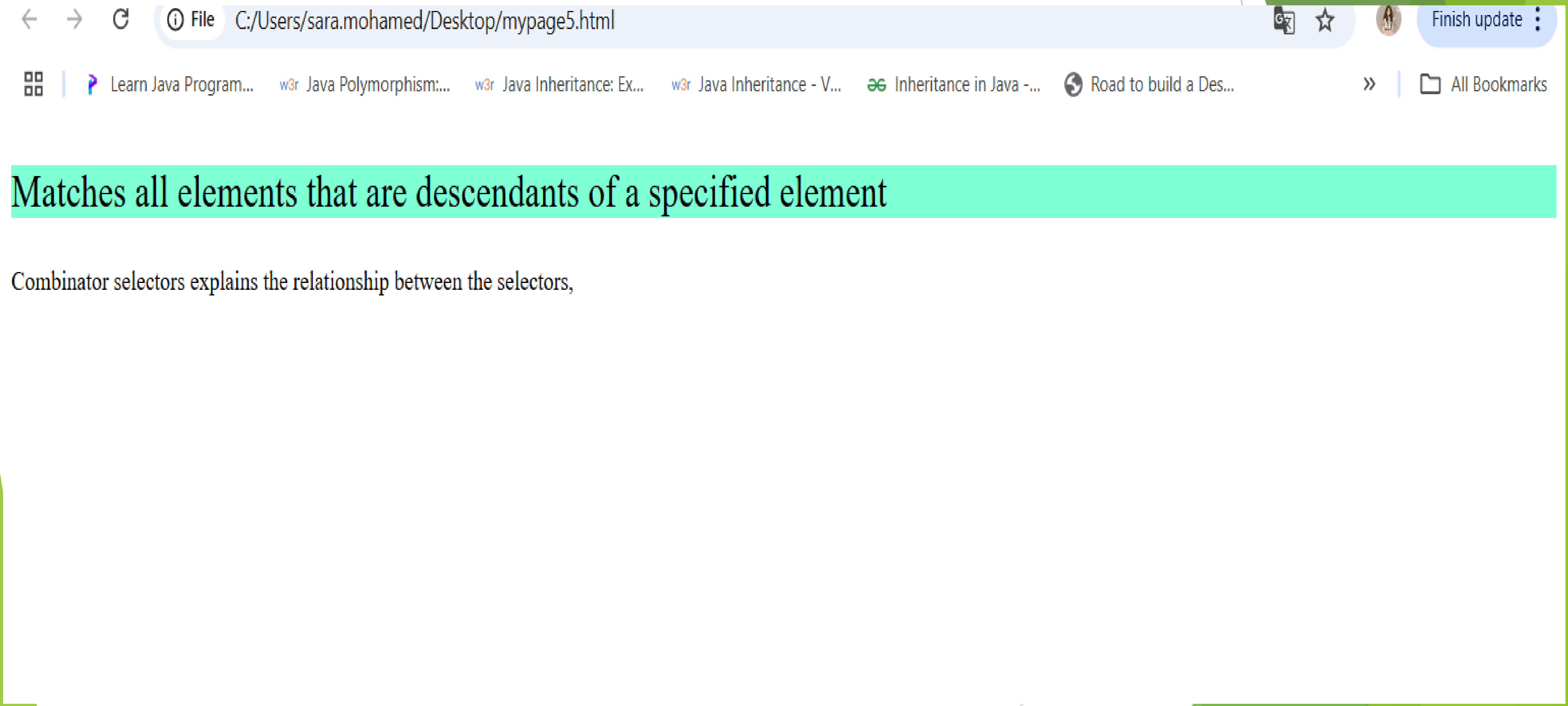
child selector (>)

- Example_1: Selects all elements that are the children of a specified element.
- selects all elements that are children of an element.



```
  <style>
      div>p{
          background-color:aquamarine;
          font-size: 20pt;
      }

      div p{
          background-color: burlywood;
          font-size: 40pt;
   }

</style>

</head>
<body>

<div><p>Matches all elements that are descendants of a specified element </p></div>
 <p>Combinator selectors explains the relationship between the selectors, </p>
```

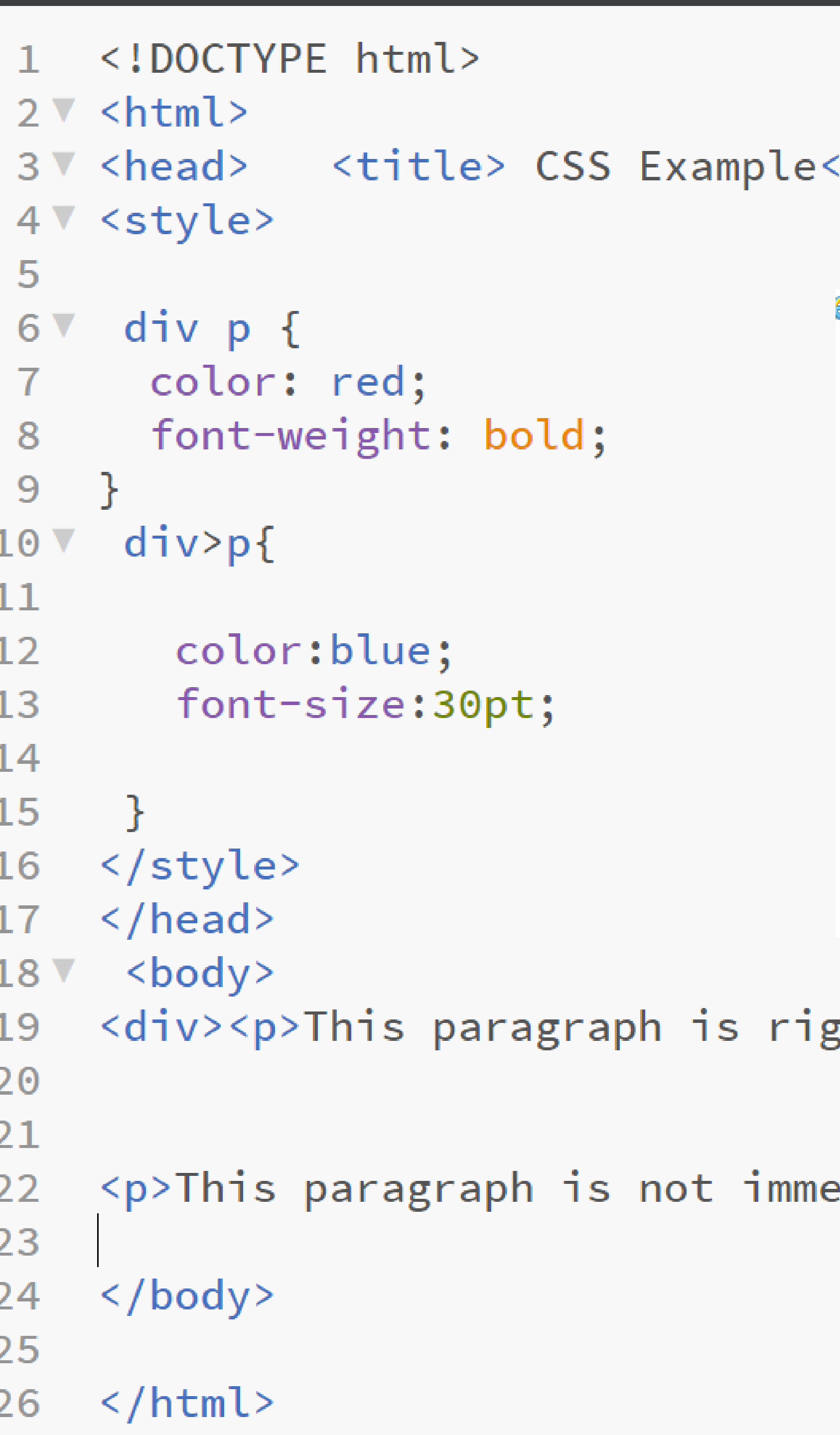
The output of code



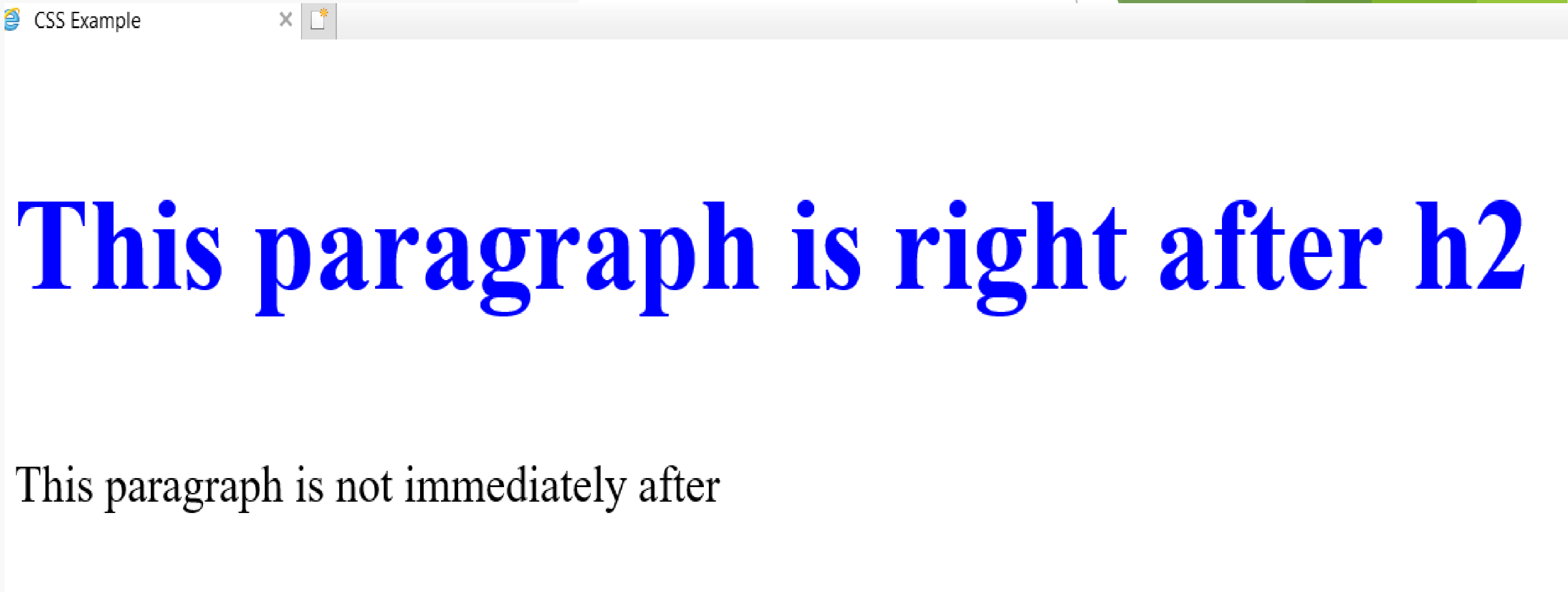
Matches all elements that are descendants of a specified element

Combinator selectors explains the relationship between the selectors,

Example_2



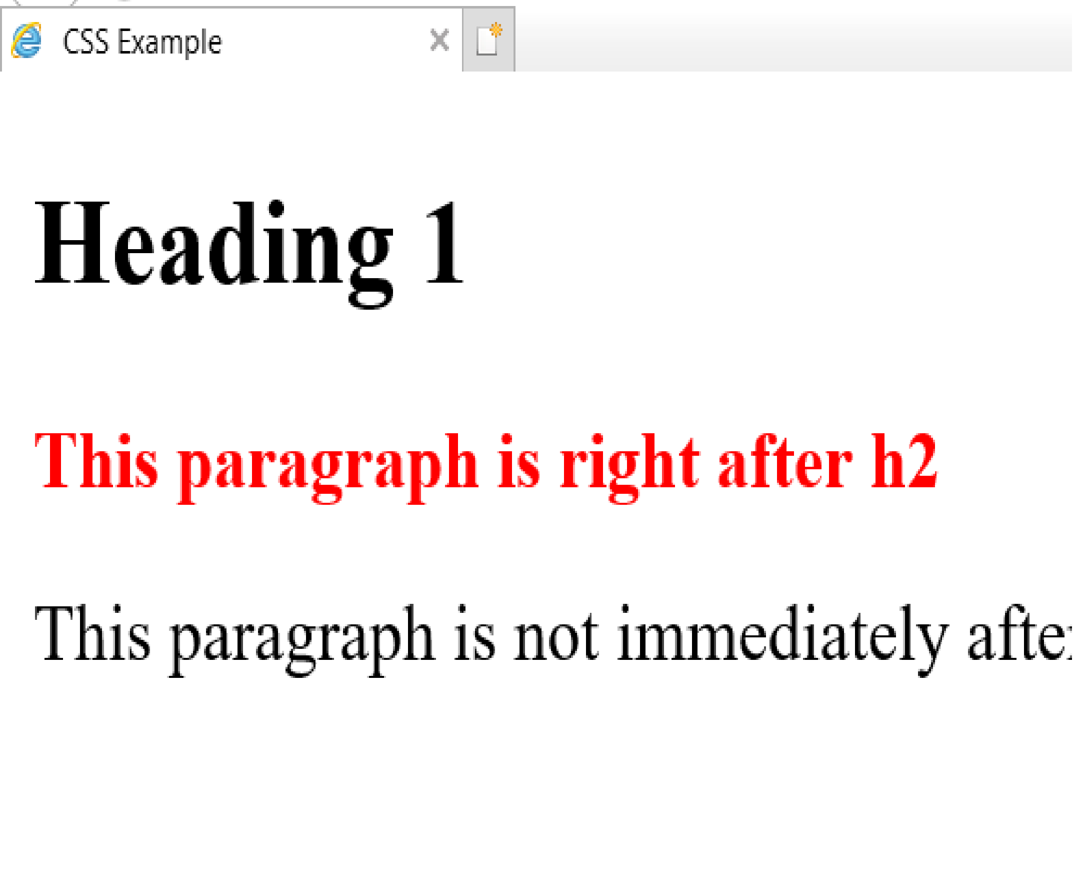
```
<!DOCTYPE html>
<html>
<head>    <title> CSS Example</title>
<style>

 div p {
  color: red;
  font-weight: bold;
}
 div>p{

   color:blue;
   font-size:30pt;

 }
</style>
</head>
 <body>
<div><p>This paragraph is right after h2 </p> </div>


<p>This paragraph is not immediately after </p>

</body>

</html>
```



This paragraph is right after h2

This paragraph is not immediately after

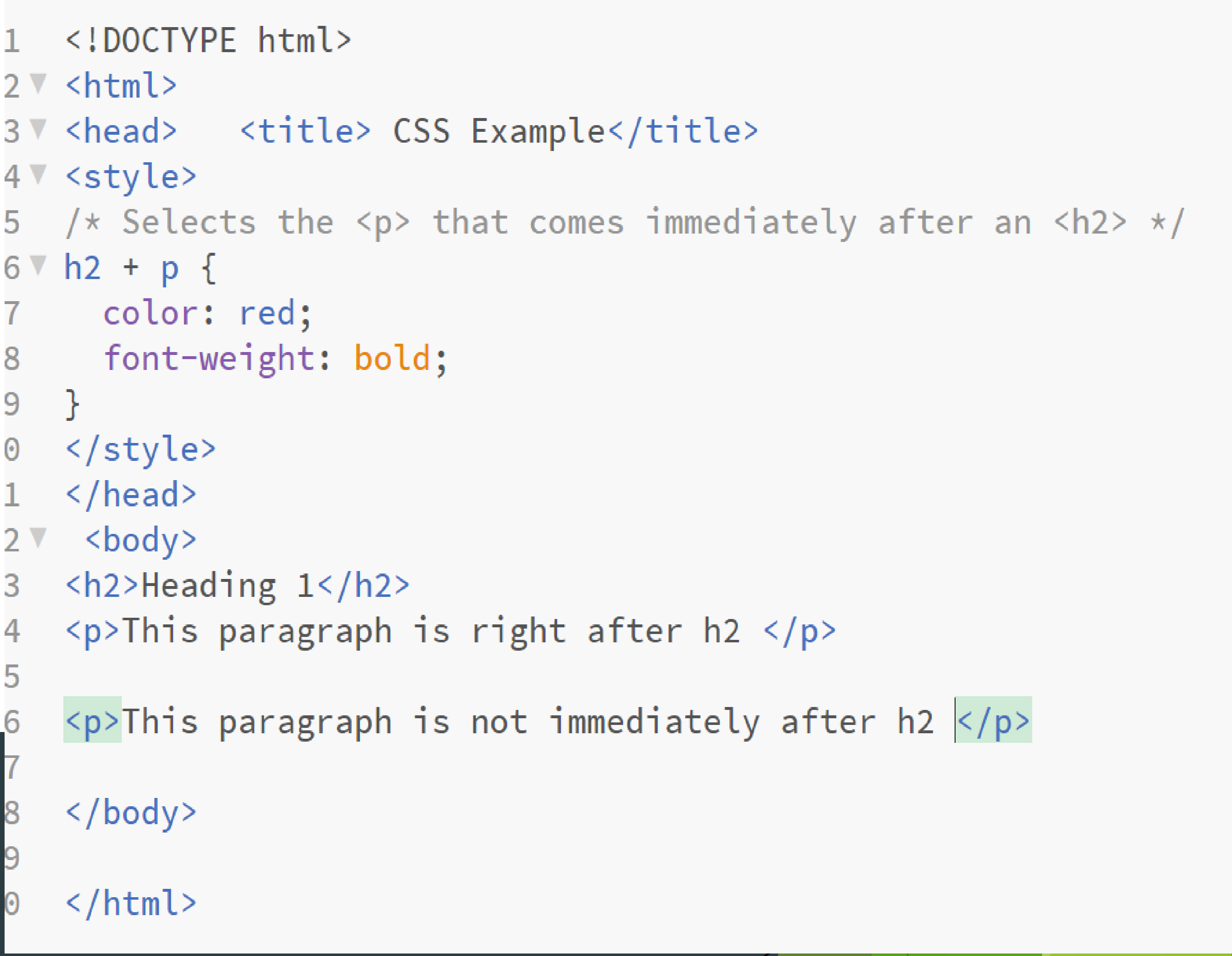
The **adjacent sibling selector (+)** in CSS



Heading 1

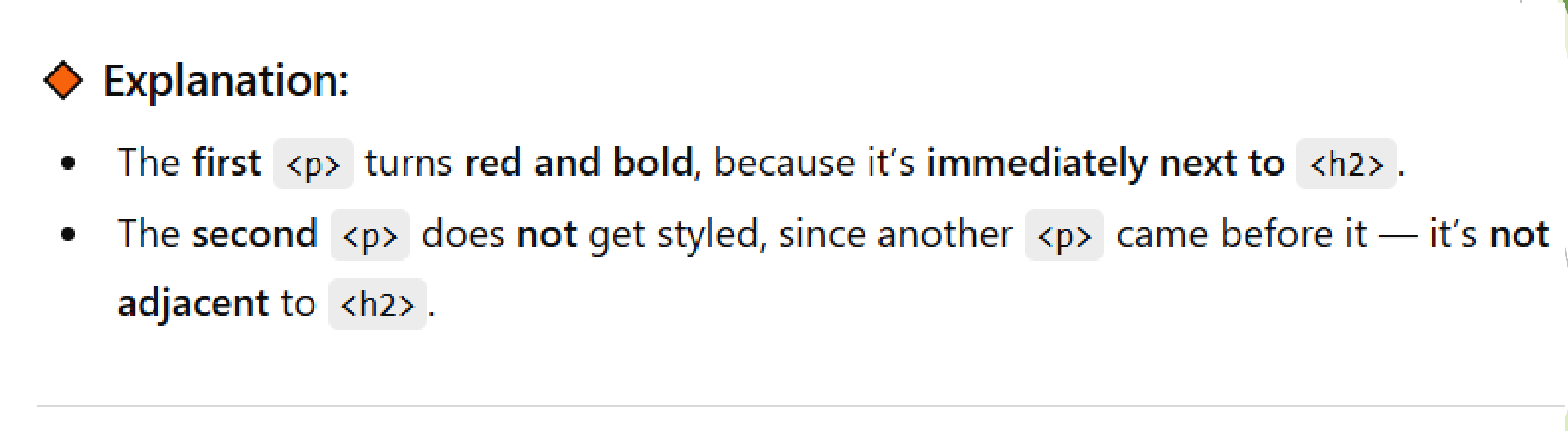
This paragraph is right after h2

This paragraph is not immediately afte



```
<!DOCTYPE html>
<html>
<head>   <title> CSS Example</title>
<style>
/* Selects the <p> that comes immediately after an <h2> */
h2 + p {
  color: red;
  font-weight: bold;
}
</style>
</head>
 <body>
<h2>Heading 1</h2>
<p>This paragraph is right after h2 </p>

<p>This paragraph is not immediately after h2 </p>

</body>

</html>
```

The adjacent sibling selector (+) in CSS selects the element that is immediately next (directly after) a specified element

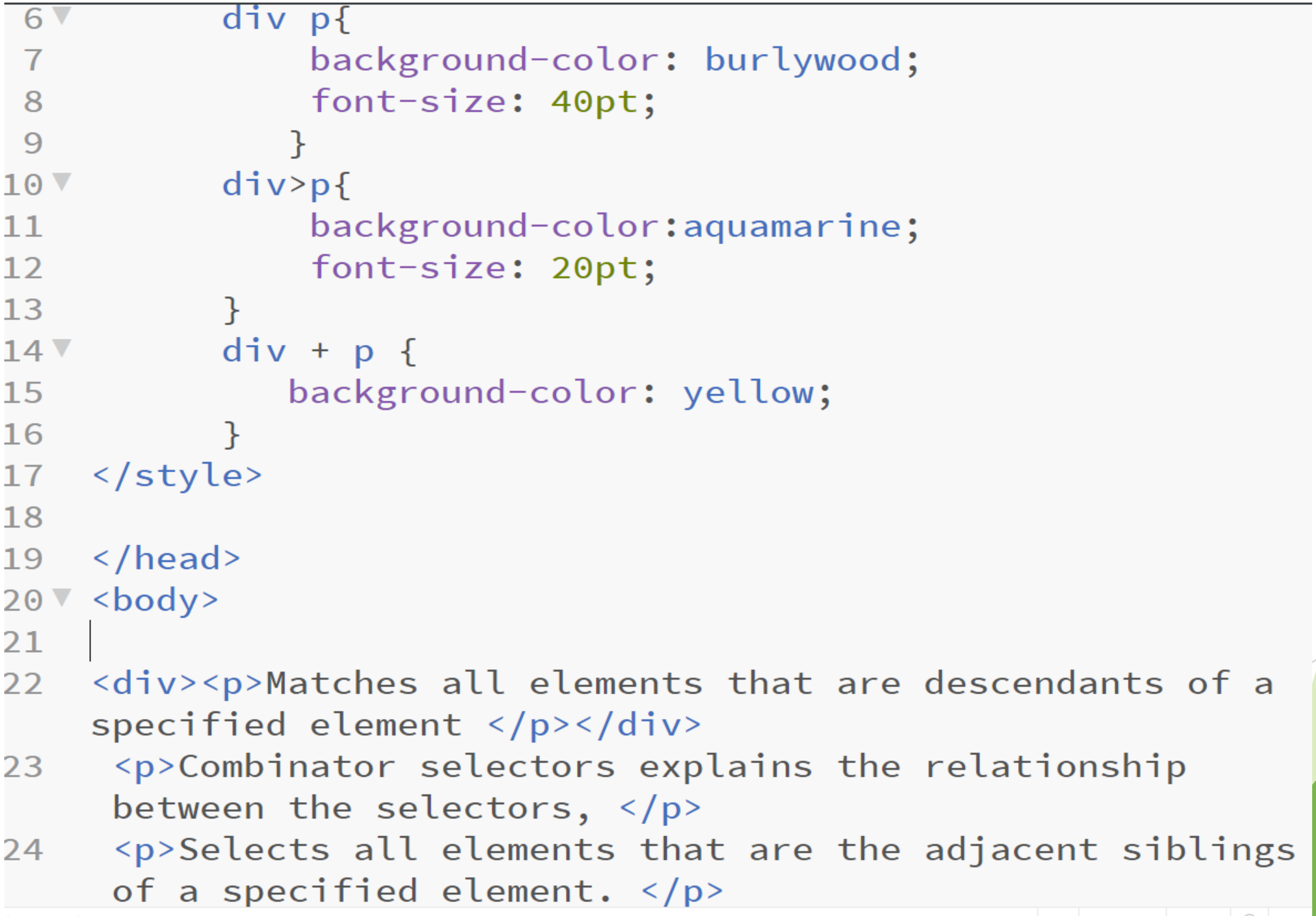


🔶 Explanation:

- The **first** `<p>` turns **red and bold**, because it's **immediately next to** `<h2>`.
- The **second** `<p>` does **not** get styled, since another `<p>` came before it — it's **not adjacent** to `<h2>`.

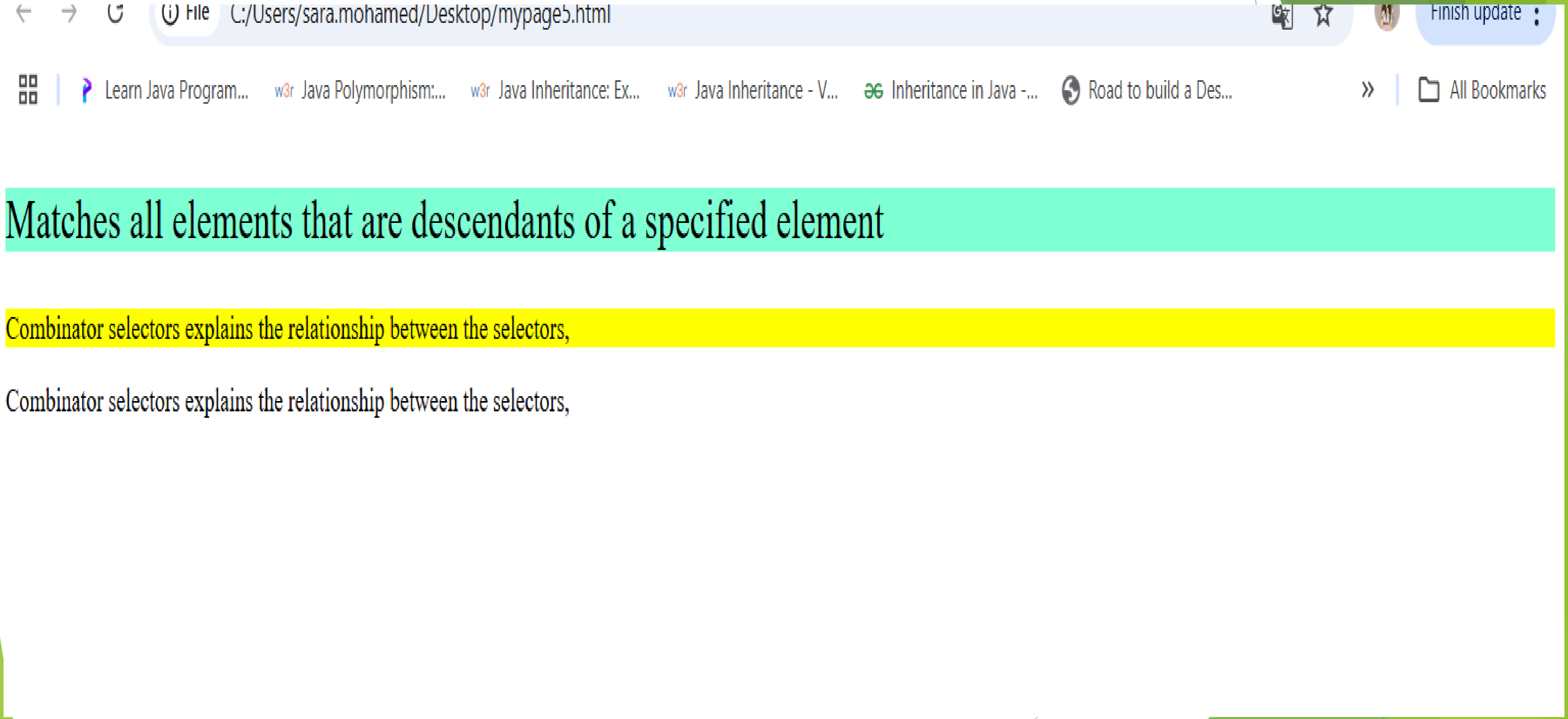
adjacent sibling selector (+)

- Selects all elements that are the adjacent siblings of a specified element.
- Sibling elements must have the same parent element and immediately following its parent.



```
      div p{
          background-color: burlywood;
          font-size: 40pt;
         }
      div>p{
          background-color:aquamarine;
          font-size: 20pt;
      }
      div + p {
         background-color: yellow;
      }
</style>

</head>
<body>

<div><p>Matches all elements that are descendants of a specified element </p></div>
 <p>Combinator selectors explains the relationship between the selectors, </p>
 <p>Selects all elements that are the adjacent siblings of a specified element. </p>
```

Output



Matches all elements that are descendants of a specified element

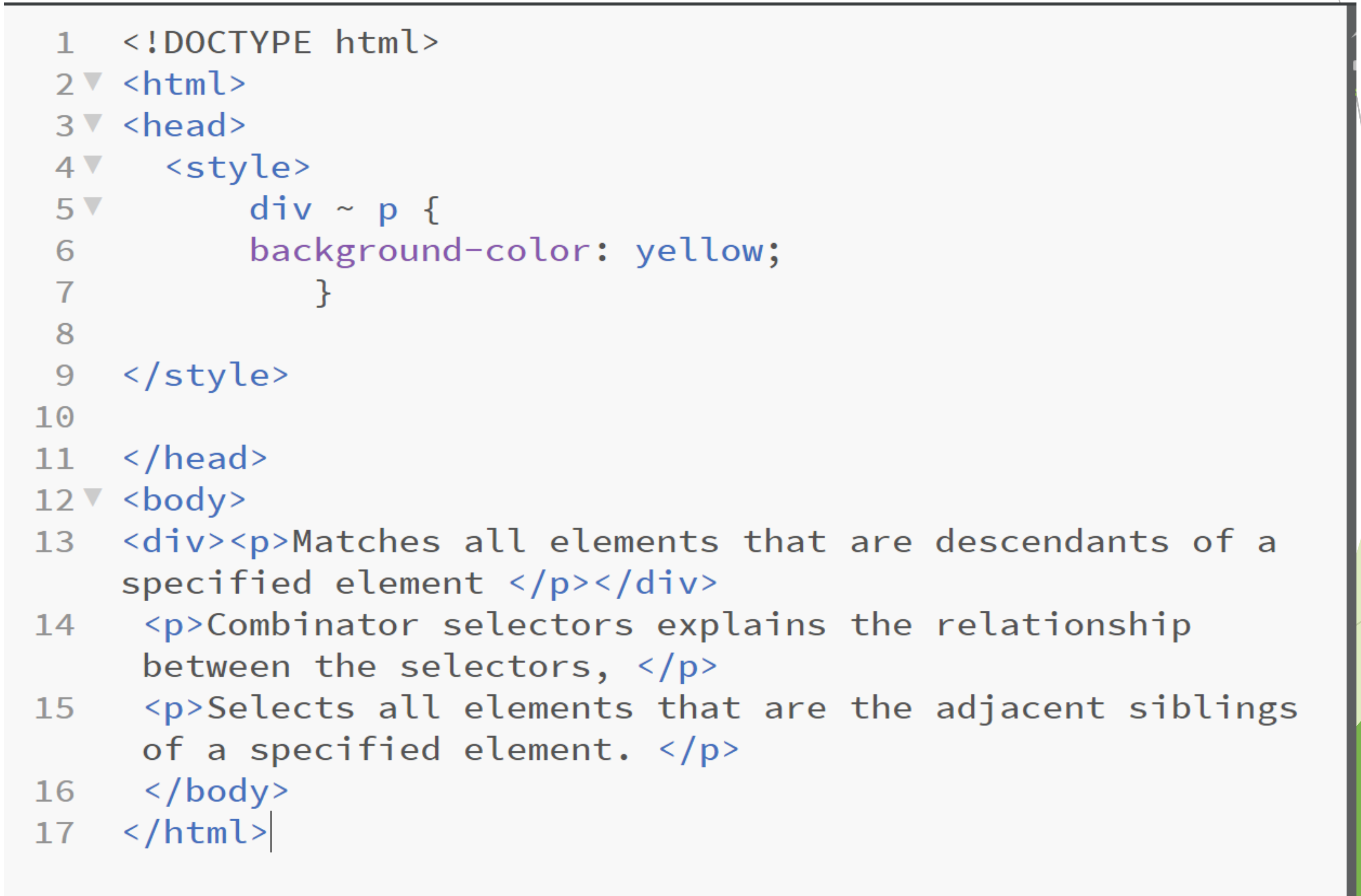
Combinator selectors explains the relationship between the selectors,

Combinator selectors explains the relationship between the selectors,

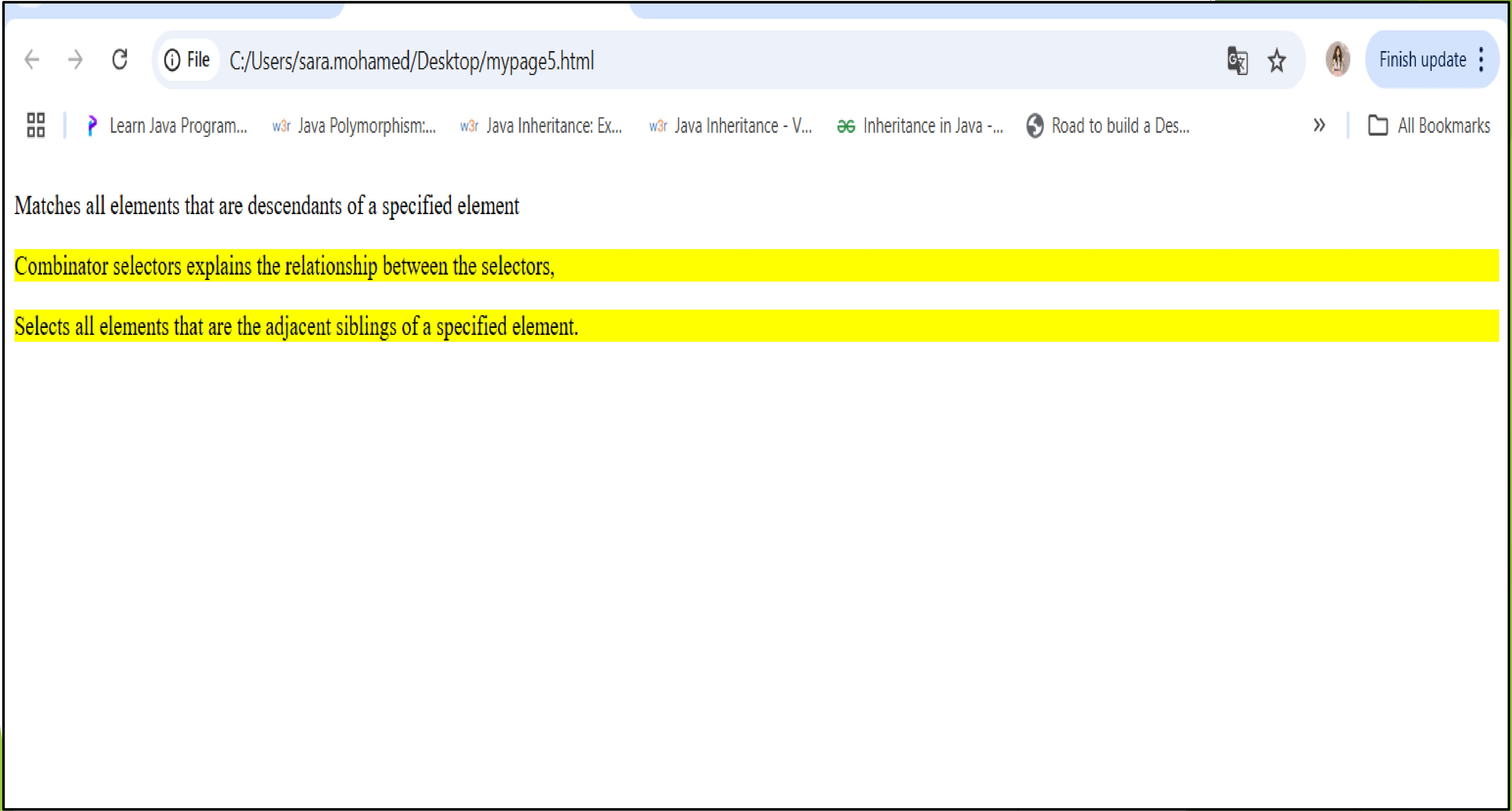
The **general sibling selector (~)** in CSS selects **all elements that share the same parent** and come **after** a specified element.

Example for the general sibling selector (~)

- The general sibling selector selects all elements that are siblings of a specified element



```
<!DOCTYPE html>
<html>
<head>
  <style>
      div ~ p {
      background-color: yellow;
         }

</style>

</head>
<body>
<div><p>Matches all elements that are descendants of a specified element </p></div>
 <p>Combinator selectors explains the relationship between the selectors, </p>
 <p>Selects all elements that are the adjacent siblings of a specified element. </p>
 </body>
</html>
```

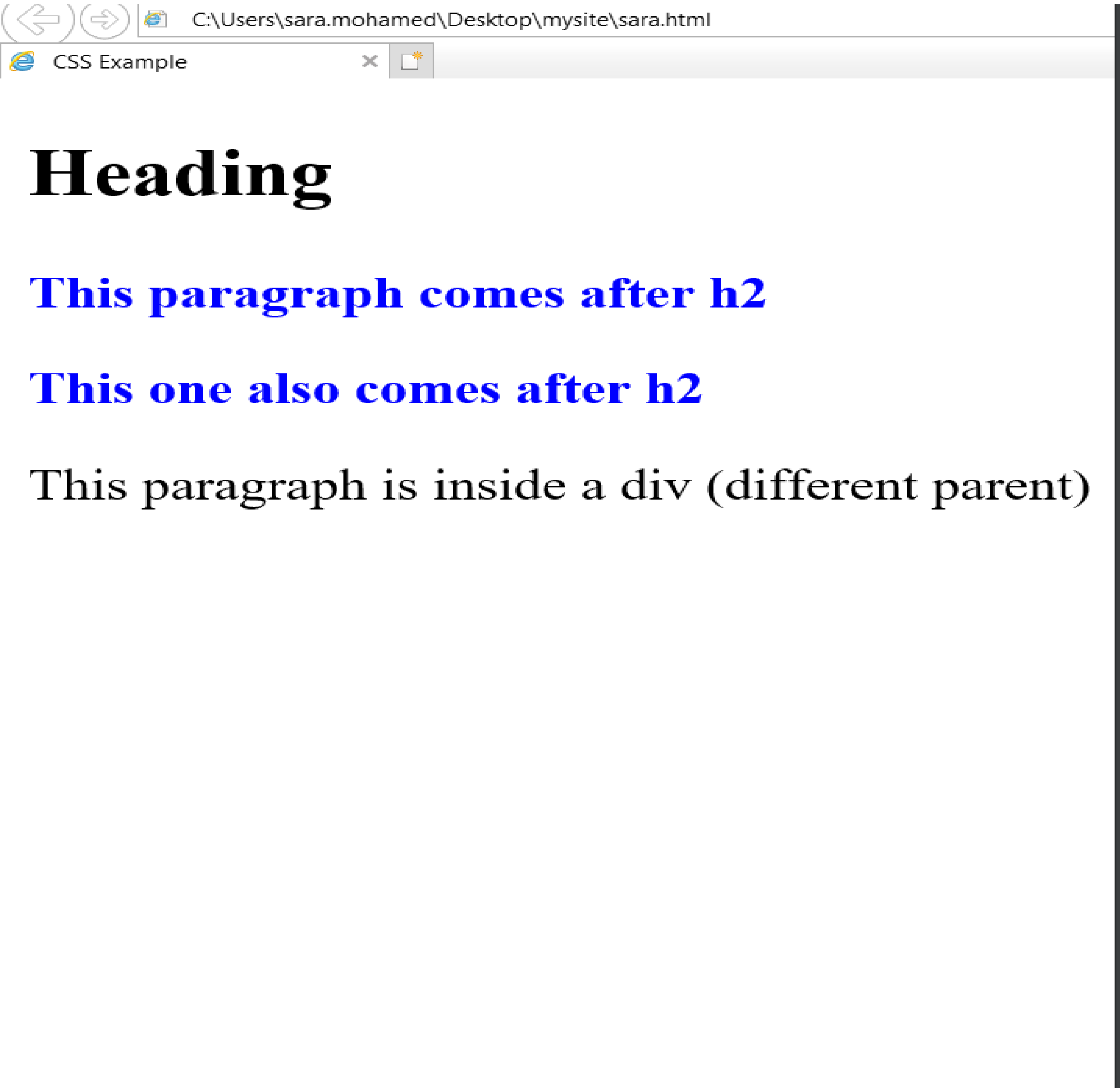


Matches all elements that are descendants of a specified element

Combinator selectors explains the relationship between the selectors,

Selects all elements that are the adjacent siblings of a specified element.

Example_2: /* Selects all <p> elements that come after an <h2> with the same parent */

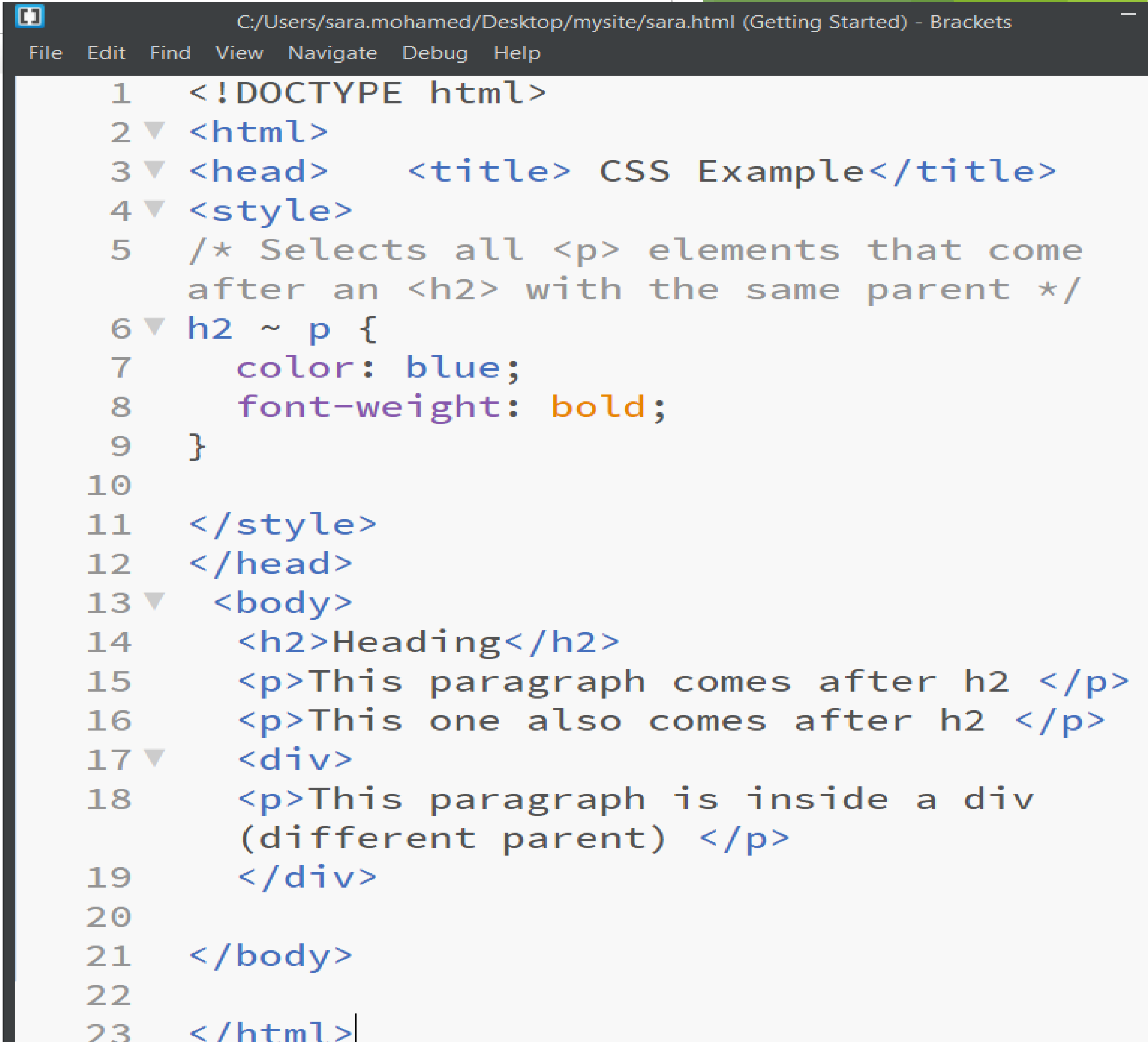


Heading

This paragraph comes after h2

This one also comes after h2

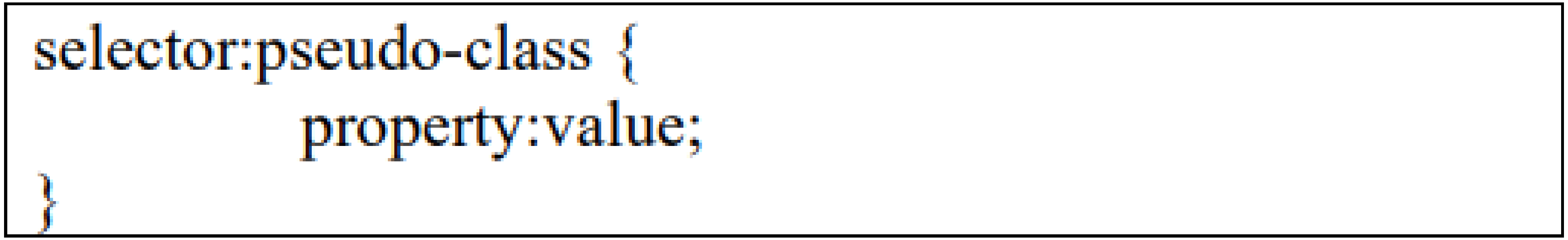
This paragraph is inside a div (different parent)


C:/Users/sara.mohamed/Desktop/mysite/sara.html (Getting Started) - Brackets
File
Edit
Find
View
Navigate
Debug
Help

```
<!DOCTYPE html>
<html>
<head>    <title> CSS Example</title>
<style>
/* Selects all <p> elements that come
after an <h2> with the same parent */
h2 ~ p {
  color: blue;
  font-weight: bold;
}

</style>
</head>
 <body>
  <h2>Heading</h2>
  <p>This paragraph comes after h2 </p>
  <p>This one also comes after h2 </p>
  <div>
  <p>This paragraph is inside a div
  (different parent) </p>
  </div>

</body>

</html>
```

Pseudo-class selectors

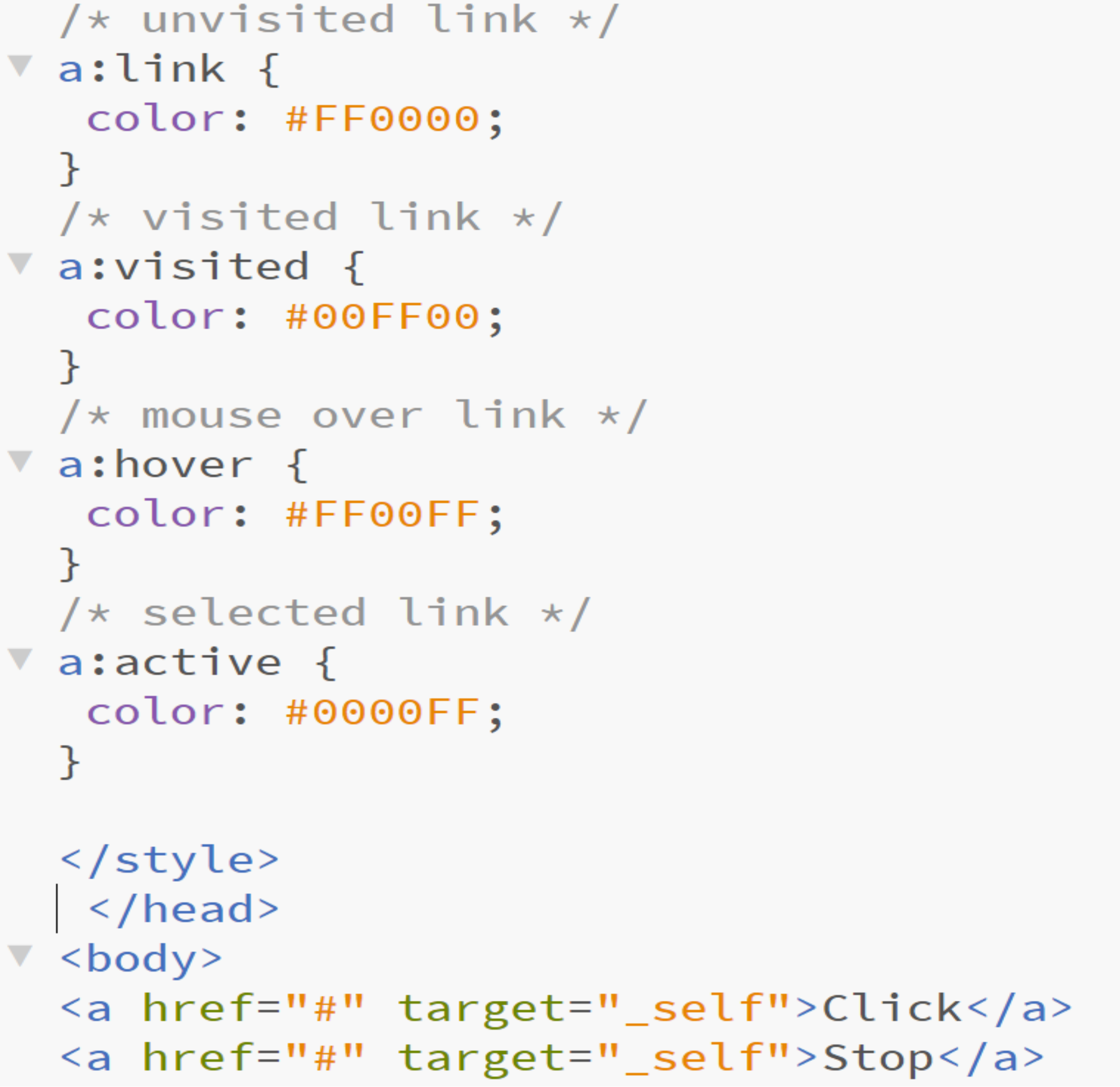
The syntax for pseudo-classes selector



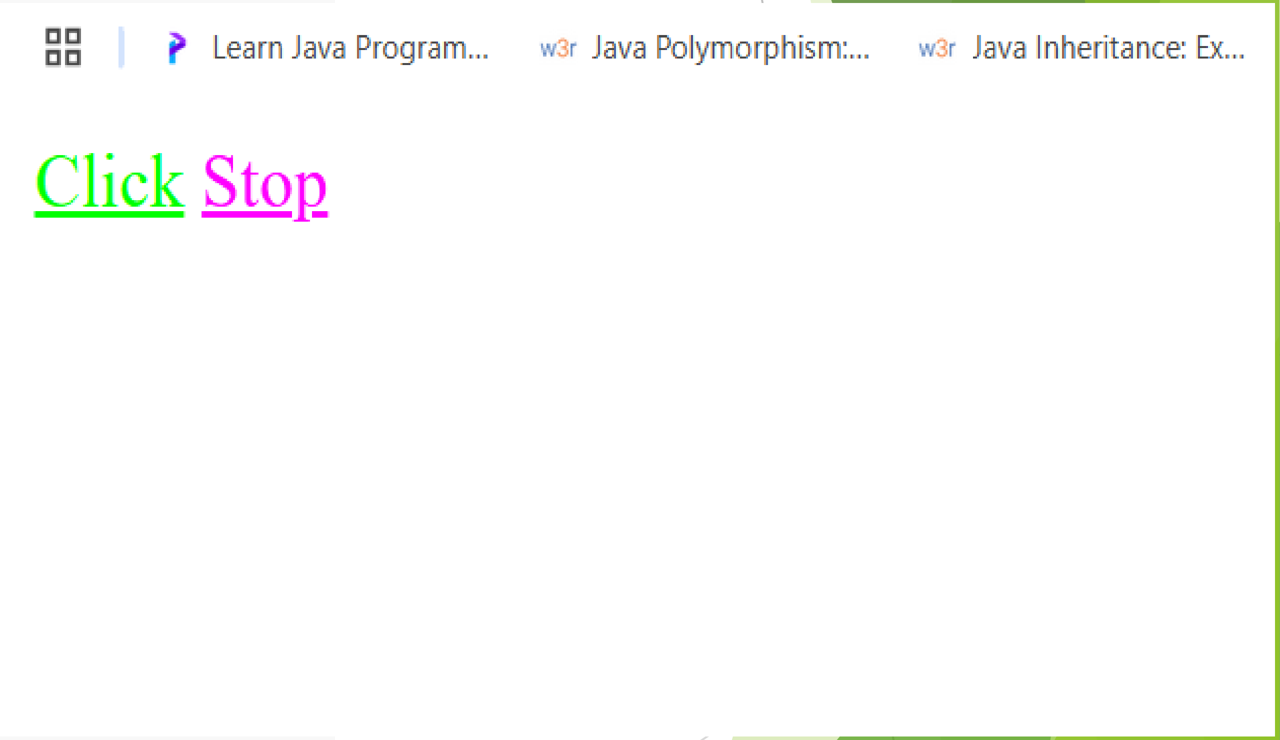
```
selector:pseudo-class {
            property:value;
}
```

- Is used to define a special state of an element
- You can combine more than one pseudo-classes with CSS classes.
- Note: a:hover MUST come after a:link and a:visited in the CSS definition to be effective.
- Note: a:active MUST come after a:hover in the CSS definition in order to be effective.

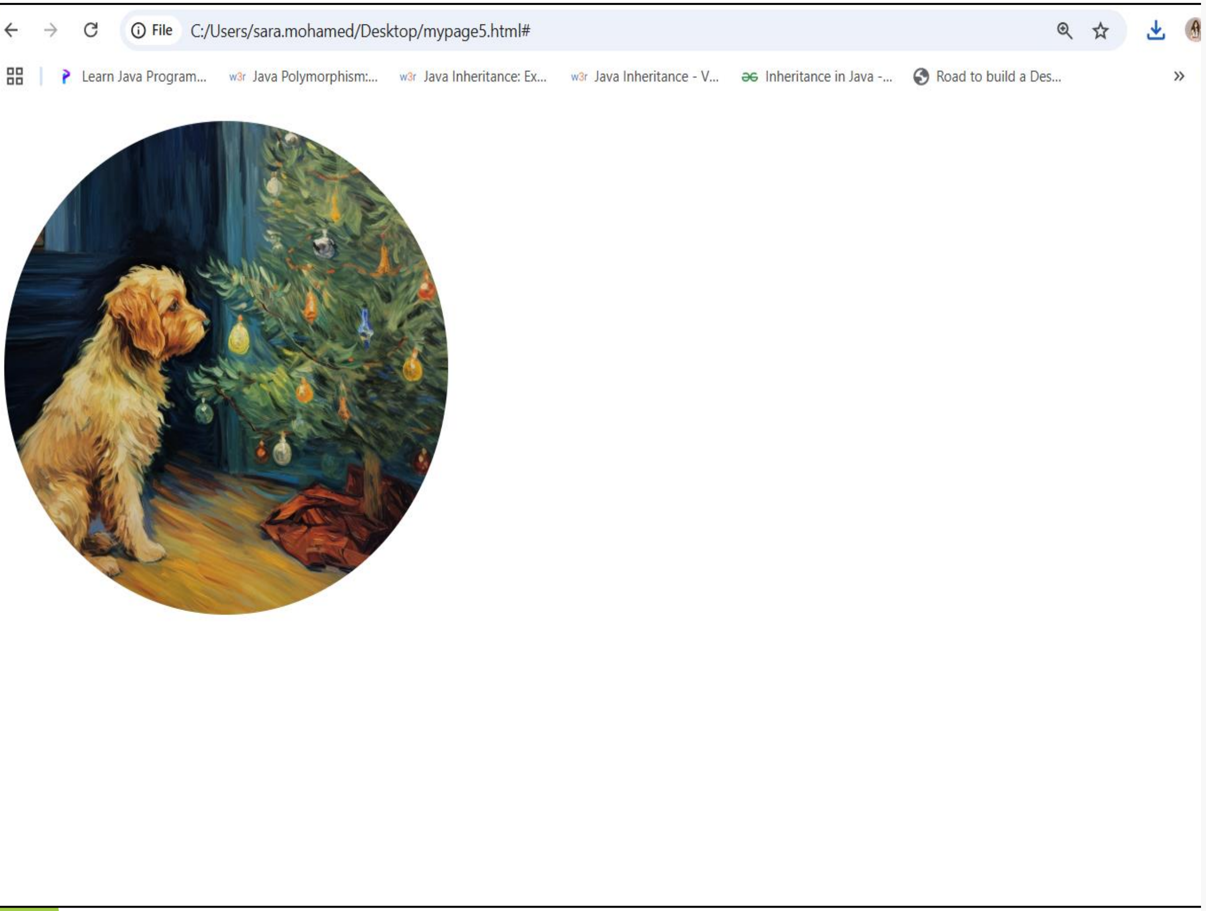
For example, you can use pseudo-classes selector to customize hyperlinks different states appearance

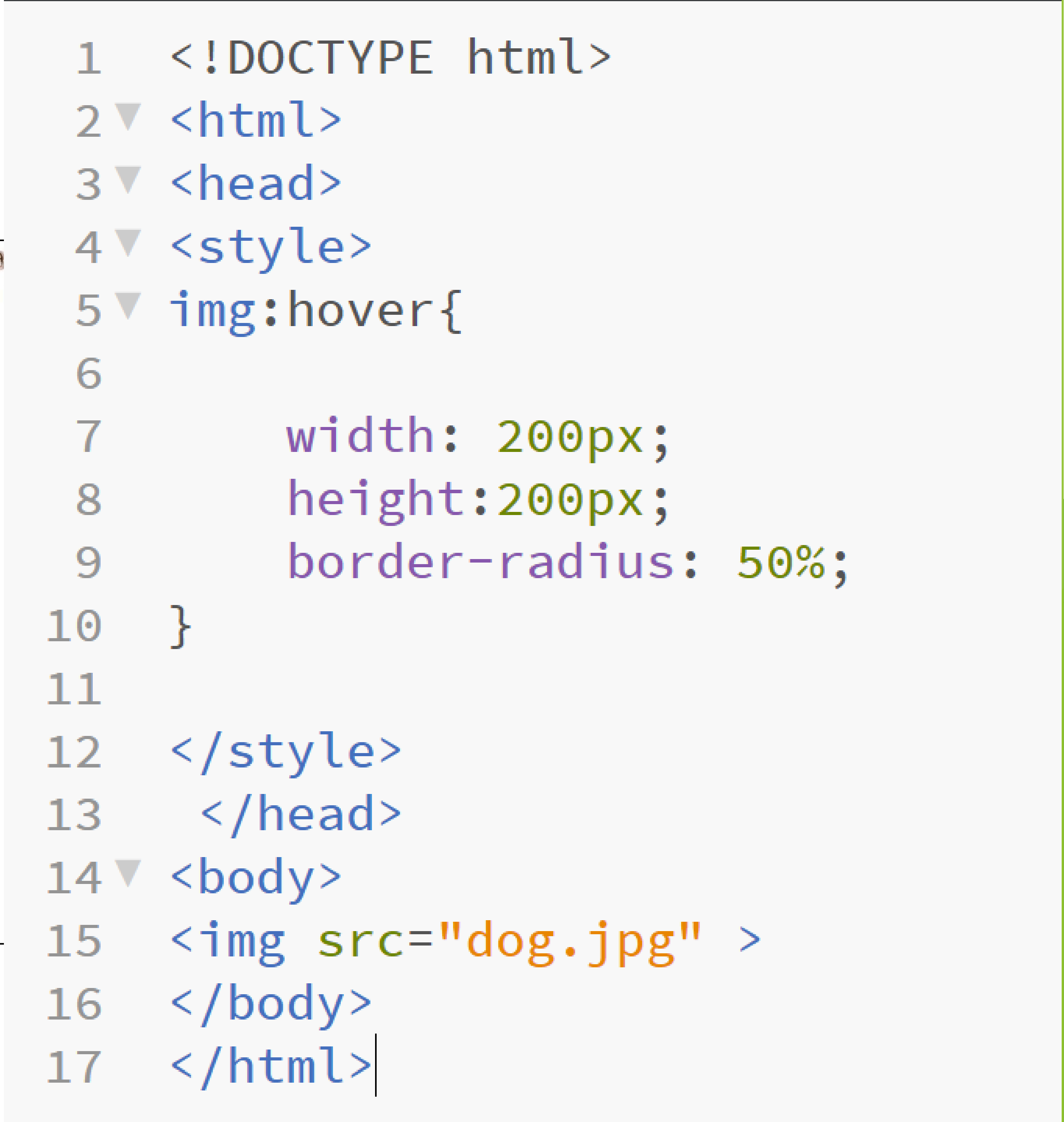


```
/* unvisited link */
a:link {
 color: #FF0000;
}
/* visited link */
a:visited {
 color: #00FF00;
}
/* mouse over link */
a:hover {
 color: #FF00FF;
}
/* selected link */
a:active {
 color: #0000FF;
}

</style>
 </head>
<body>
<a href="#" target="_self">Click</a>
<a href="#" target="_self">Stop</a>
```



img:hover example

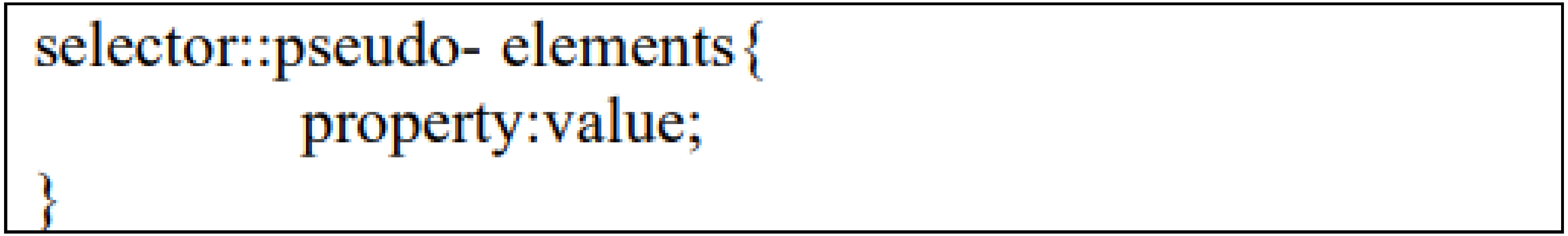




```
<!DOCTYPE html>
<html>
<head>
<style>
img:hover{

        width: 200px;
        height:200px;
        border-radius: 50%;
}

</style>
 </head>
<body>
<img src="dog.jpg" >
</body>
</html>
```

Pseudo-elements selectors

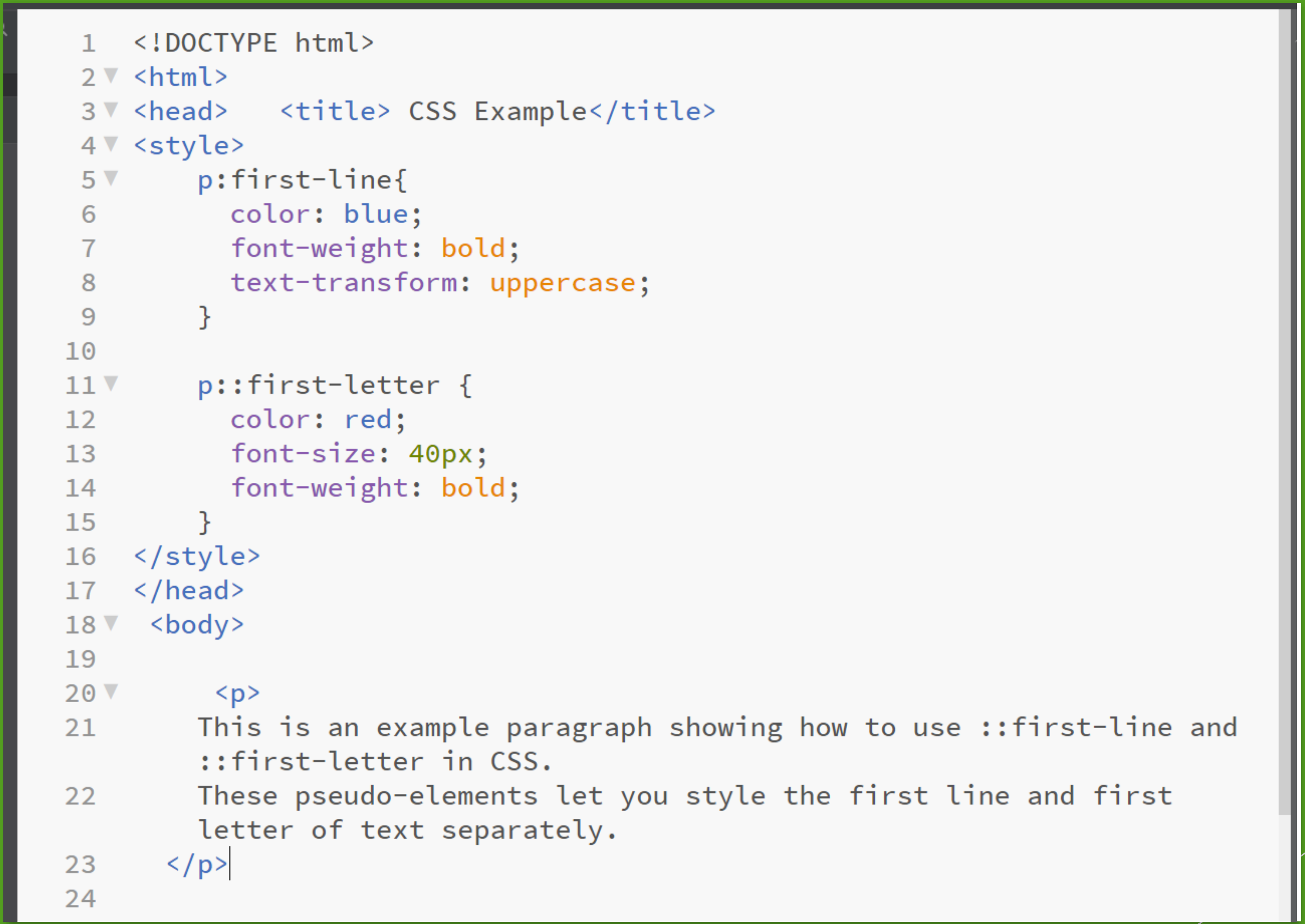
- The syntax for pseudo- elements selector:



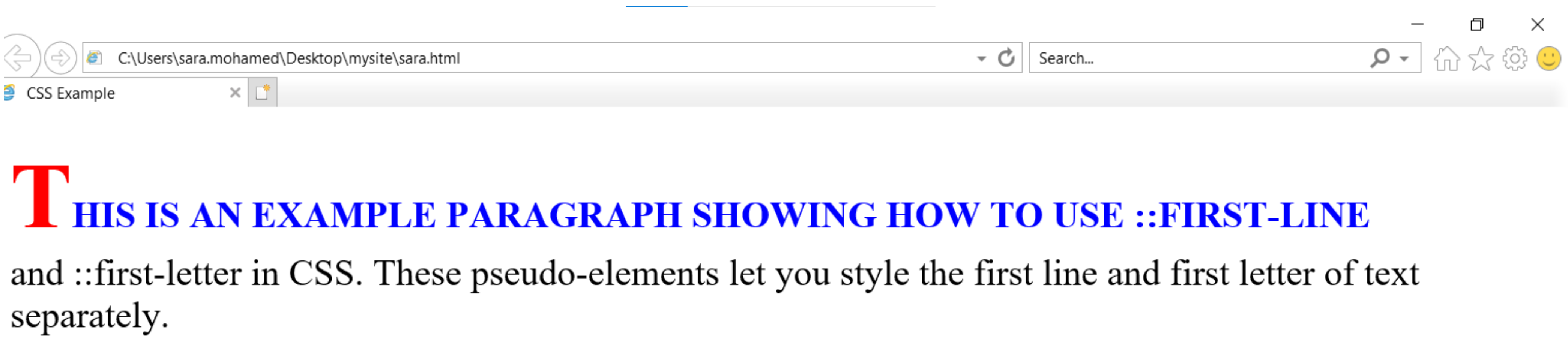
```
selector::pseudo- elements{
            property:value;
}
```

- Used to style specified parts of an element.
- For example, it can be used to the first letter, or line, of an element or Insert content before, or after, the content of an element

- First-line Pseudo-element used to add a special style to the first line of a text.
- First-letter Pseudo-element used to add a special style to the first letter of a text.

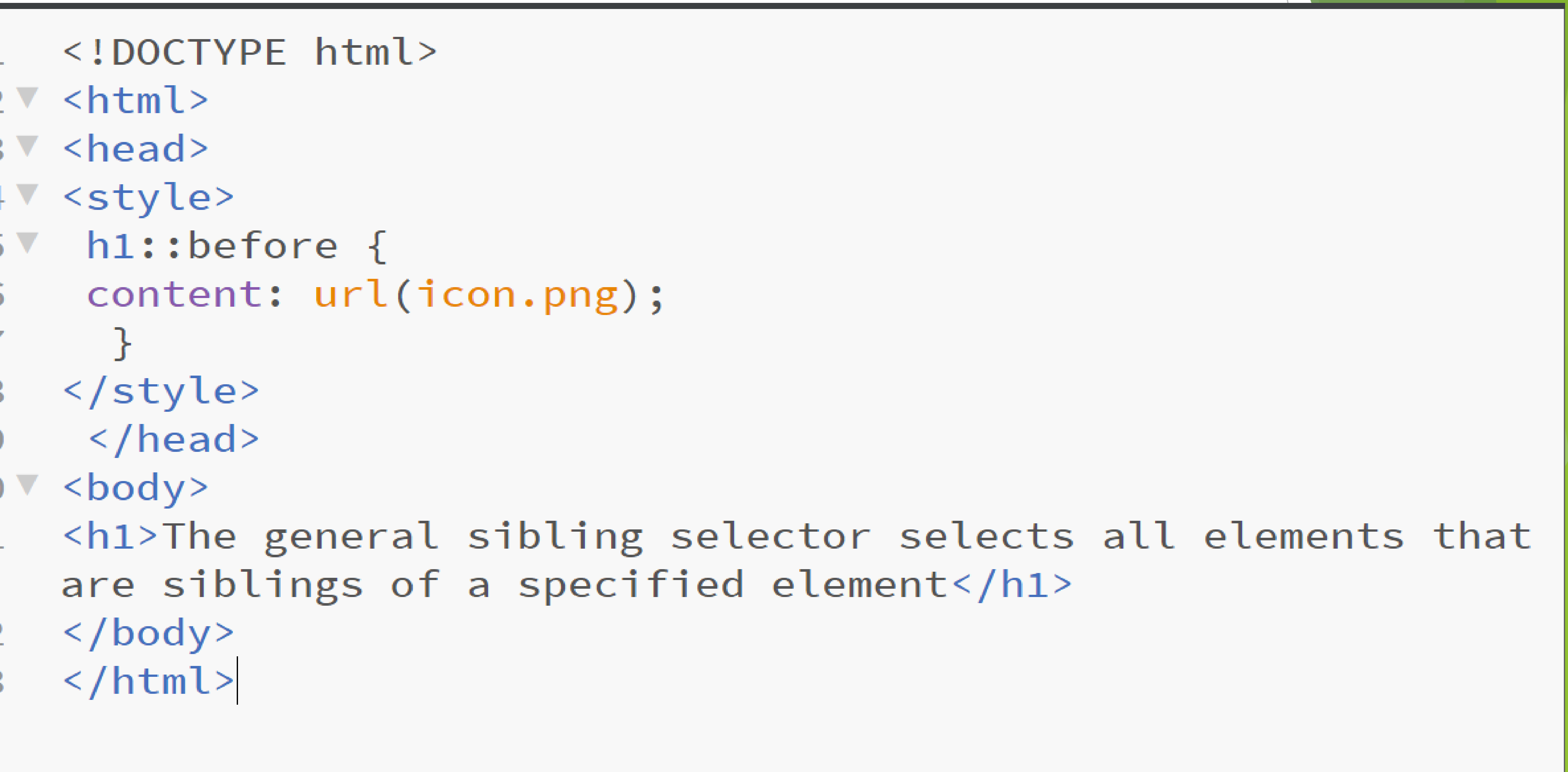


```
<!DOCTYPE html>
<html>
<head>   <title> CSS Example</title>
<style>
    p:first-line{
      color: blue;
      font-weight: bold;
      text-transform: uppercase;
    }

    p::first-letter {
      color: red;
      font-size: 40px;
      font-weight: bold;
    }
</style>
</head>
 <body>

     <p>
    This is an example paragraph showing how to use ::first-line and ::first-letter in CSS.
    These pseudo-elements let you style the first line and first letter of text separately.
  </p>

```



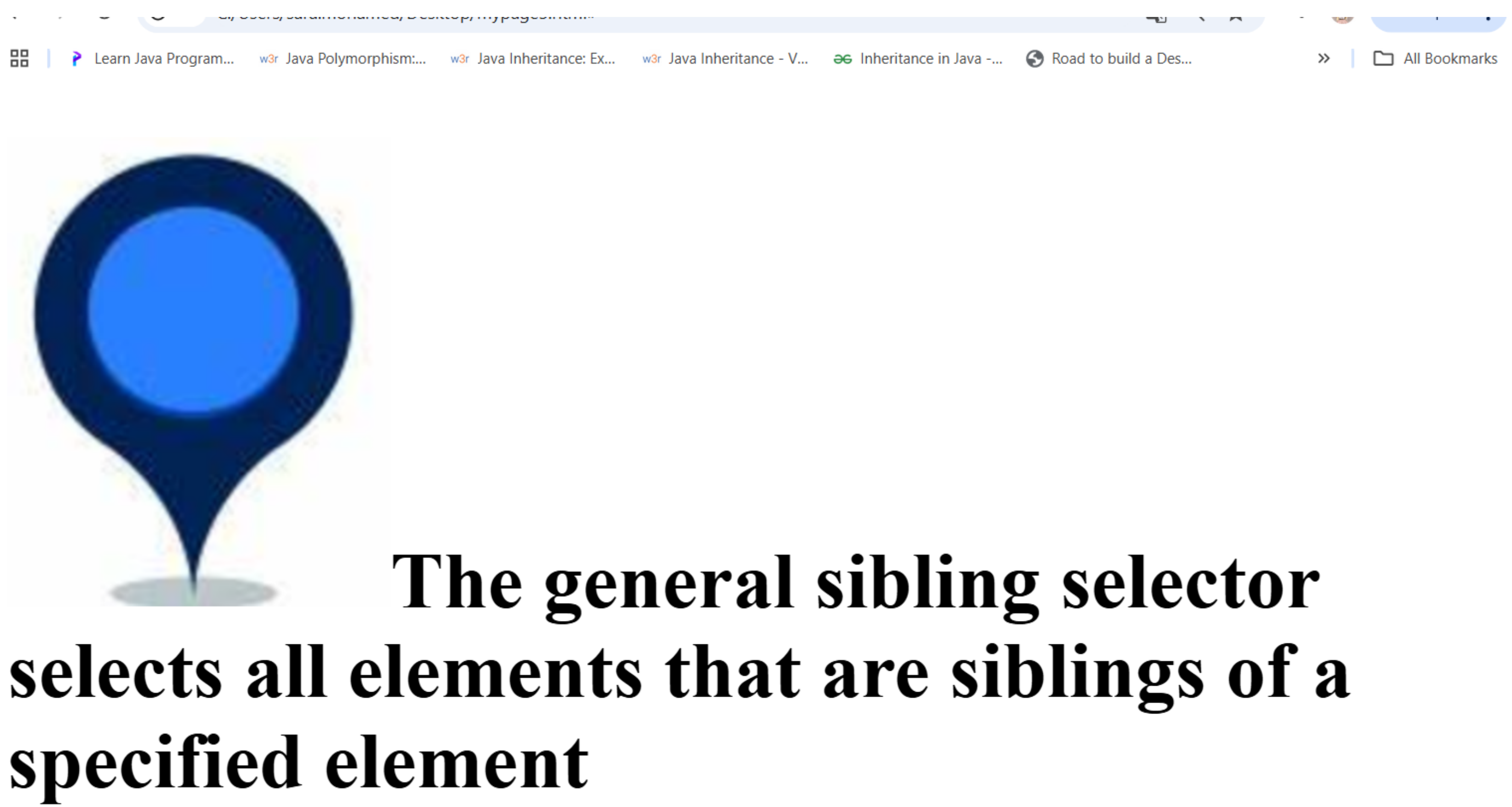
THIS IS AN EXAMPLE PARAGRAPH SHOWING HOW TO USE ::FIRST-LINE and ::first-letter in CSS. These pseudo-elements let you style the first line and first letter of text separately.

before Pseudo-element



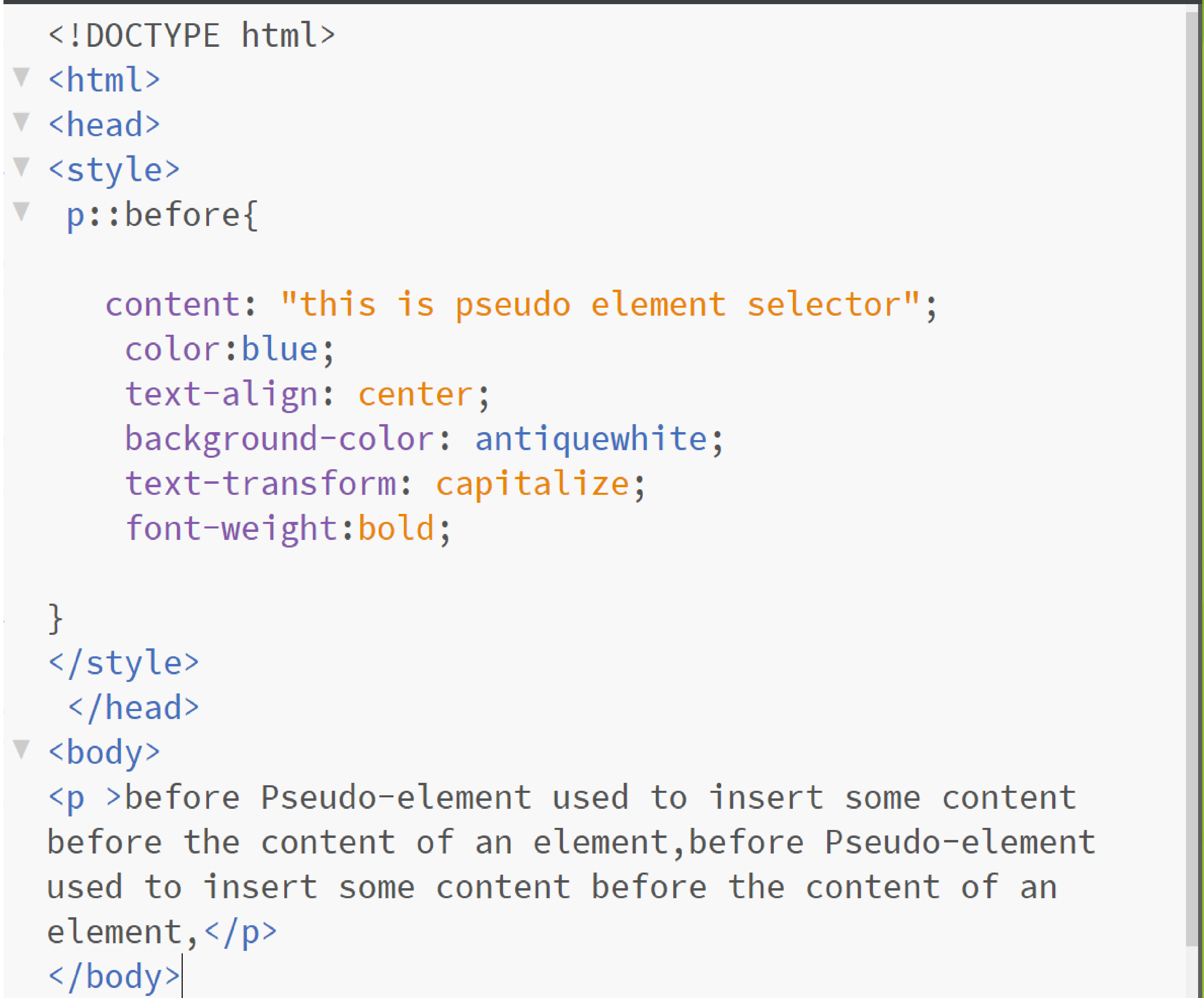
```
<!DOCTYPE html>
<html>
<head>
<style>
 h1::before {
 content: url(icon.png);
  }
</style>
 </head>
<body>
<h1>The general sibling selector selects all elements that are siblings of a specified element</h1>
</body>
</html>
```

before Pseudo-element used to insert some content before the content of an element,

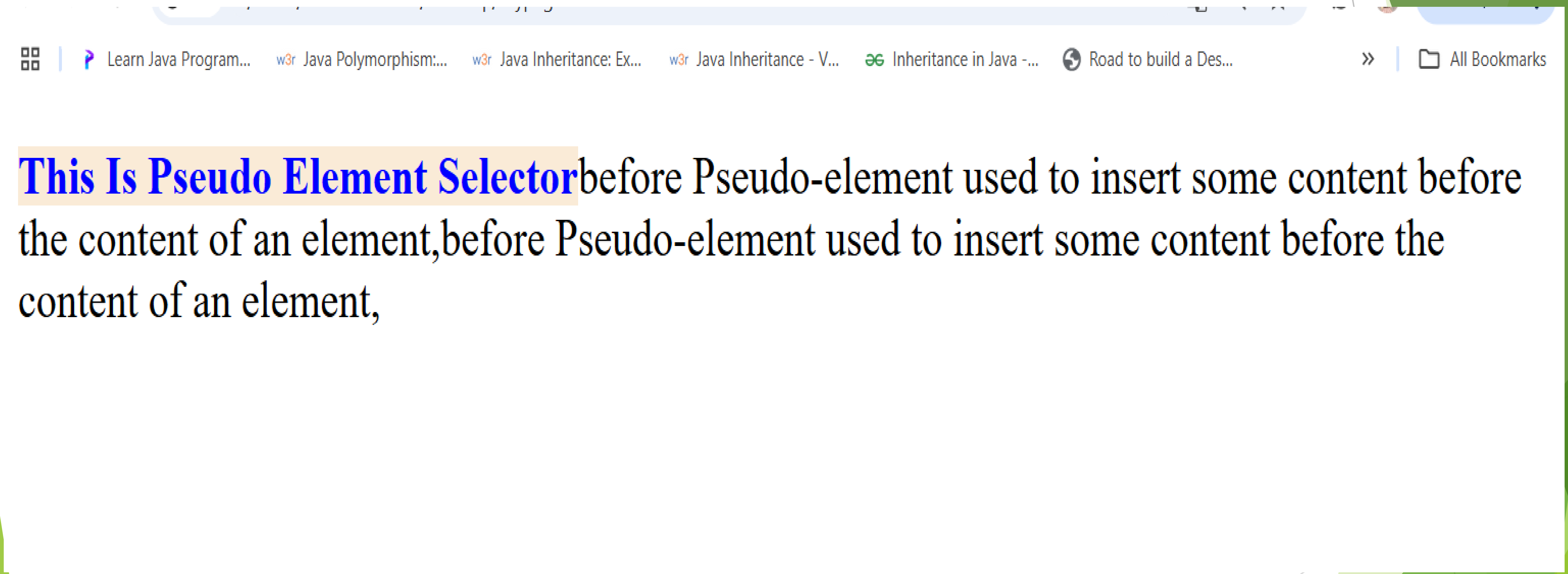


The general sibling selector selects all elements that are siblings of a specified element

Example2

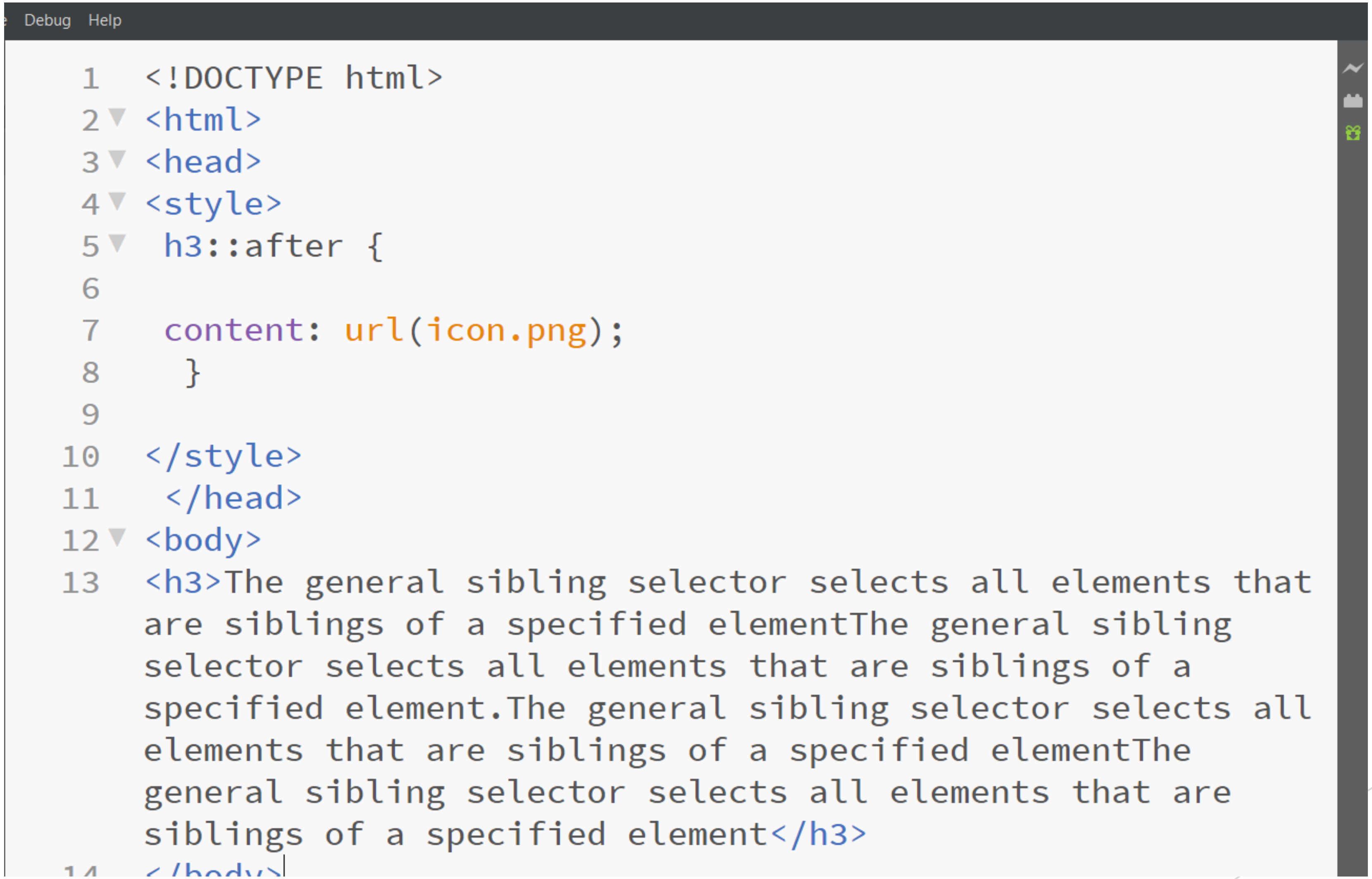


```
<!DOCTYPE html>
<html>
<head>
<style>
 p::before{

   content: "this is pseudo element selector";
    color:blue;
    text-align: center;
    background-color: antiquewhite;
    text-transform: capitalize;
    font-weight:bold;

}
</style>
 </head>
<body>
<p >before Pseudo-element used to insert some content
before the content of an element,before Pseudo-element
used to insert some content before the content of an
element,</p>
</body>
```



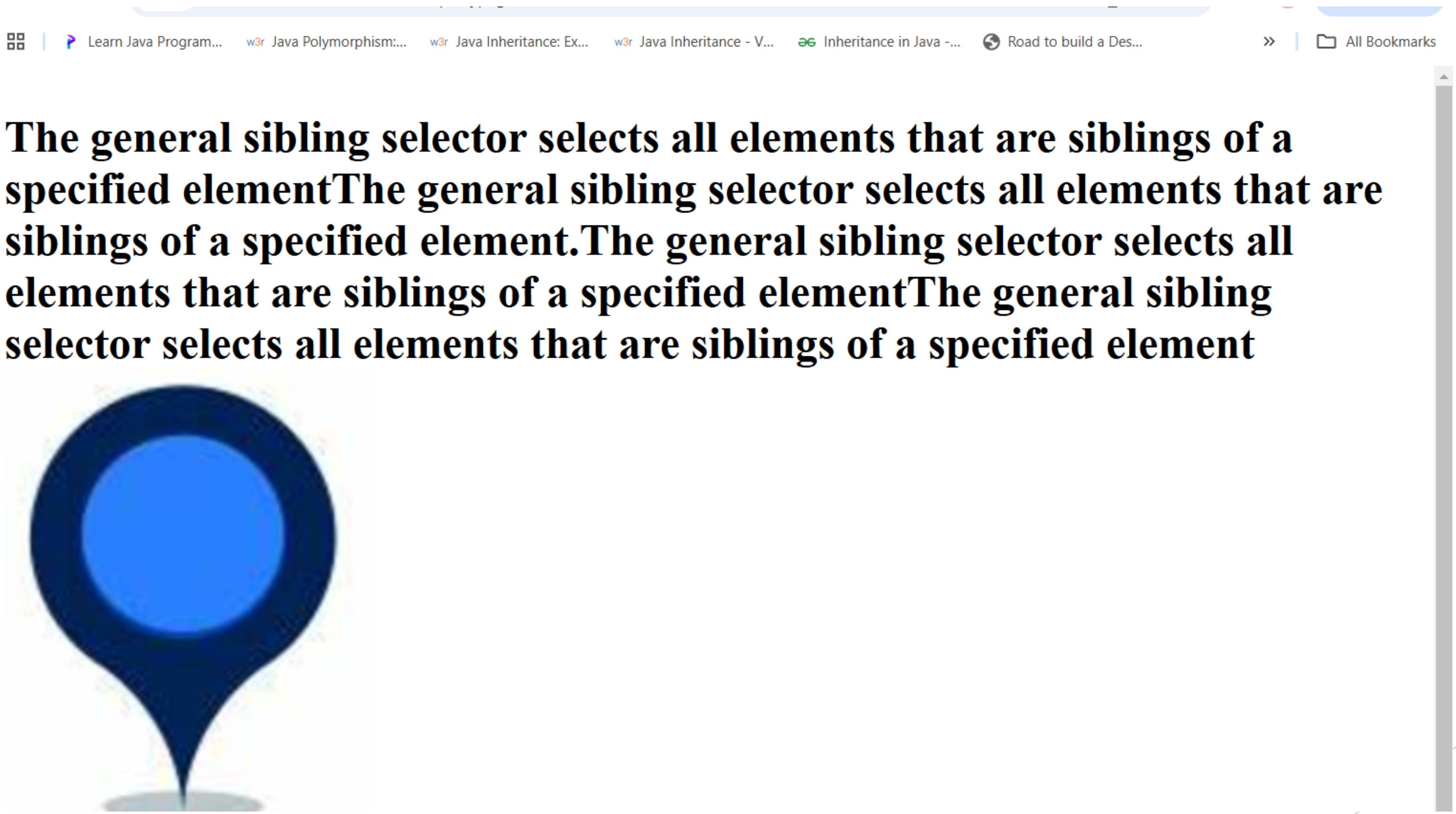
This Is Pseudo Element Selectorbefore Pseudo-element used to insert some content before the content of an element,before Pseudo-element used to insert some content before the content of an element,

after Pseudo-element used to insert some content after the content of an element

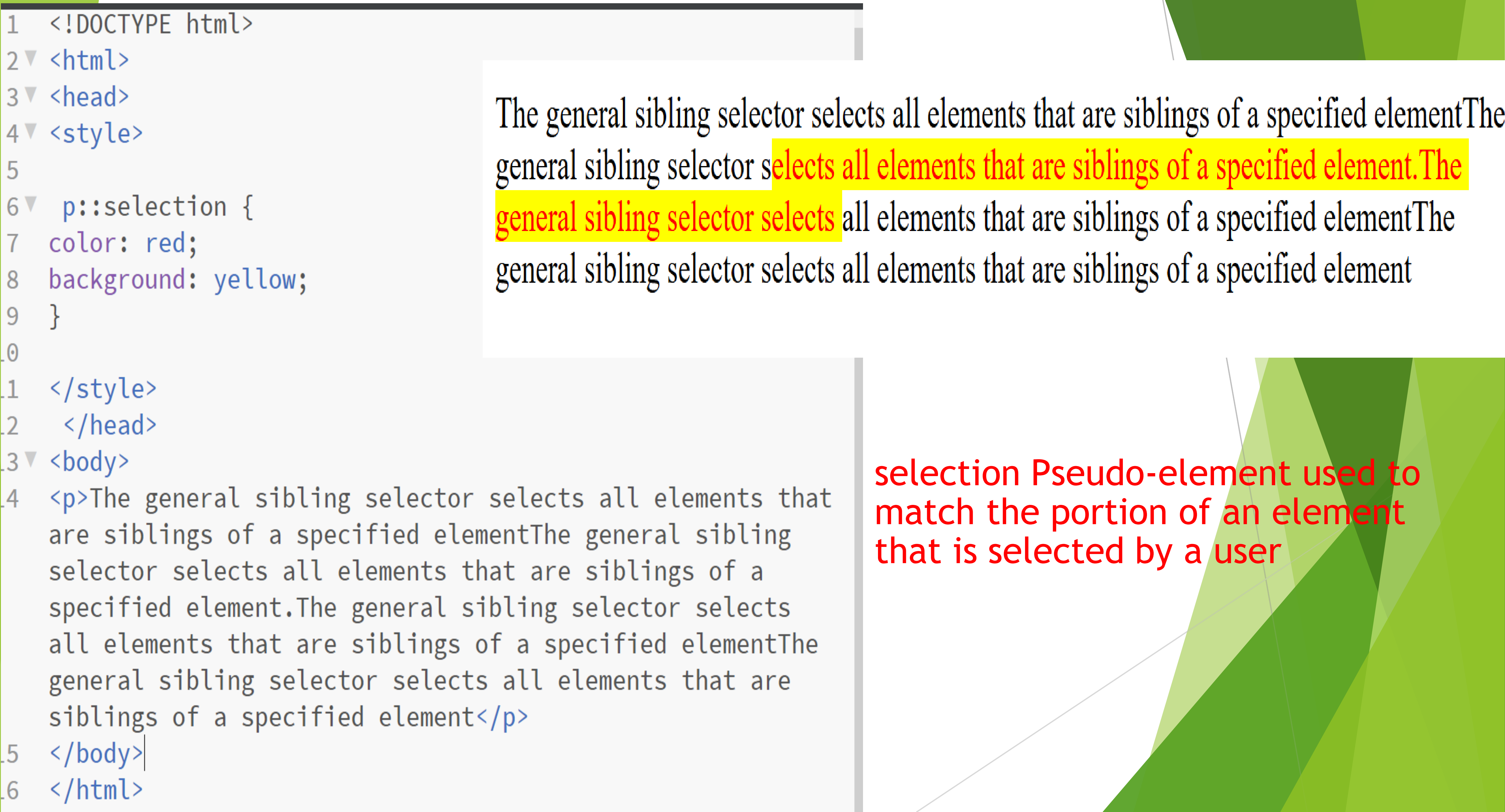


```
<!DOCTYPE html>
<html>
<head>
<style>
 h3::after {

 content: url(icon.png);
  }

</style>
 </head>
<body>
<h3>The general sibling selector selects all elements that are siblings of a specified elementThe general sibling selector selects all elements that are siblings of a specified element.The general sibling selector selects all elements that are siblings of a specified elementThe general sibling selector selects all elements that are siblings of a specified element</h3>
</body>
```

after Pseudo-element



The general sibling selector selects all elements that are siblings of a specified elementThe general sibling selector selects all elements that are siblings of a specified element.The general sibling selector selects all elements that are siblings of a specified elementThe general sibling selector selects all elements that are siblings of a specified element



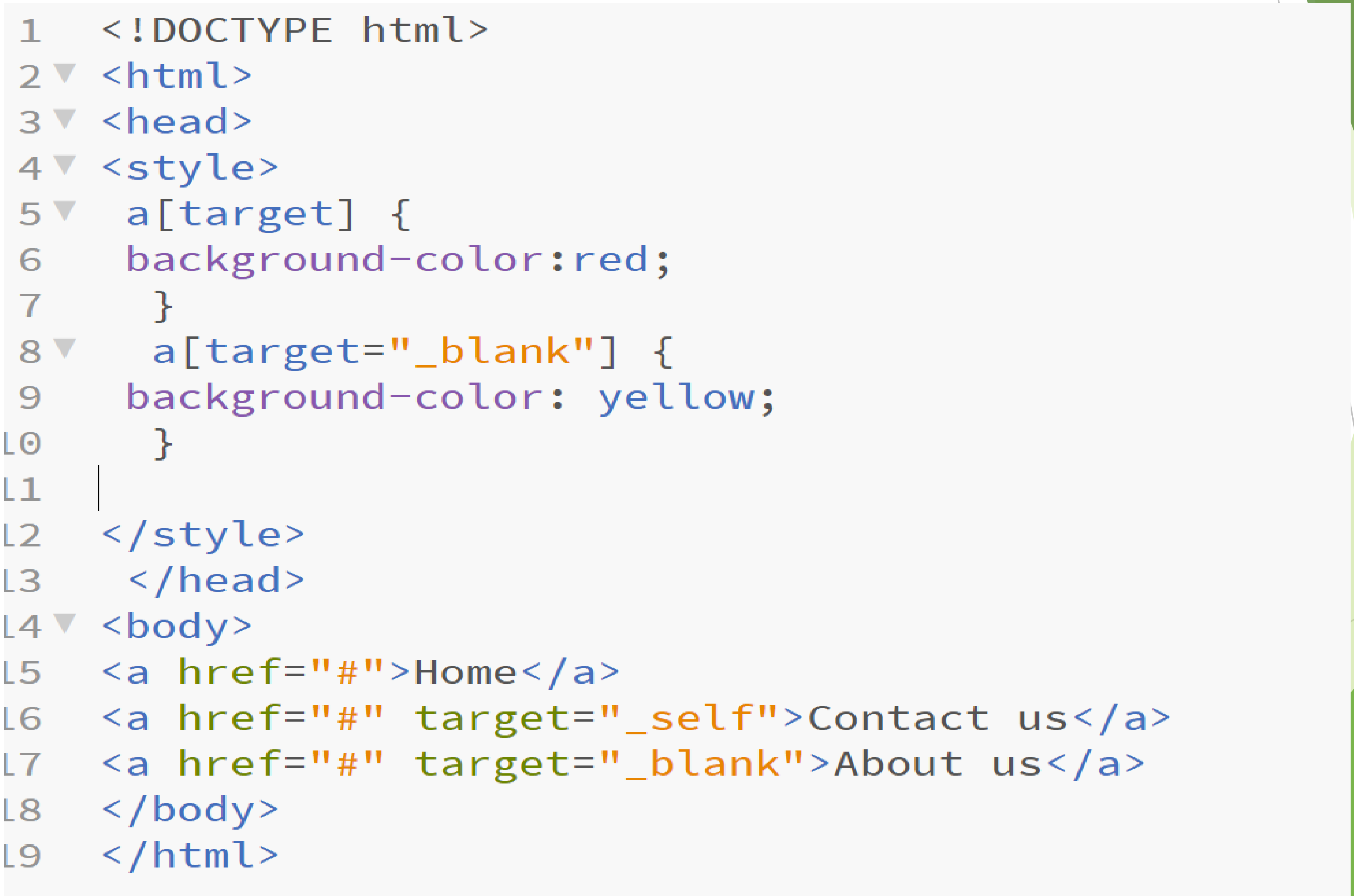
```
<!DOCTYPE html>
<html>
<head>
<style>

 p::selection {
color: red;
background: yellow;
}

</style>
 </head>
<body>
<p>The general sibling selector selects all elements that
are siblings of a specified elementThe general sibling
selector selects all elements that are siblings of a
specified element.The general sibling selector selects
all elements that are siblings of a specified elementThe
general sibling selector selects all elements that are
siblings of a specified element</p>
</body>
</html>
```

The general sibling selector selects all elements that are siblings of a specified elementThe general sibling selector selects all elements that are siblings of a specified element.The general sibling selector selects all elements that are siblings of a specified elementThe general sibling selector selects all elements that are siblings of a specified element

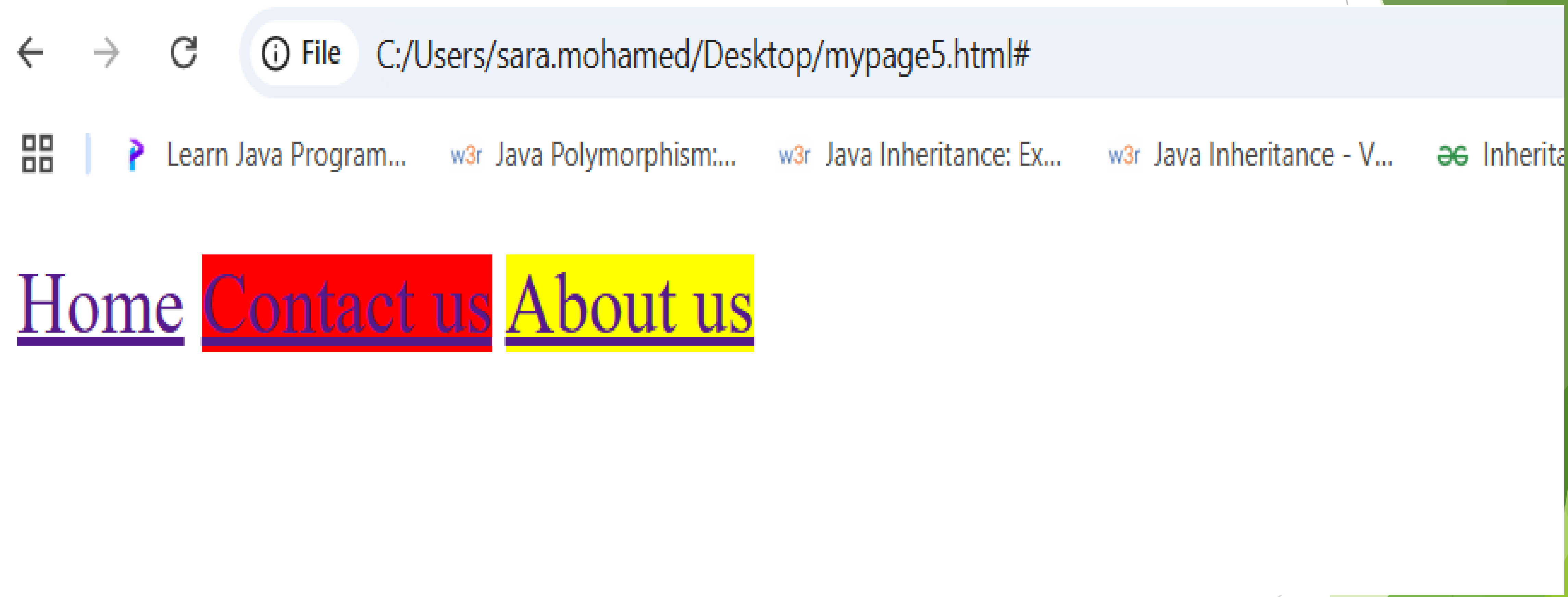
selection Pseudo-element used to match the portion of an element that is selected by a user

- Attribute selectors.

Used to style HTML elements that have specific attributes or attribute values selects all elements with a target attribute. You can also select elements with a specified attribute and value



```
<!DOCTYPE html>
<html>
<head>
<style>
 a[target] {
 background-color:red;
  }
  a[target="_blank"] {
 background-color: yellow;
  }

</style>
 </head>
<body>
<a href="#">Home</a>
<a href="#" target="_self">Contact us</a>
<a href="#" target="_blank">About us</a>
</body>
</html>
```

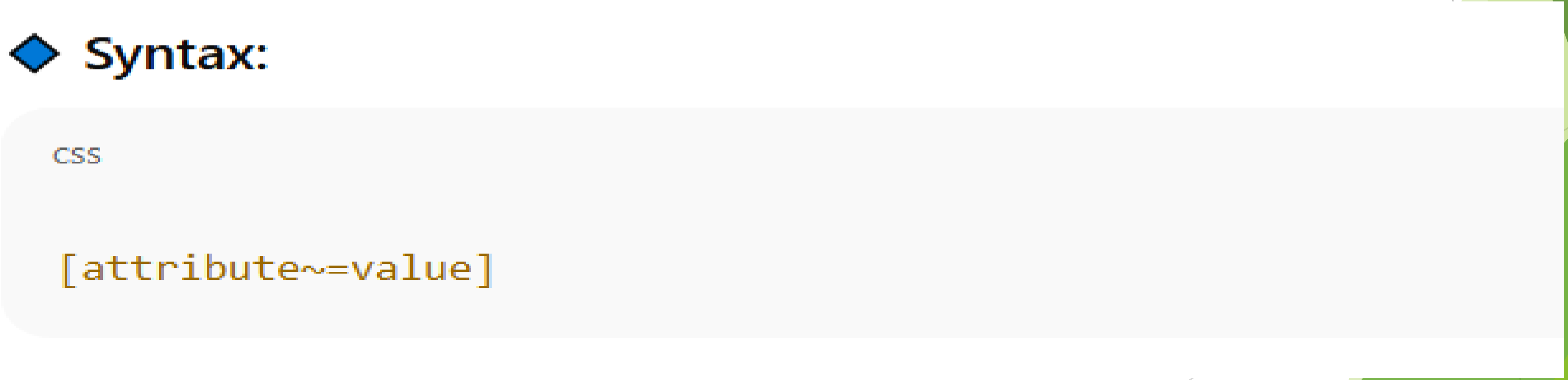

File
C:/Users/sara.mohamed/Desktop/mypage5.html#
Learn Java Program...
w3r Java Polymorphism:...
w3r Java Inheritance: Ex...
w3r Java Inheritance - V...
Inherita

Home Contact us About us

The attribute selectors can be useful for styling forms without class or ID

The attribute selector ~= in CSS

- It selects elements that have an attribute whose value contains the **whole word** value, separated by spaces.

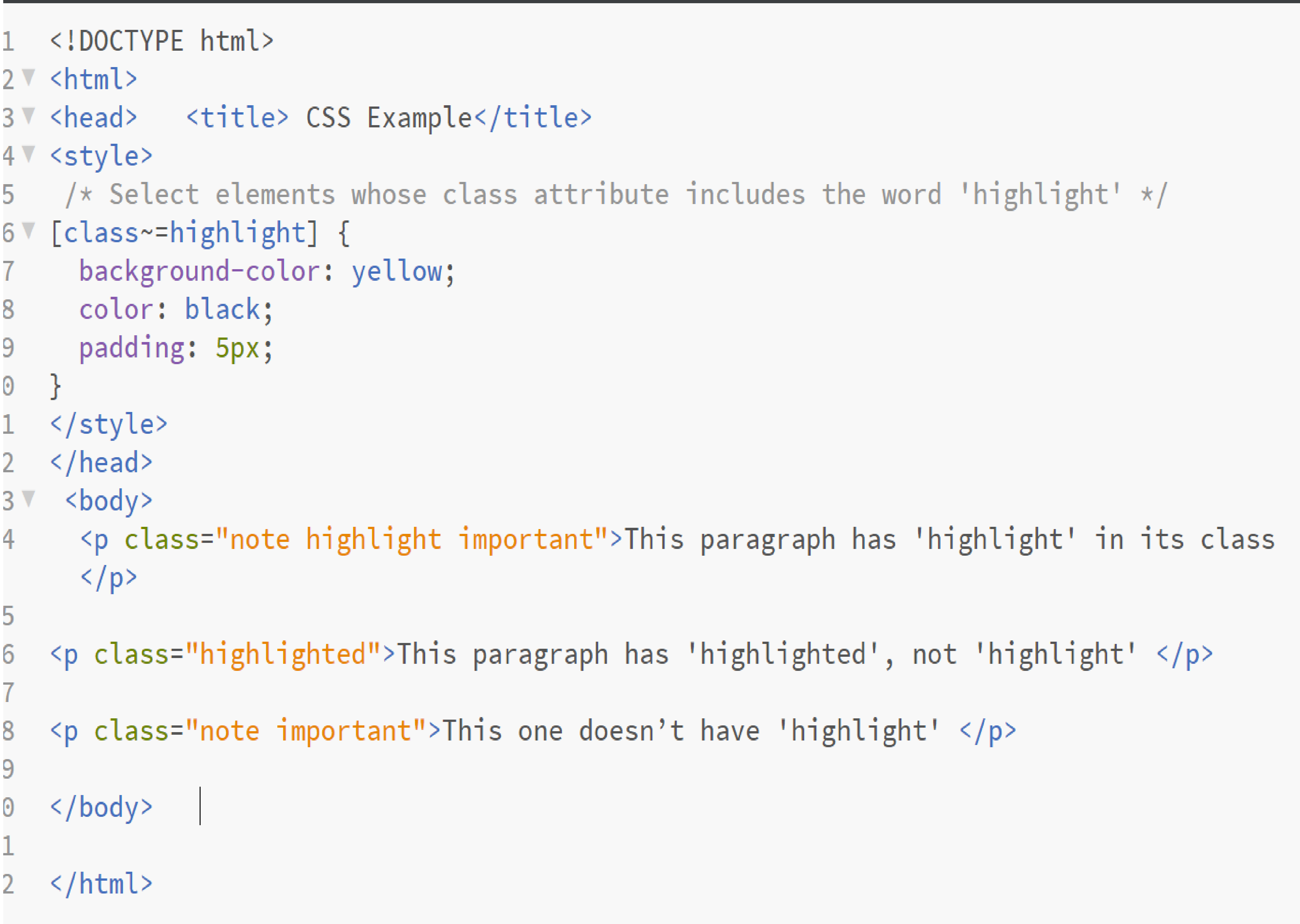


◆ **Syntax:**

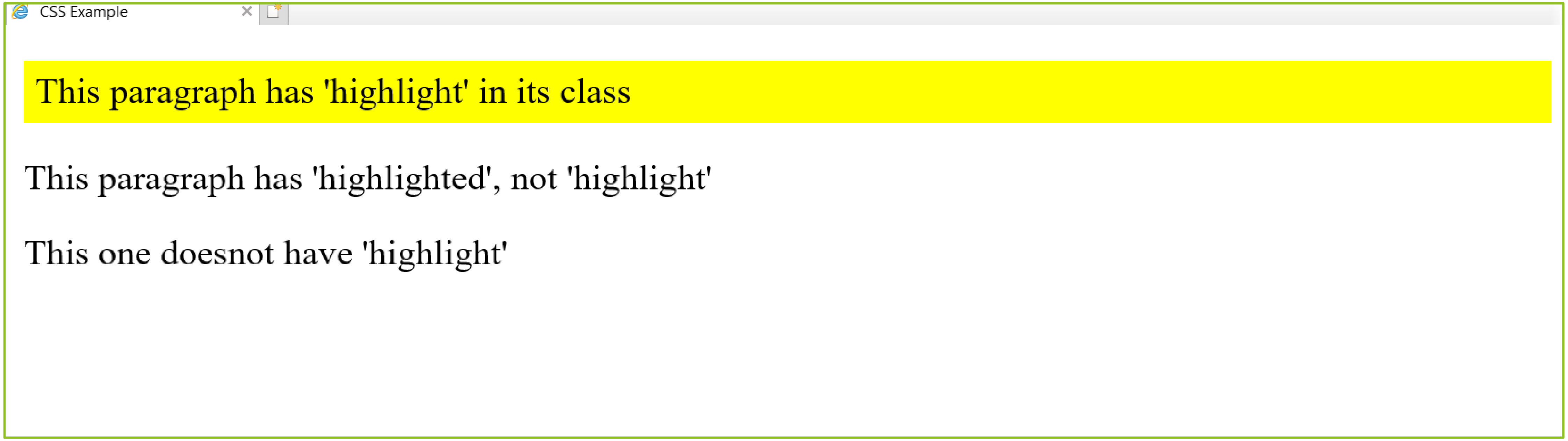
css

```css
[attribute~=value]
```

Example for **selector ~=**



```
<!DOCTYPE html>
<html>
<head>   <title> CSS Example</title>
<style>
 /* Select elements whose class attribute includes the word 'highlight' */
[class~=highlight] {
  background-color: yellow;
  color: black;
  padding: 5px;
}
</style>
</head>
 <body>
  <p class="note highlight important">This paragraph has 'highlight' in its class </p>

<p class="highlighted">This paragraph has 'highlighted', not 'highlight' </p>

<p class="note important">This one doesn’t have 'highlight' </p>

</body>

</html>
```



This paragraph has 'highlight' in its class

This paragraph has 'highlighted', not 'highlight'

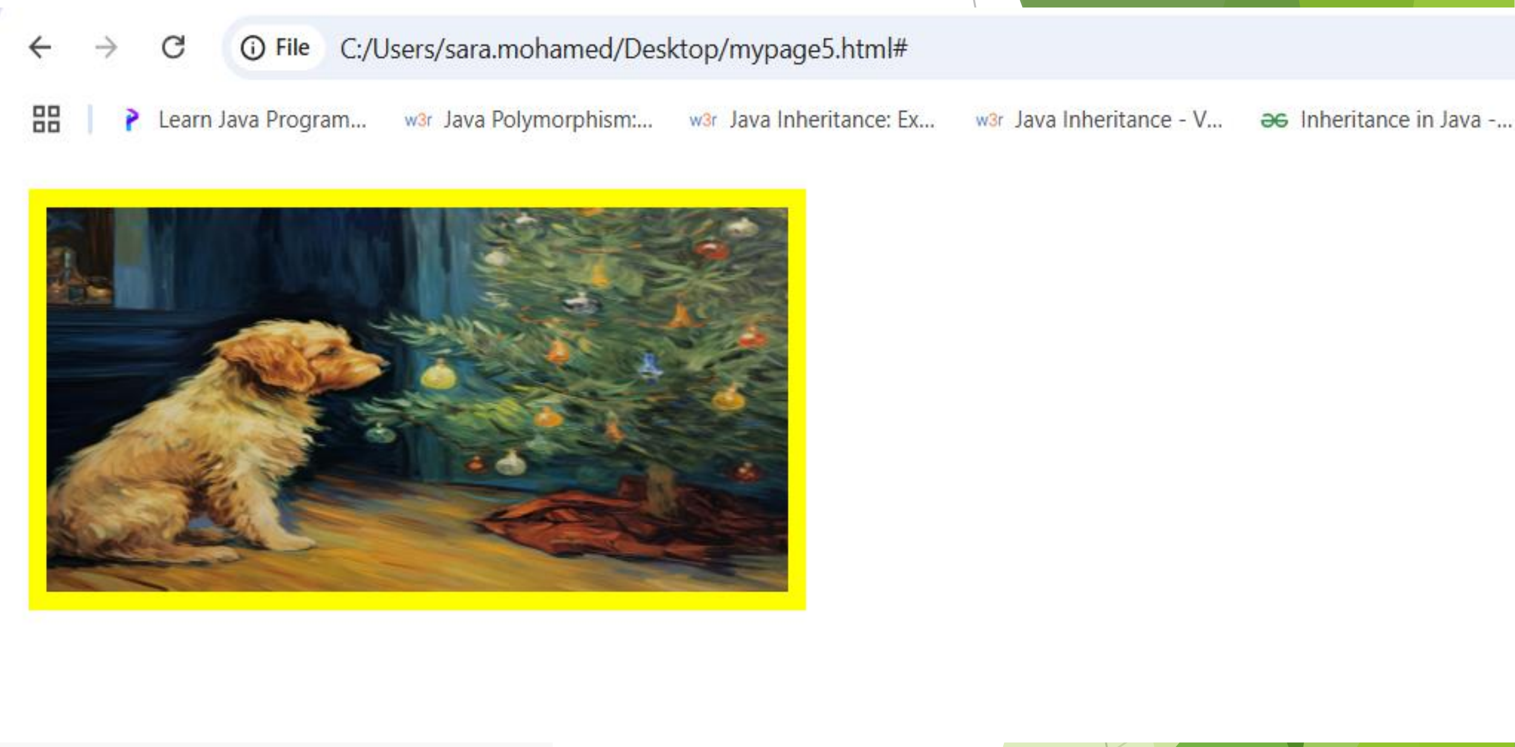
This one doesnot have 'highlight'

🔶 Explanation:

- ✅ The first `<p>` is styled because its `class` attribute includes the **word** `highlight`.
- ❌ The second `<p>` is **not** matched, because `"highlighted"` is **not the same word** as `"highlight"`.
- ❌ The third `<p>` is ignored because it lacks `"highlight"` entirely.

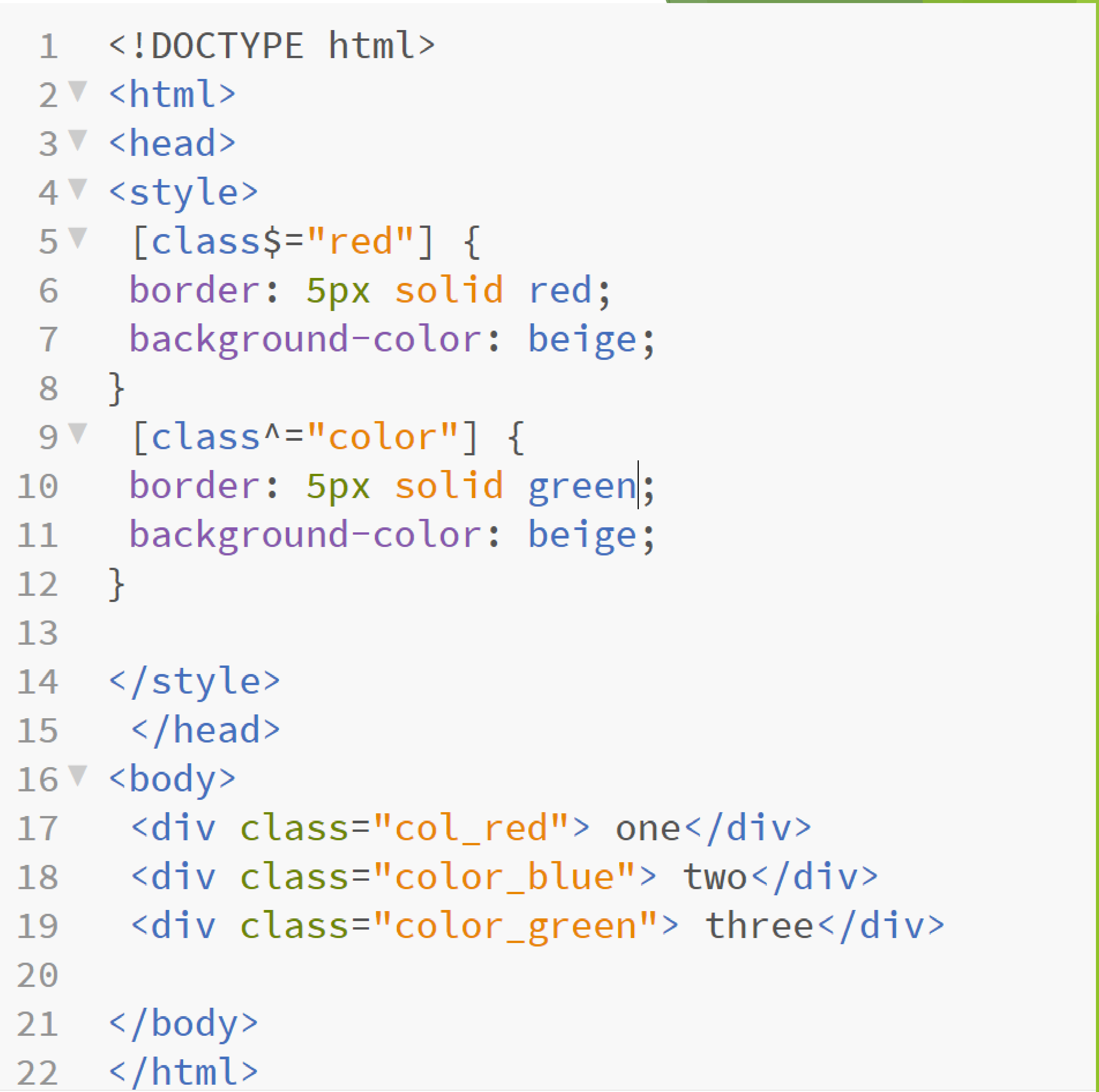
Attribute contains a value selector

```
<!DOCTYPE html>
<html>
<head>
<style>
 [title~="flower"] {
 border: 5px solid yellow;
}

</style>
 </head>
<body>
<img src="dog.jpg"  width="200" height="100"
title="flower"/>

</body>
</html>
```

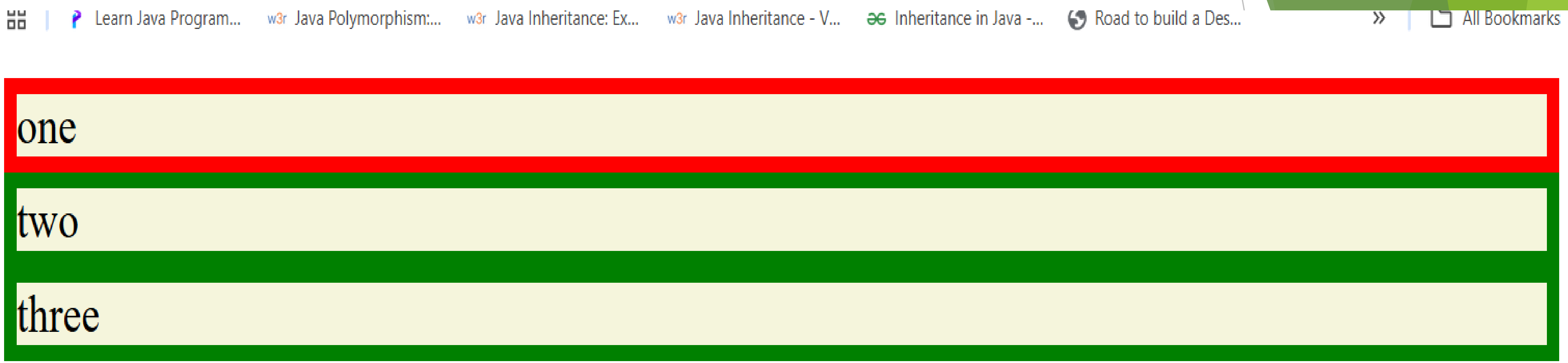


Begins with a specified value

Ends with a specified value



```
<!DOCTYPE html>
<html>
<head>
<style>
 [class$="red"] {
 border: 5px solid red;
 background-color: beige;
}
 [class^="color"] {
 border: 5px solid green;
 background-color: beige;
}

</style>
 </head>
<body>
 <div class="col_red"> one</div>
 <div class="color_blue"> two</div>
 <div class="color_green"> three</div>

</body>
</html>
```

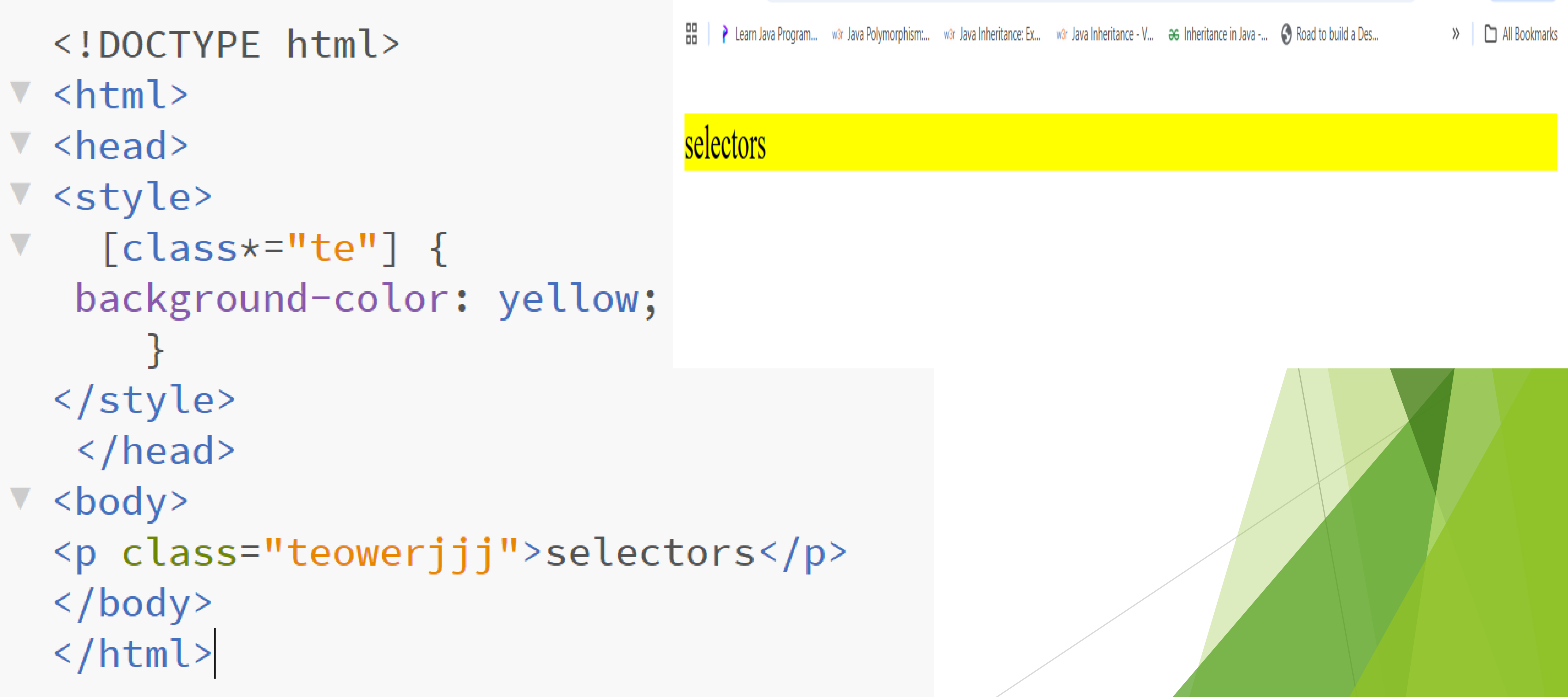


one

two

three

Contains a specified value



```
<!DOCTYPE html>
<html>
<head>
<style>
  [class*="te"] {
 background-color: yellow;
     }
</style>
 </head>
<body>
<p class="teowerjjj">selectors</p>
</body>
</html>
```

selectors

Thank you