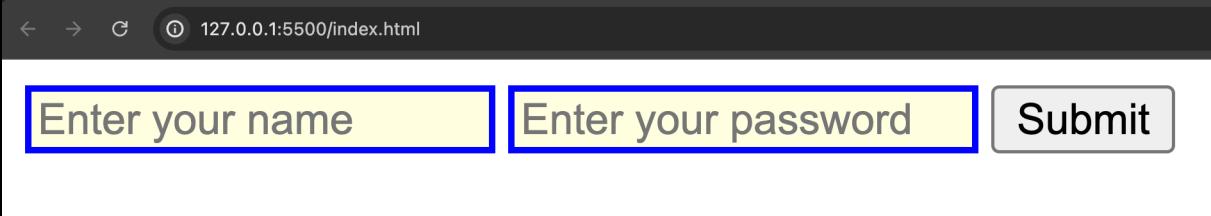


Selectors (Attribute selector)

```
<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>
```

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)

```
<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>
```

Adjacent Sibling Selector

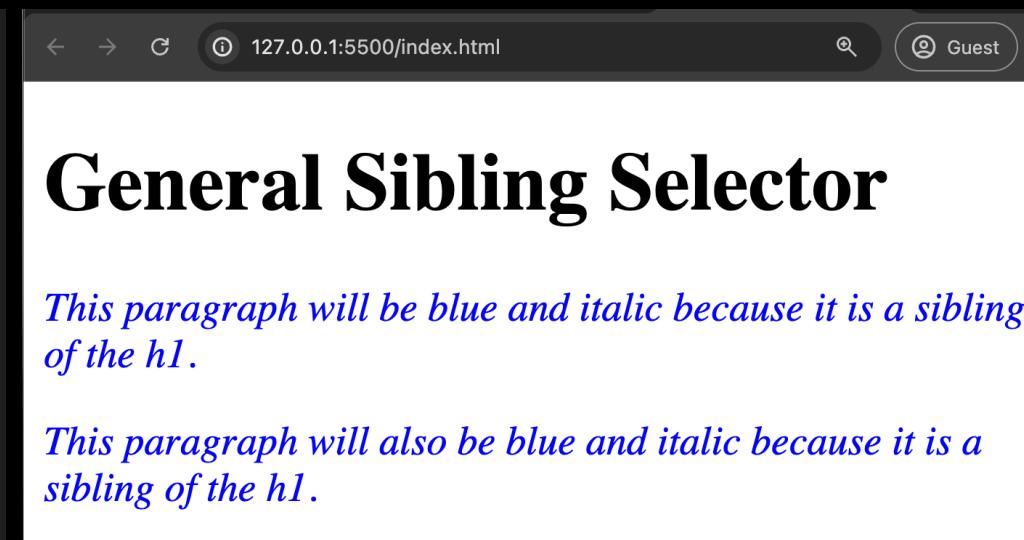
This paragraph will be green, bold, and larger because it is immediately after the h1.

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>
```



- 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>
```

The screenshot shows a browser window at the URL 127.0.0.1:5500/index.html. The page content includes:

- Anchor Pseudo-Classes:** A list of three links:
 - [Hover over me!](#)
 - [Click me!](#)
- Focus Pseudo-Class:** A button labeled "Click to focus".
- Child Pseudo-Class:** An ordered list:
 - First item
 - Second item

Practice Set

Advance Selectors

```
1 <div class="store">
2   <header>
3     <h1>Welcome to MyStore</h1>
4     <nav>
5       <ul class="main-nav">
6         <li>Home</li>
7         <li>About</li>
8         <li>
9           Products
10          <ul>
11            <li>Electronics</li>
12            <li>Books</li>
13          </ul>
14        </li>
15        <li>Contact</li>
16      </ul>
17    </nav>
18  </header>
19  <section>
20    <h2>Featured Products</h2>
21    <div class="product" data-category="electronics">
22      
23      <p>Description of the product goes here.</p>
24      <button type="submit">Buy Now</button>
25    </div>
26    <div class="product" data-category="books">
27      
28      <p>Description of another product.</p>
29      <button type="submit">Buy Now</button>
30    </div>
31  </section>
32  <footer>
33    <p>Copyright 2023 by MyStore. All rights reserved.</p>
34    <p>Follow us on <a href="#" class="social">Social Media</a></p>
35  </footer>
36 </div>
```

Practice Set

Advance Selectors

1. Attribute Selector:

- Style all `div.product` elements with a `border: 1px solid gray;`.
- Highlight products where the `data-category` attribute contains “electronics” with a `background-color: lightblue;`.

2. Child Selector:

- Style direct children `li` elements of `ul.main-nav` with `color: darkblue;`.
- Increase the left margin for the nested `ul` inside `li.Products` to `20px`.

3. Adjacent Sibling Selector:

- Style the paragraph immediately following the `h2` tag in the section with `font-weight: bold;`.

4. General Sibling Selector:

- Apply a `font-size: 12px;` to all paragraphs that are siblings of an `img` element within `div.product`.

5. Pseudo-Classes:

- Style the button when hovered over with `background-color: green; color: white;`.
- Style the first `li` in `ul.main-nav` to be `font-weight: bold;`.

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;
  }

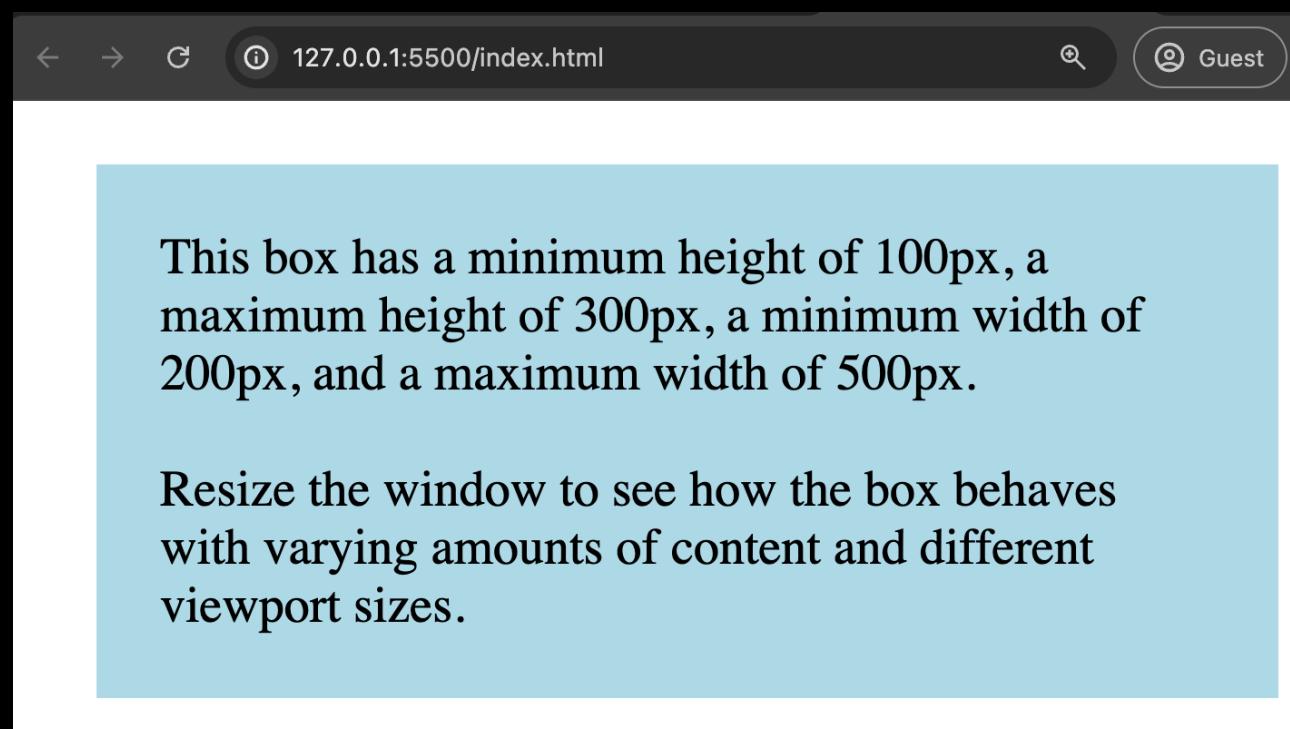
  .auto {
    overflow: auto;
    background-color: lightyellow;
  }
</style>
```

The screenshot shows a browser window with four colored boxes demonstrating different overflow behaviors:

- Red Box (top-left):** Shows a div with `overflow: visible;`. The content overflows the box, with a message: "This is a div with overflow: visible. Content can overflow out of this box." The content includes: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla at turpis eros. In hac habitasse platea dictumst."
- Green Box (top-right):** Shows a div with `overflow: hidden;`. The content is clipped, with a message: "This is a div with overflow: hidden. Content is clipped and will not be visible outside the box." The content includes: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla at turpis eros. In hac habitasse platea dictumst."
- Blue Box (bottom-left):** Shows a div with `overflow: scroll;`. The content is clipped and scrollbars are present, with a message: "This is a div with overflow: scroll. Scrollbars are added to see overflowing content." The content includes: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla at turpis eros. In hac habitasse platea dictumst."
- Yellow Box (bottom-right):** Shows a div with `overflow: auto;`. The content is clipped and scrollbars appear only if necessary, with a message: "This is a div with overflow: auto. Scrollbars appear only if necessary." The content includes: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla at turpis eros. In hac habitasse platea dictumst."

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>
```



- **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.

Box-Shadow Property

```
<head>
  <title>Box-Shadow Example</title>
  <style>
    .box {
      width: 200px;
      height: 150px;
      background-color: #3498db;
      color: white;
      display: flex;
      align-items: center;
      justify-content: center;
      border-radius: 10px;
      box-shadow: 10px 10px 15px rgba(0, 0, 0, 0.3);
    }
  </style>
</head>
<body>
  <div class="box">
    This box has a shadow!
  </div>
</body>
```



The **box-shadow** property in CSS adds shadow effects around an element's frame.

Syntax:

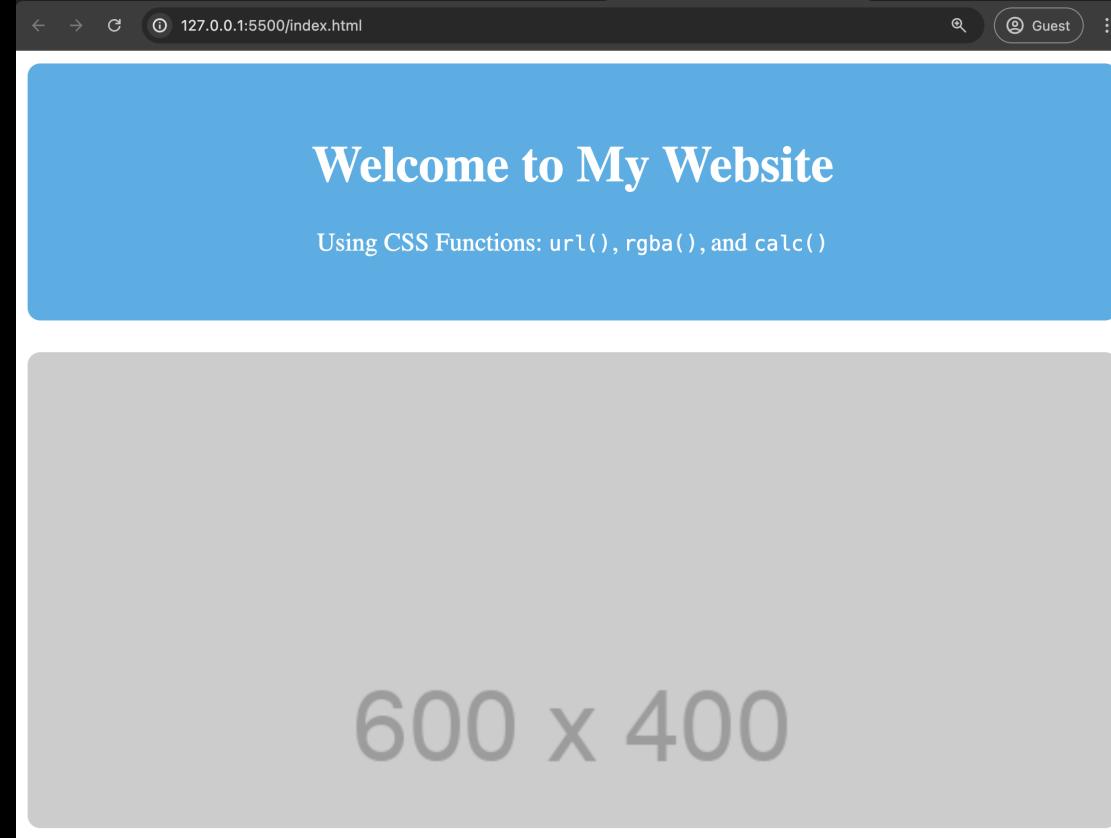
- **box-shadow: h-offset v-offset blur spread color;**
- **h-offset:** Horizontal offset of the shadow.
- **v-offset:** Vertical offset of the shadow.
- **blur:** Optional. Blur radius of the shadow.
- **spread:** Optional. Spread radius of the shadow.
- **color:** Color of the shadow.

CSS Functions

```
<head>
  <title>CSS Functions Example</title>
  <style>
    .header {
      background-color: rgba(52, 152, 219, 0.8); /* rgba function */
      padding: calc(10px + 2vw); /* calc function */
      color: white;
      text-align: center;
      border-radius: 8px;
    }

    .content {
      background-image: url('https://via.placeholder.com/600x400'); /* url function */
      background-size: cover;
      height: 300px;
      margin-top: 20px;
      border-radius: 8px;
    }
  </style>
</head>
<body>
  <div class="header">
    <h1>Welcome to My Website</h1>
    <p>Using CSS Functions: <code>url()</code>, <code>rgba()</code>, and <code>calc()</code></p>
  </div>

  <div class="content">
  </div>
</body>
```



- CSS functions are **special constructs** that perform specific **operations or calculations** and return values that **can be used** in CSS properties.

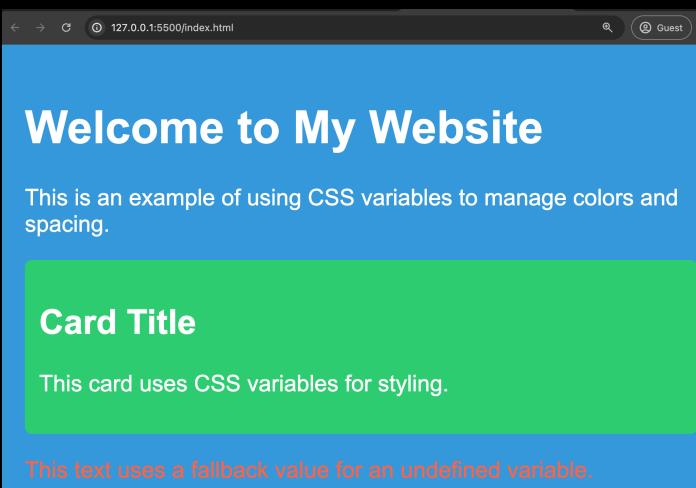
CSS Variables

- CSS variables, also known as custom properties, are entities defined by CSS authors that contain specific values to be reused throughout a document.
- Declared using the -- prefix, e.g., --main-color: #3498db;.
- Accessed using the var() function, e.g., color: var(--main-color);.
- var() function can accept a fallback value if the variable is not defined, e.g., color: var(--secondary-color, #2ecc71);

```
<body>
  <h1>Welcome to My Website</h1>
  <p>This is an example of using CSS
variables to manage colors and spacing.</p>

  <div class="card">
    <h2>Card Title</h2>
    <p>This card uses CSS variables
for styling.</p>
  </div>

  <p class="fallback-example">This text
uses a fallback value for an undefined
variable.</p>
</body>
```



```
<title>CSS Variables Example</title>
<style>
  html {
    --main-bg-color: #3498db;
    --main-text-color: #ffffff;
    --secondary-bg-color: #2ecc71;
    --secondary-text-color: #ffff00;
    --padding: 10px;
    --border-radius: 5px;
  }

  body {
    background-color: var(--main-bg-color);
    color: var(--main-text-color);
    font-family: Arial, sans-serif;
    padding: var(--padding);
  }

  .card {
    background-color: var(--secondary-bg-color);
    color: var(--secondary-text-color);
    padding: var(--padding);
    border-radius: var(--border-radius);
    margin: var(--padding) 0;
  }

  .fallback-example {
    /* Uses fallback color */
    color: var(--undefined-variable, #ff6347);
  }
</style>
```

Practice Set

CSS Advanced

```
1 <body>
2   <header>
3     <h1>My Blog</h1>
4   </header>
5   <section class="content">
6     <article class="post">
7       <h2>First Post</h2>
8       <div class="post-content">
9         <p>
10        | Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do
11        | eiusmod tempor incididunt ut labore et dolore magna aliqua.
12        </p>
13      </div>
14    </article>
15    <article class="post">
16      <h2>Second Post</h2>
17      <div class="post-content">
18        <p>
19        | Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do
20        | eiusmod tempor incididunt ut labore et dolore magna aliqua.
21        </p>
22      </div>
23    </article>
24  </section>
25  <footer>
26    <p>© 2023 My Blog</p>
27  </footer>
28 </body>
```

Practice Set

CSS Advanced

1. Overflow Property:

- Ensure that the .post-content divs have a maximum height of 100px and set the overflow to auto to handle any overflowing content.

2. Min-Max Height and Width:

- Set the header to have a minimum height of 80px and a maximum height of 150px.
- Ensure the section.content has a minimum width of 300px and a maximum width of 800px.

3. Box-Shadow Property:

- Apply a box shadow to the .post elements with the following properties: horizontal offset 5px, vertical offset 5px, blur radius 10px, spread radius 0, and color rgba(0, 0, 0, 0.1).

4. CSS Functions:

- Use the calc function to set the padding of the .post-content divs to calc(10px + 2%).

5. CSS Variables:

- Declare a CSS variable --main-bg-color with the value #f0f0f0.
- Apply this variable to the background color of the body.
- Declare another CSS variable --header-color with the value #333.
- Use this variable to set the color of the h1 in the header.

Media Queries

declaration

The diagram shows a code block for a media query. At the top left are three colored circles (red, yellow, green). A red arrow points from the text "Media Type" to the word "screen" in the query. Another red arrow points from the text "Specifying amount of screen to cover" to the "max-width: 768px" condition. A third red arrow points from the text "styles to apply when all conditions are met" to the block of CSS rules below. The code is as follows:

```
@media screen and (max-width: 768px){  
    .container{  
        // Write styles here  
    }  
}
```

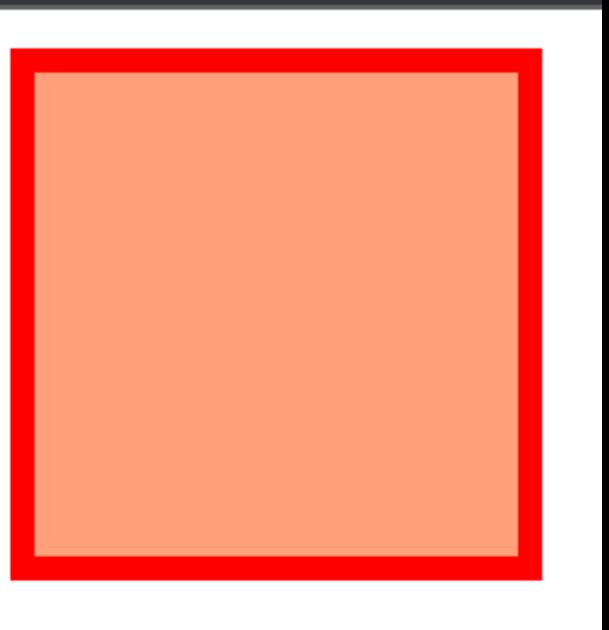
styles to apply
when all conditions
are met

Specifying amount
of screen to cover

- Tailor styles for specific device characteristics.
- Use to create **responsive** web designs.
- Apply styles based on conditions like screen size.
- Syntax: **@media (condition) { CSS rules }**.
- Can **combine** multiple conditions using and, or.

Media Queries (width)

```
.box {  
    height: 100px;  
    width: 100px;  
    background-color: lightsalmon;  
}  
  
@media screen and (width: 250px) {  
    .box {  
        border: 5px solid red;  
    }  
}
```



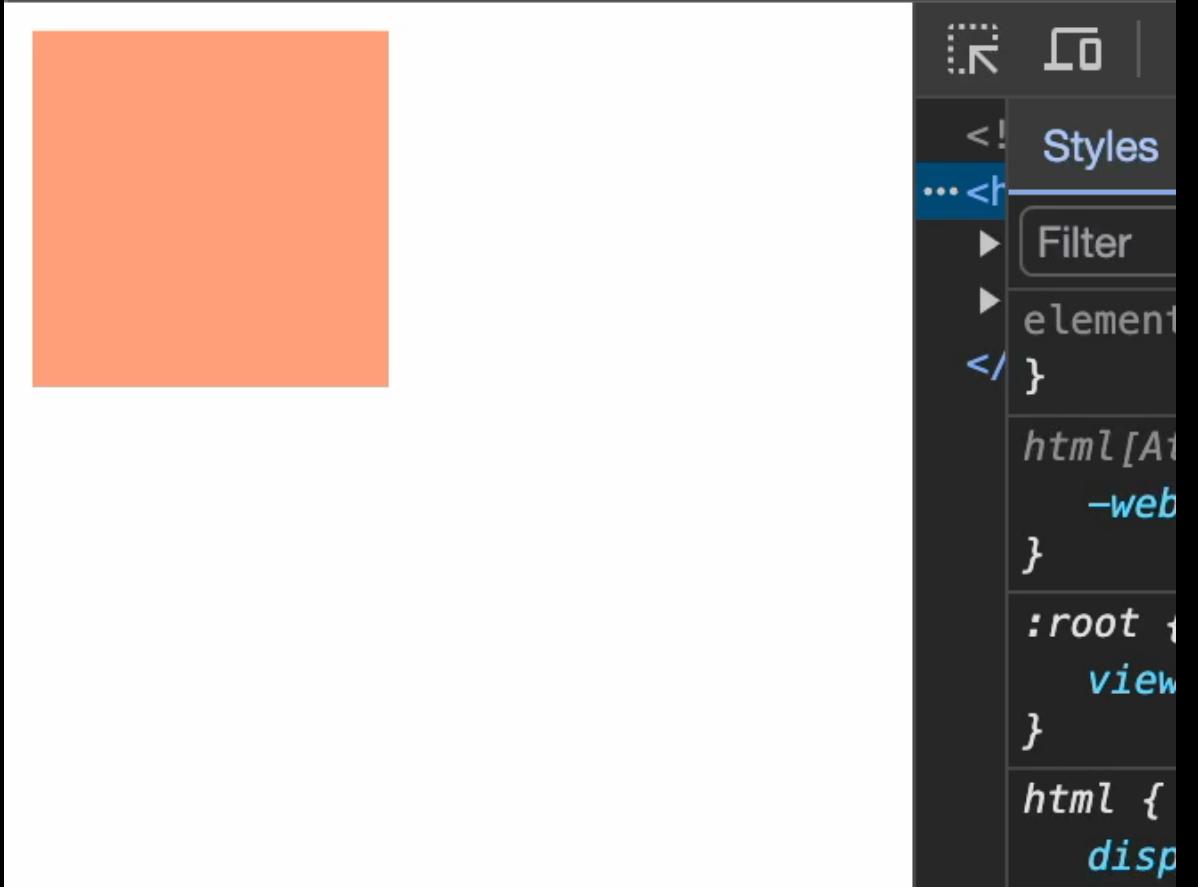
Media Queries (min-width)

```
.box {  
    height: 100px;  
    width: 100px;  
    background-color: lightsalmon;  
}  
  
@media screen and (min-width: 300px) {  
    .box {  
        height: 150px;  
        width: 150px;  
    }  
}
```



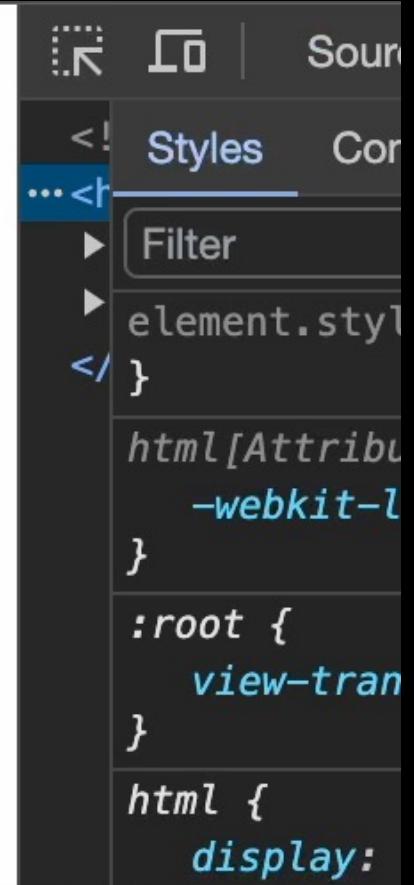
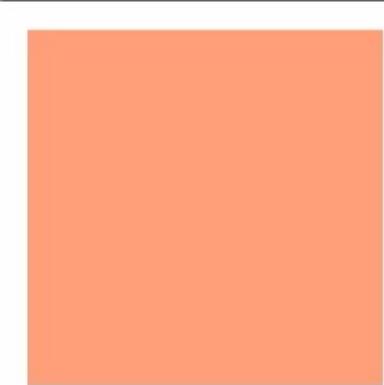
Media Queries (max-width)

```
.box {  
    height: 100px;  
    width: 100px;  
    background-color: #lightsalmon;  
}  
  
@media screen and (max-width: 250px) {  
    .box {  
        height: 50px;  
        width: 50px;  
    }  
}
```



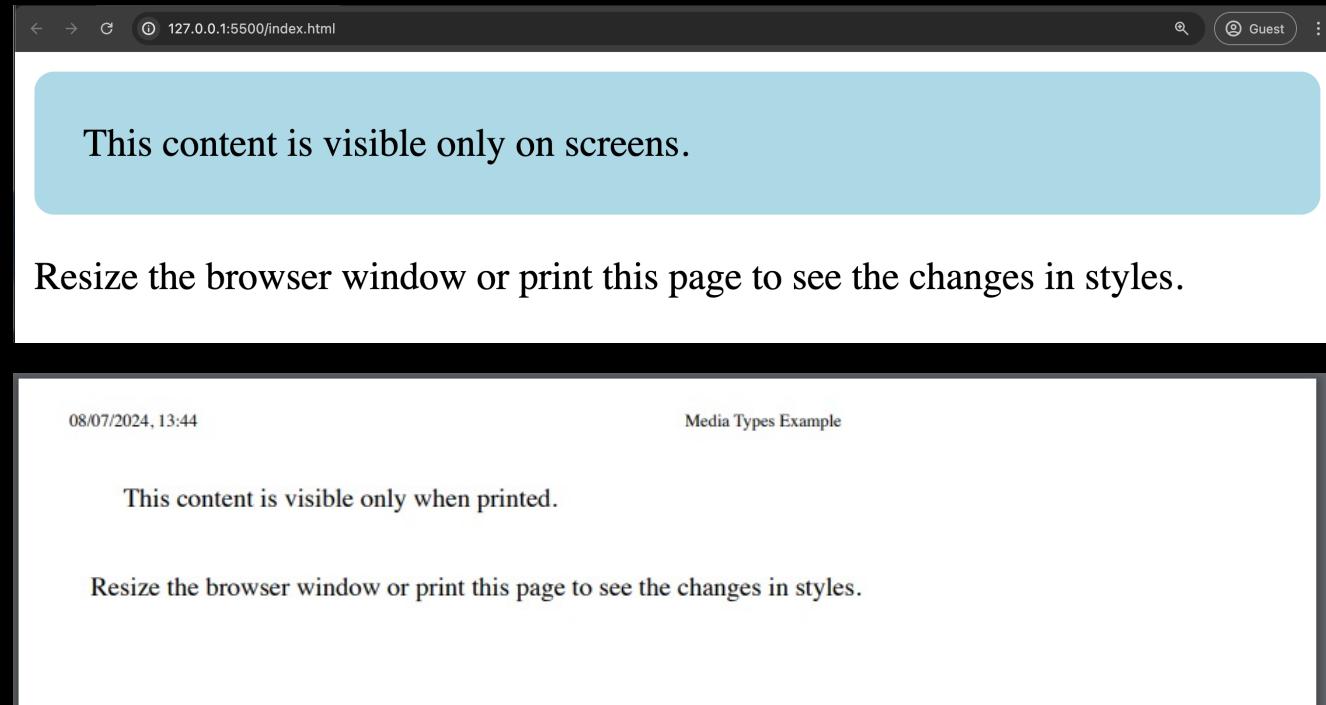
Media Queries (combination)

```
.box {  
    height: 100px;  
    width: 100px;  
    background-color: lightcoral;  
}  
  
@media screen and (min-width: 250px)  
and (max-width: 300px) {  
    .box {  
        border-radius: 50%;  
    }  
}
```

A screenshot of a browser's developer tools showing the 'Styles' tab. It displays the CSS rules applied to the element, including the global styles and the media query rule from the code block above.

Media Queries (different media types)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Media Types Example</title>
    <style>
        .screen-only {
            background-color: #lightblue;
            padding: 20px;
            border-radius: 8px;
        }
        .print-only {
            display: none;
        }
        @media print {
            .screen-only {
                display: none;
            }
            .print-only {
                display: block;
                background-color: #lightgray;
                padding: 20px;
                border-radius: 8px;
            }
        }
    </style>
</head>
<body>
    <div class="screen-only">
        This content is visible only on screens.
    </div>
    <div class="print-only">
        This content is visible only when printed.
    </div>
    <p>Resize the browser window or print this page to see the changes in styles.</p>
</body>
</html>
```



Different Media types are:

- All
- Print
- Screen
- Speech
- Braille, embossed

Media Queries (different condition types)

1. **min-width**: Applies styles when the **viewport width** is greater than or equal.
2. **max-width**: Applies styles when the **viewport width** is less than or equal.
3. **min-height**: Applies styles when the **viewport height** is greater than or equal.
4. **max-height**: Applies styles when the **viewport height** is less than or equal.
5. **orientation**: Applies styles based on the **device's orientation**, either portrait or landscape.
6. **min-resolution**: Applies styles when the **device's resolution** is greater than or equal.
7. **max-resolution**: Applies styles when the **device's resolution** is less than or equal.
8. **min-aspect-ratio**: Applies styles when the **device's aspect ratio** is greater than.
9. **max-aspect-ratio**: Applies styles when the **device's aspect ratio** is less than.
10. **min-device-width**: Applies styles when the **device's width** is greater than.
11. **max-device-width**: Applies styles when the **device's width** is less than or equal.
12. **min-device-height**: Applies styles when the **device's height** is greater than or equal.
13. **max-device-height**: Applies styles when the **device's height** is less than or equal.

Practice Set

Media Queries

- Create a div of 50px by 50px with color green. It should respond to following conditions:
 - Keep color **green** under 300px of view port size
 - Change color to **red** from 300px to 400px
 - Change color to **blue** after 400px



Practice Set Advanced (Media Queries)

```
<header>
  <h1>My Portfolio</h1>
  <nav>
    <ul class="main-nav">
      <li>Home</li>
      <li>About</li>
      <li>Projects</li>
      <li>Contact</li>
    </ul>
  </nav>
</header>
<section class="intro">
  <h2>Welcome to My Portfolio</h2>
  <p>This is a brief introduction about myself.</p>
</section>
<section class="projects">
  <h2>Projects</h2>
  <div class="project">
    <h3>Project One</h3>
    <p>Description of project one.</p>
  </div>
  <div class="project">
    <h3>Project Two</h3>
    <p>Description of project two.</p>
  </div>
</section>
<footer>
  <p>&copy; 2023 My Portfolio</p>
</footer>
```

- 1. Basic Media Query:** Apply a background-color: lightblue; to the body when the screen width is less than 600px.
- 2. Min-Width Media Query:** Style the header with background-color: darkblue; and color: white; when the screen width is at least 600px.
- 3. Max-Width Media Query:** Set the header font-size to 1.2em when the screen width is at most 800px.
- 4. Orientation Media Query:** Apply a background-color: lightgreen; to the .projects section when the device orientation is landscape.
- 5. Min-Height Media Query:** Set the .intro section's padding: 20px; when the viewport height is at least 400px.
- 6. Combination Media Query:** Style the .project divs with a border: 2px solid black; and padding: 10px; when the screen width is between 600px and 1200px.
- 7. Different Media Types:** Apply a font-size: 12pt; to the body when printing.
- 8. Min-Resolution Media Query:** Apply a font-size: 1.5em; to the body when the device resolution is at least 2dppx.