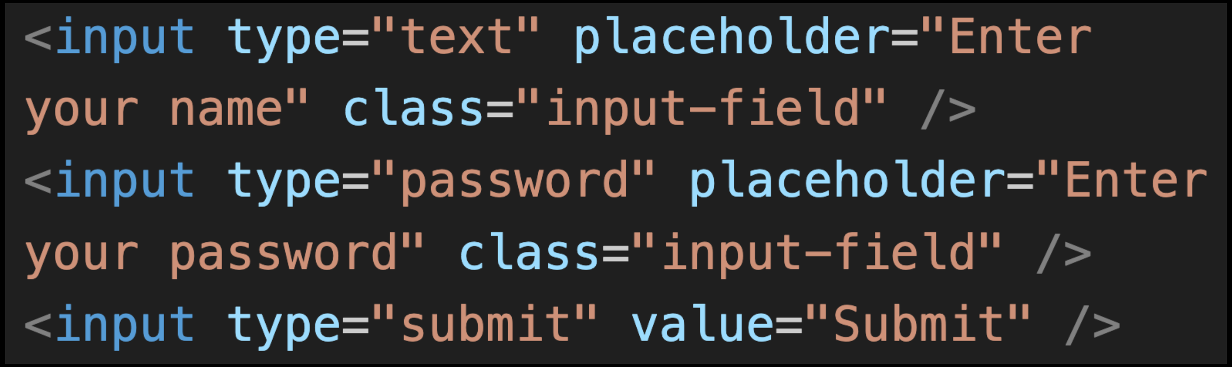
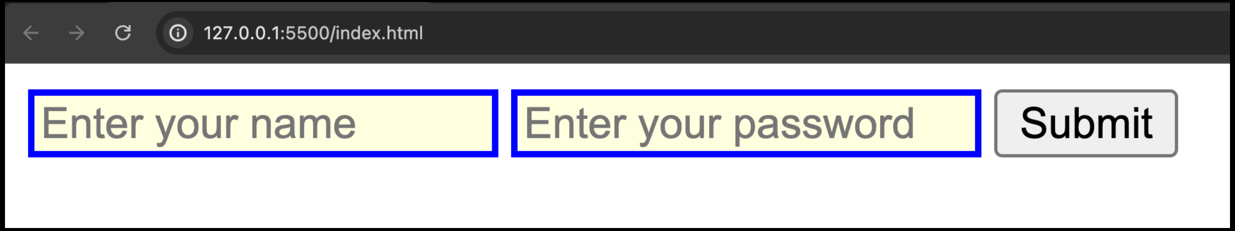
Selectors (Attribute selector)



```
<input type="text" placeholder="Enter
your name" class="input-field" />
<input type="password" placeholder="Enter
your password" class="input-field" />
<input type="submit" value="Submit" />
```

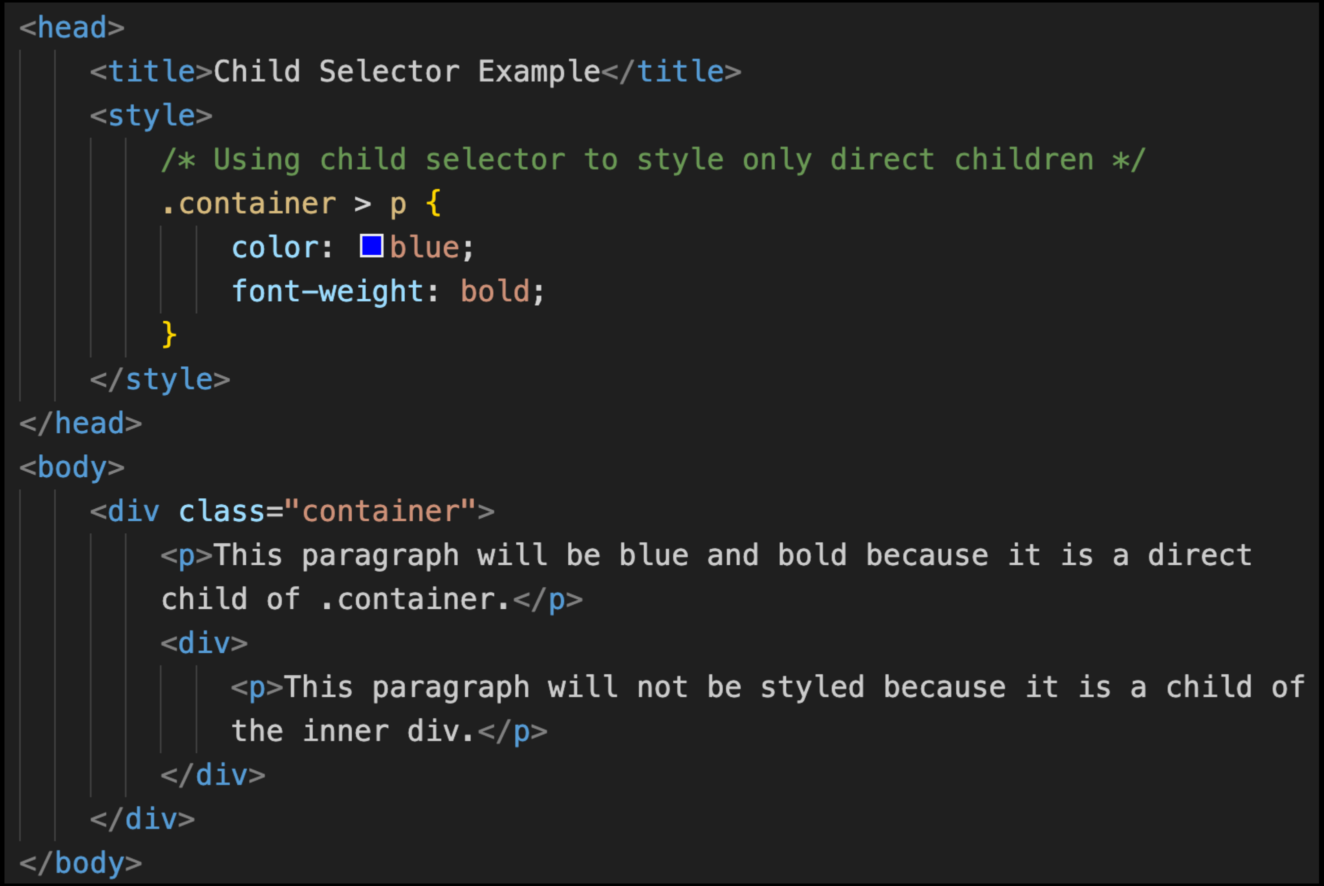


Syntax:

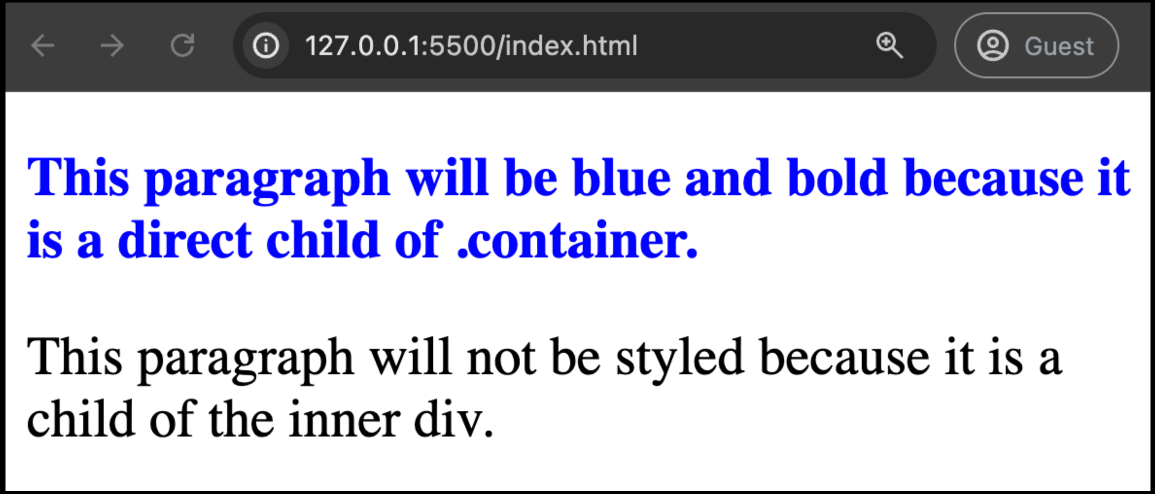
- Basic Attribute: [attribute]
- Exact match: [attribute="value"]
- Starts with: [attribute^="value"]
- Ends with: [attribute$="value"]
- Contains: [attribute*="value"]

- Attribute selectors are used to select elements based on their attribute values.
- Attribute values are case-sensitive.
- Useful for selecting elements without adding additional classes or IDs.

Selectors (Child selector)



```
<head>
    <title>Child Selector Example</title>
    <style>
        /* Using child selector to style only direct children */
        .container > p {
            color: blue;
            font-weight: bold;
        }
    </style>
</head>
<body>
    <div class="container">
        <p>This paragraph will be blue and bold because it is a direct
        child of .container.</p>
        <div>
            <p>This paragraph will not be styled because it is a child of
            the inner div.</p>
        </div>
    </div>
</body>
```



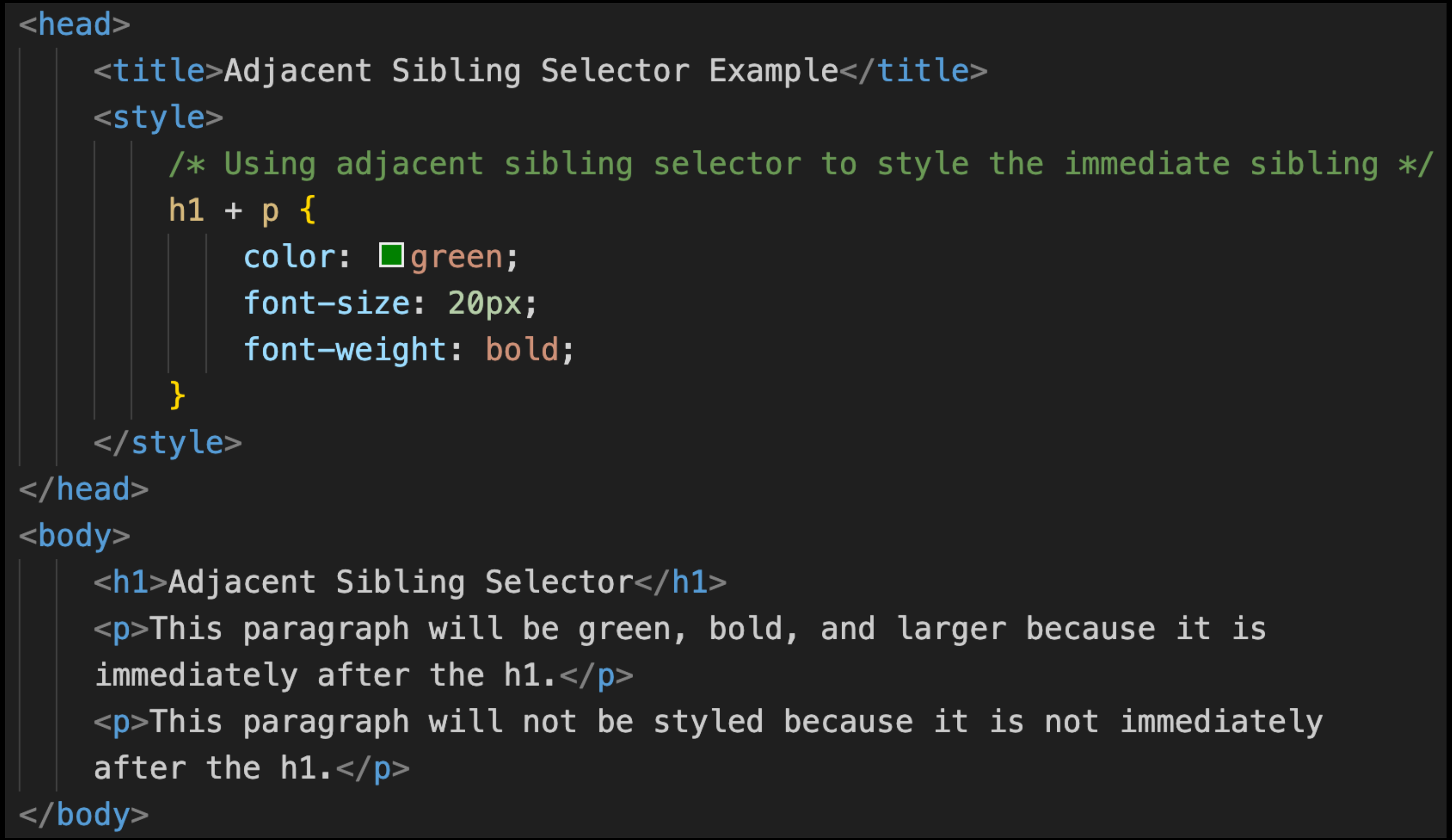
127.0.0.1:5500/index.html

This paragraph will be blue and bold because it is a direct child of .container.

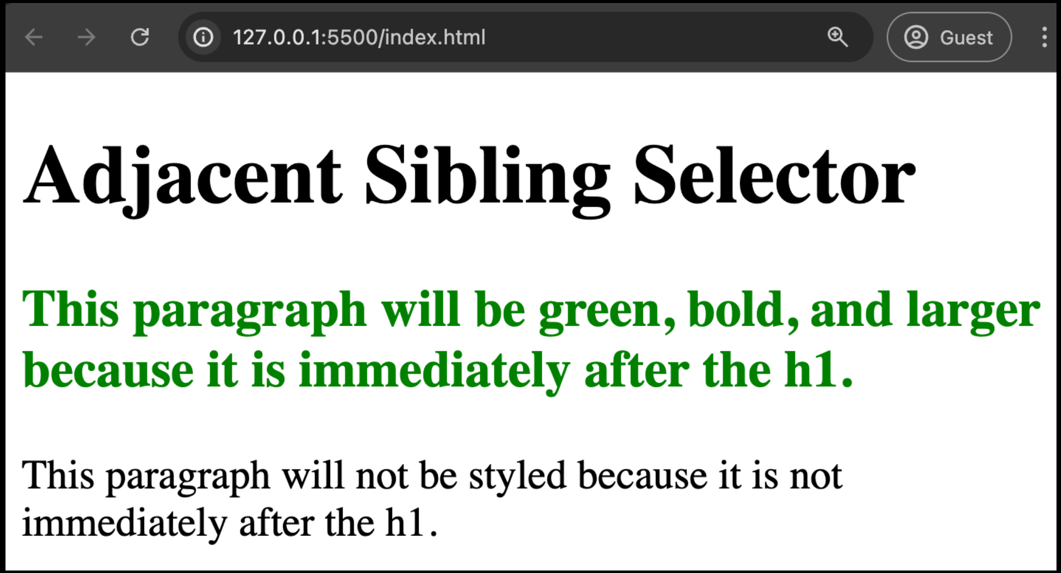
This paragraph will not be styled because it is a child of the inner div.

- The child selector in CSS is used to select only the direct children of a specified element.
- The child selector is denoted by the > symbol.
- It targets immediate children elements, not grandchildren or other descendants.
- The child selector is more specific than the descendant selector (space), which selects all descendants regardless of their depth in the hierarchy.

Selectors (Adjacent Sibling selector)



```html
<head>
    <title>Adjacent Sibling Selector Example</title>
    <style>
        /* Using adjacent sibling selector to style the immediate sibling */
        h1 + p {
            color: 🟩green;
            font-size: 20px;
            font-weight: bold;
        }
    </style>
</head>
<body>
    <h1>Adjacent Sibling Selector</h1>
    <p>This paragraph will be green, bold, and larger because it is
    immediately after the h1.</p>
    <p>This paragraph will not be styled because it is not immediately
    after the h1.</p>
</body>
```



127.0.0.1:5500/index.html

Guest

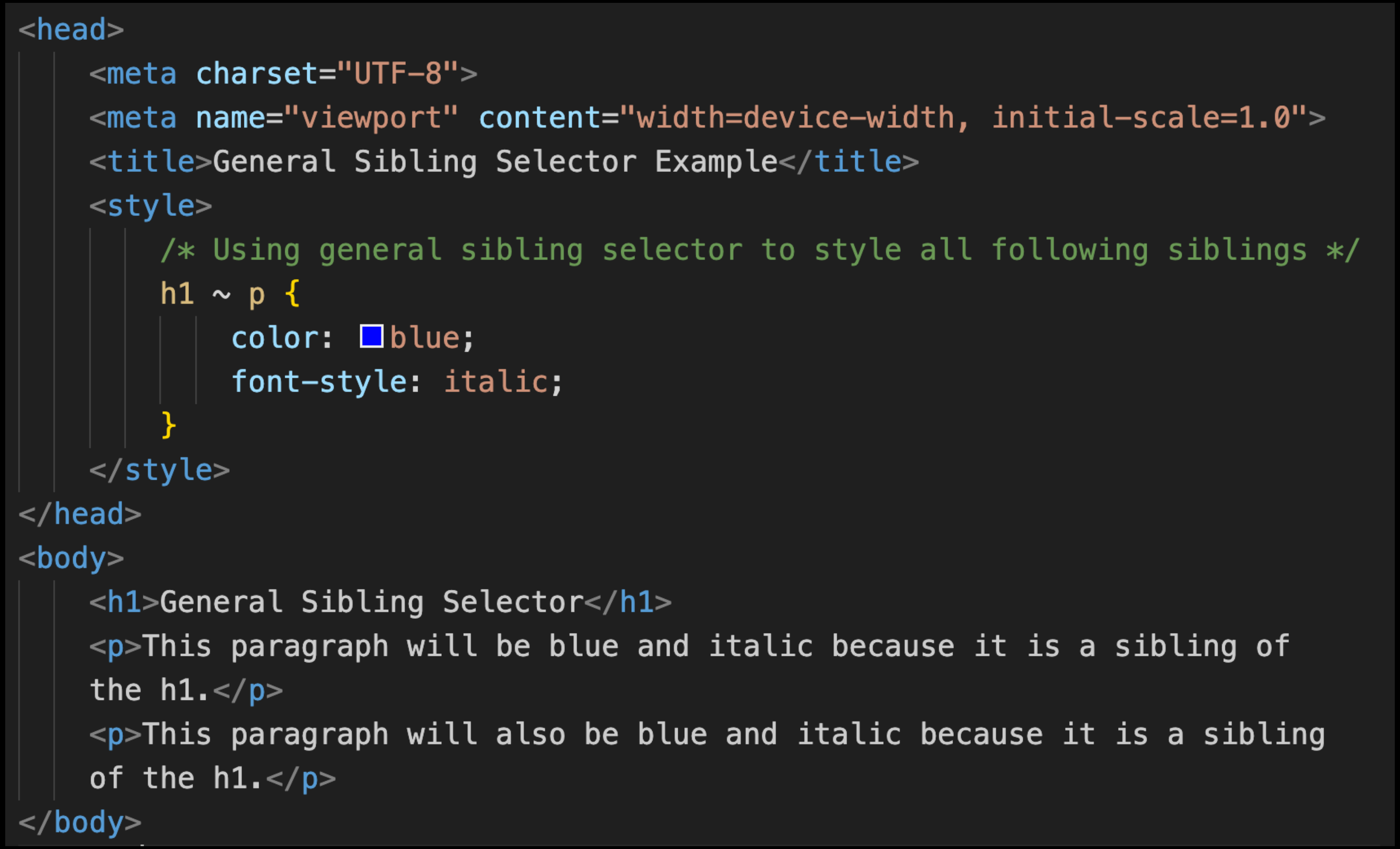
Adjacent Sibling Selector

This paragraph will be green, bold, and larger because it is immediately after the h1.

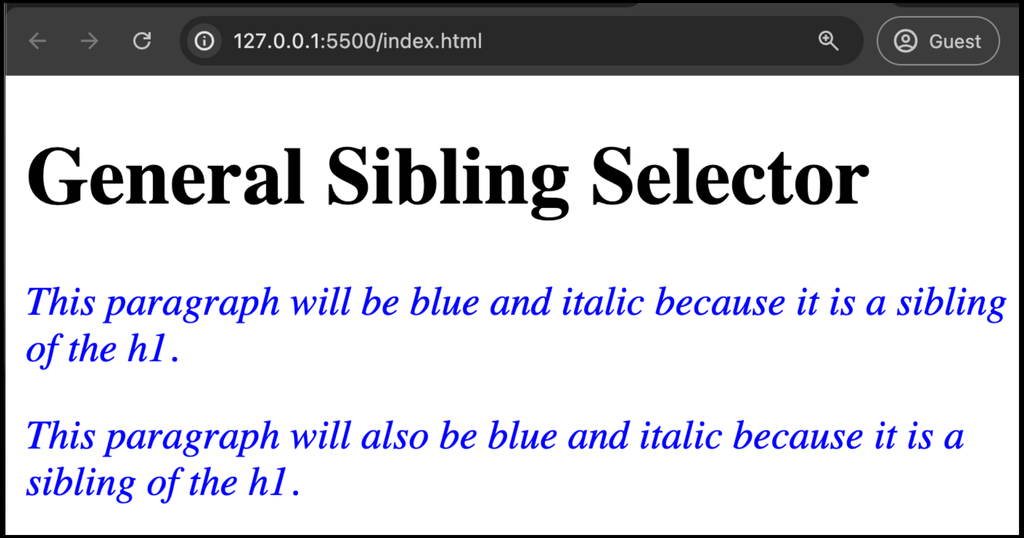
This paragraph will not be styled because it is not immediately after the h1.

- The adjacent sibling selector in CSS selects an element that is immediately preceded by a specified element.
- It targets the element that comes directly after the specified element.
- The adjacent sibling selector is specific to the immediate following sibling and does not affect any other siblings.

Selectors (General Sibling selector)



```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>General Sibling Selector Example</title>
    <style>
        /* Using general sibling selector to style all following siblings */
        h1 ~ p {
            color: blue;
            font-style: italic;
        }
    </style>
</head>
<body>
    <h1>General Sibling Selector</h1>
    <p>This paragraph will be blue and italic because it is a sibling of
    the h1.</p>
    <p>This paragraph will also be blue and italic because it is a sibling
    of the h1.</p>
</body>
```



127.0.0.1:5500/index.html

Guest

General Sibling Selector

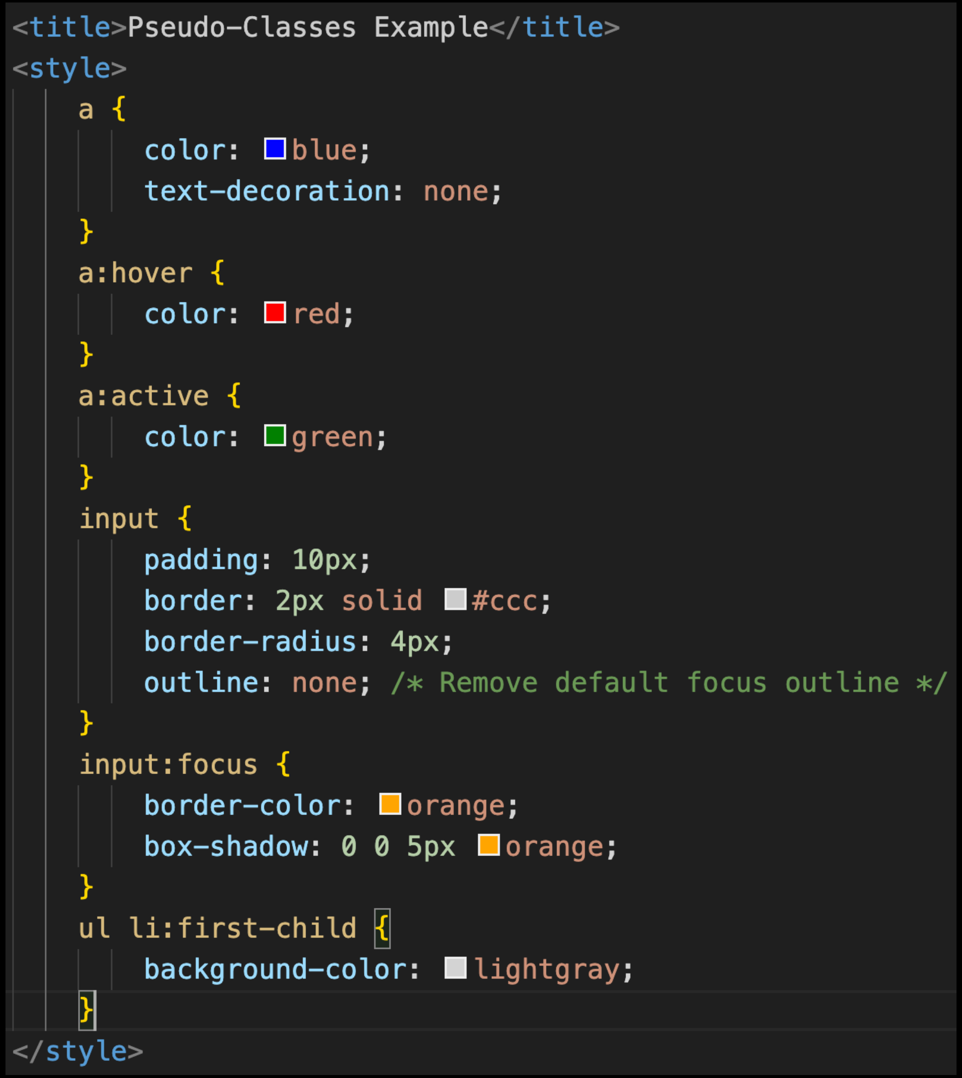
This paragraph will be blue and italic because it is a sibling of the h1.

This paragraph will also be blue and italic because it is a sibling of the h1.

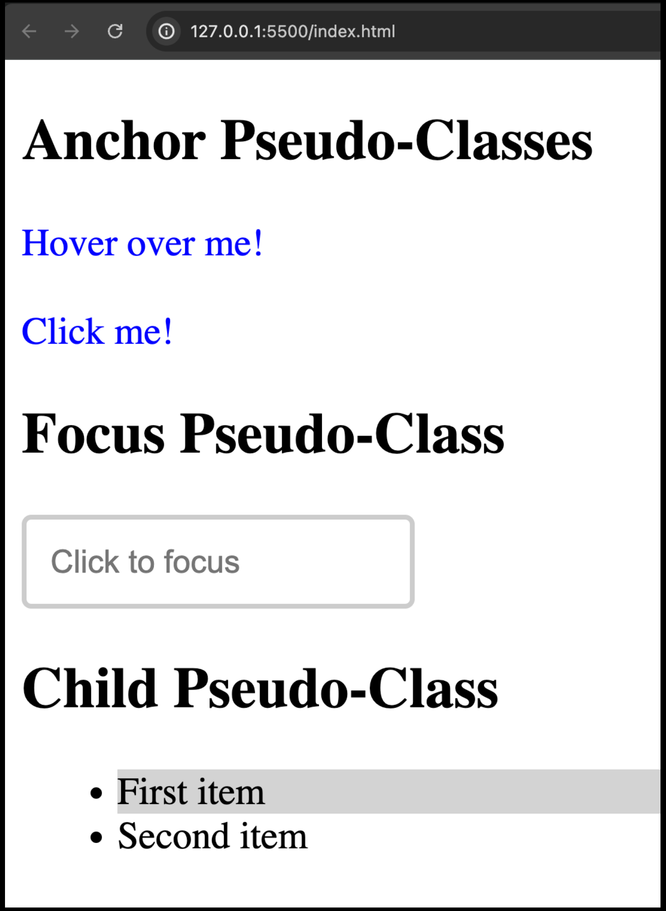
- The general sibling selector in CSS selects all elements that are siblings of a specified element, regardless of their position.
- It targets all siblings that follow the specified element, not just the immediate one.
- Currently, there is no direct CSS selector to target previous siblings.

Pseudo Classes

- Pseudo-classes are keywords added to selectors that specify a special state of the selected elements.
- They help to style elements based on their state, such as when an element is hovered over or when a form input is checked.
- Syntax: selector:pseudo-class { styles }.
- Common examples: :hover, :active, :focus, :first-child.



```
<title>Pseudo-Classes Example</title>
<style>
    a {
        color: blue;
        text-decoration: none;
    }
    a:hover {
        color: red;
    }
    a:active {
        color: green;
    }
    input {
        padding: 10px;
        border: 2px solid #ccc;
        border-radius: 4px;
        outline: none; /* Remove default focus outline */
    }
    input:focus {
        border-color: orange;
        box-shadow: 0 0 5px orange;
    }
    ul li:first-child {
        background-color: lightgray;
    }
</style>
```



127.0.0.1:5500/index.html

Anchor Pseudo-Classes

Hover over me!

Click me!

Focus Pseudo-Class

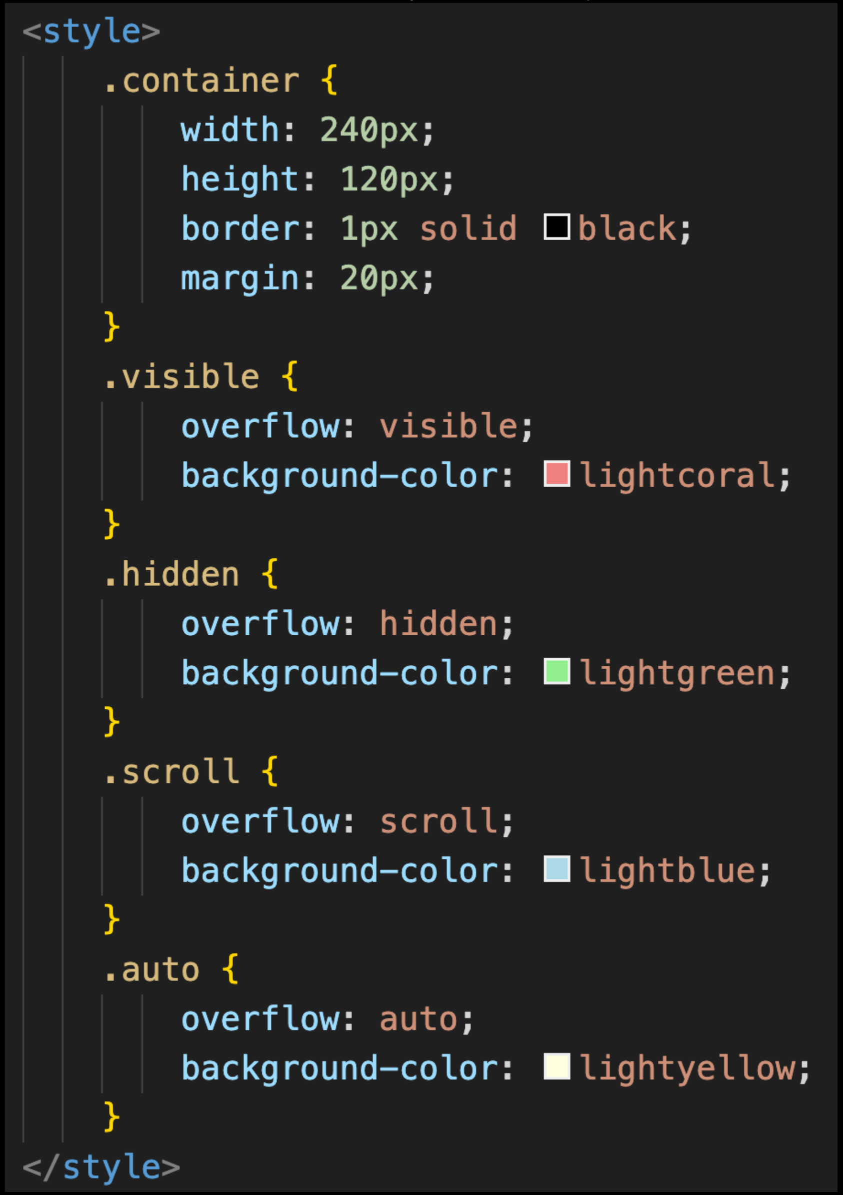
Click to focus

Child Pseudo-Class

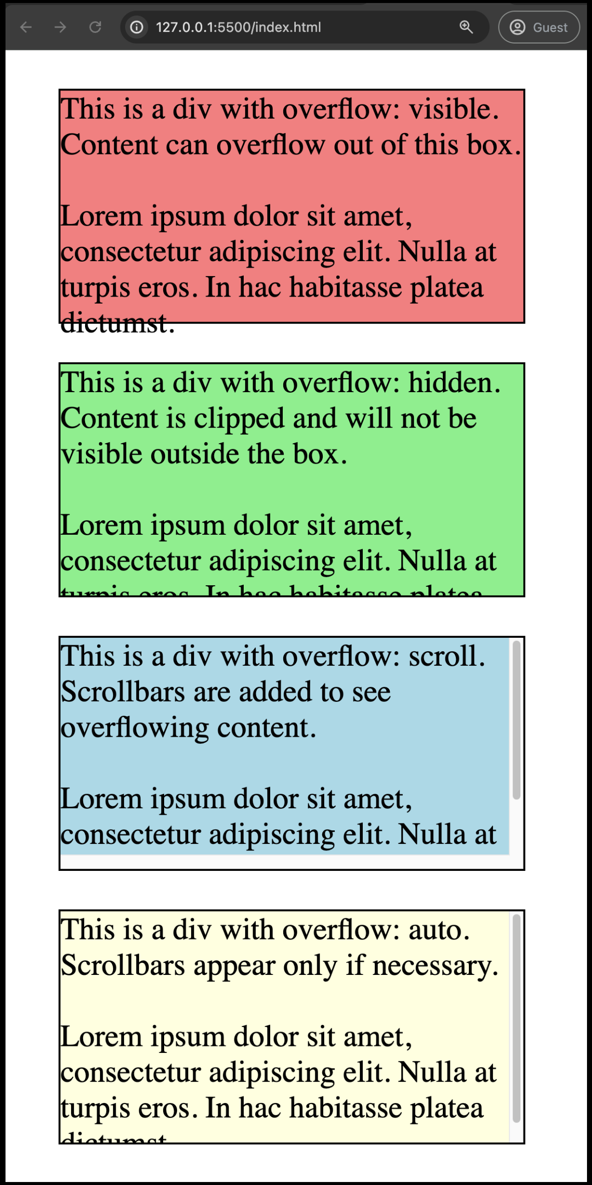
- First item
- Second item

Overflow Property

- Purpose: Controls how content is handled when it overflows an element’s box.
- visible: Default; content is not clipped and may overflow the element’s box.
- hidden: Content is clipped and not visible beyond the element’s box.
- scroll: Content is clipped, but scrollbars are added to allow scrolling.
- auto: Scrollbars are added only when necessary to see the overflowing content.



```
<style>
    .container {
        width: 240px;
        height: 120px;
        border: 1px solid black;
        margin: 20px;
    }
    .visible {
        overflow: visible;
        background-color: lightcoral;
    }
    .hidden {
        overflow: hidden;
        background-color: lightgreen;
    }
    .scroll {
        overflow: scroll;
        background-color: lightblue;
    }
    .auto {
        overflow: auto;
        background-color: lightyellow;
    }
</style>
```



127.0.0.1:5500/index.html

This is a div with overflow: visible. Content can overflow out of this box.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla at turpis eros. In hac habitasse platea dictumst.

This is a div with overflow: hidden. Content is clipped and will not be visible outside the box.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla at

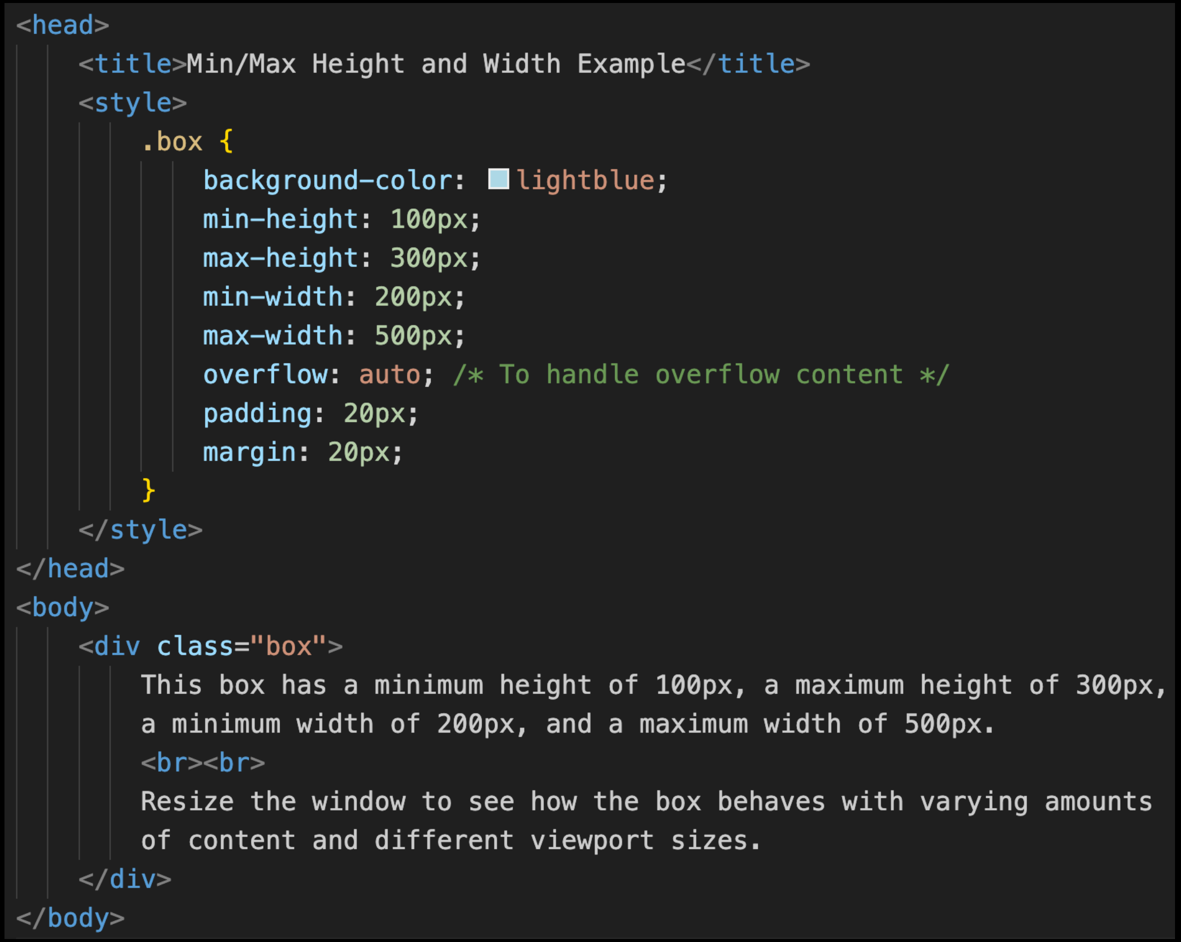
This is a div with overflow: scroll. Scrollbars are added to see overflowing content.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla at

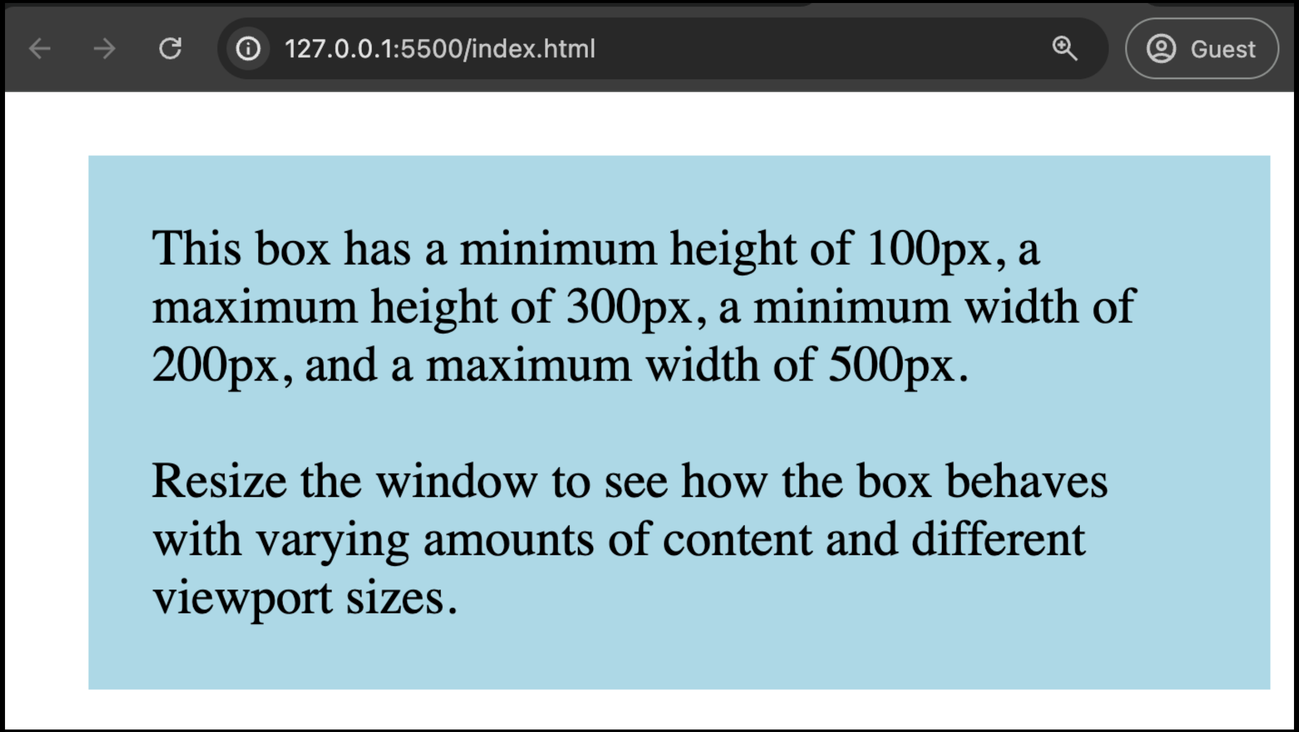
This is a div with overflow: auto. Scrollbars appear only if necessary.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla at turpis eros. In hac habitasse platea

Min-Max Height and Width



```
<head>
    <title>Min/Max Height and Width Example</title>
    <style>
        .box {
            background-color: lightblue;
            min-height: 100px;
            max-height: 300px;
            min-width: 200px;
            max-width: 500px;
            overflow: auto; /* To handle overflow content */
            padding: 20px;
            margin: 20px;
        }
    </style>
</head>
<body>
    <div class="box">
        This box has a minimum height of 100px, a maximum height of 300px,
        a minimum width of 200px, and a maximum width of 500px.
        <br><br>
        Resize the window to see how the box behaves with varying amounts
        of content and different viewport sizes.
    </div>
</body>
```



This box has a minimum height of 100px, a maximum height of 300px, a minimum width of 200px, and a maximum width of 500px.

Resize the window to see how the box behaves with varying amounts of content and different viewport sizes.

- min-height: Sets the minimum height an element can be.
- max-height: Sets the maximum height an element can be.
- min-width: Sets the minimum width an element can be.
- max-width: Sets the maximum width an element can be.