CSC-336
Web Technologies

Lecture 3:

Topics:

- Box Model, DIV element, Cascading Order
- Combining CSS Selectors, CSS Positioning

Muhammad Naveed Shaikh
Department of Computer Engineering
naveedshaikh@cuiatd.edu.pk

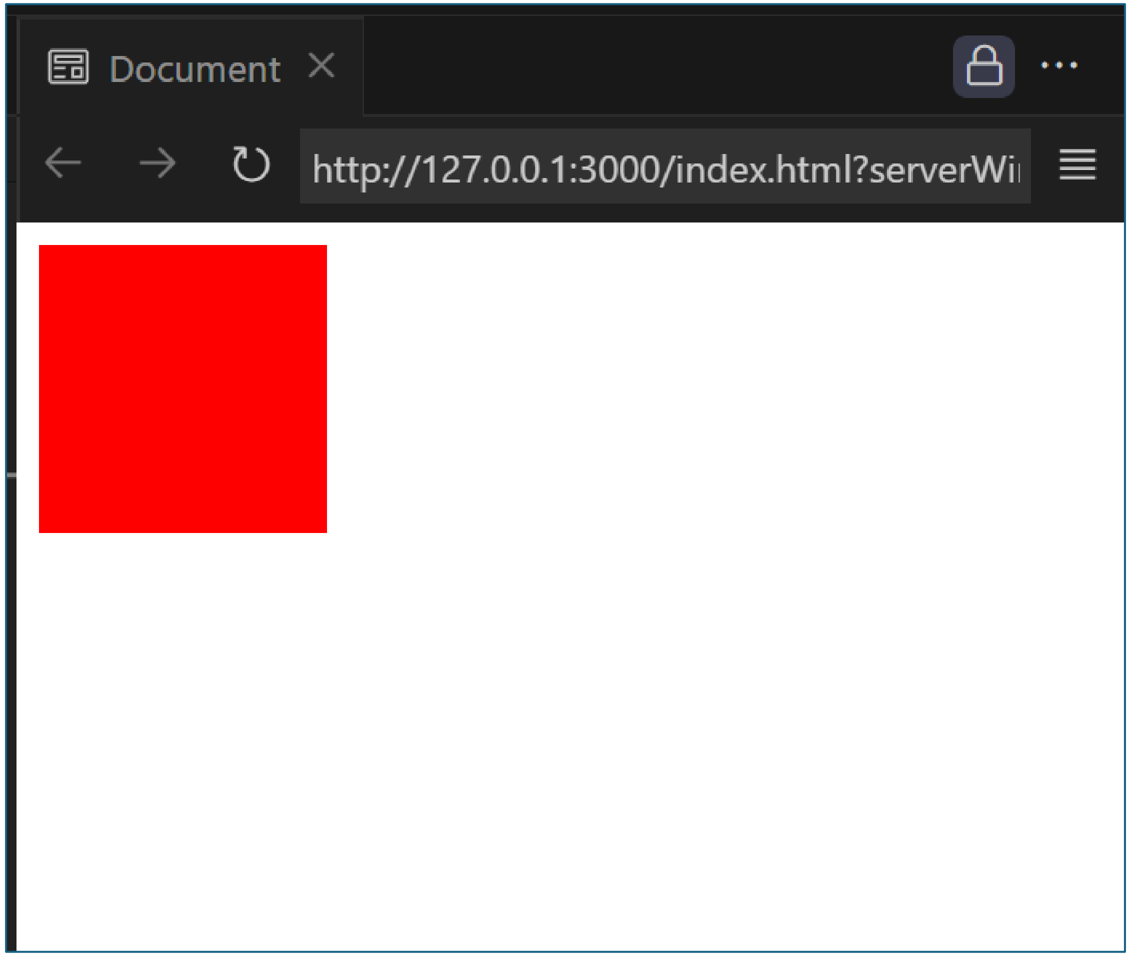
COMSATS University Islamabad | Abbottabad Campus

The Box Model

- Every Element is box
  - I highly recommend to visit any web of your choice
    - Hover over various elements ,
    - you will notice all these are simply boxes in some organized way.
    - It will add further visualization if you get enabled Developer Options.

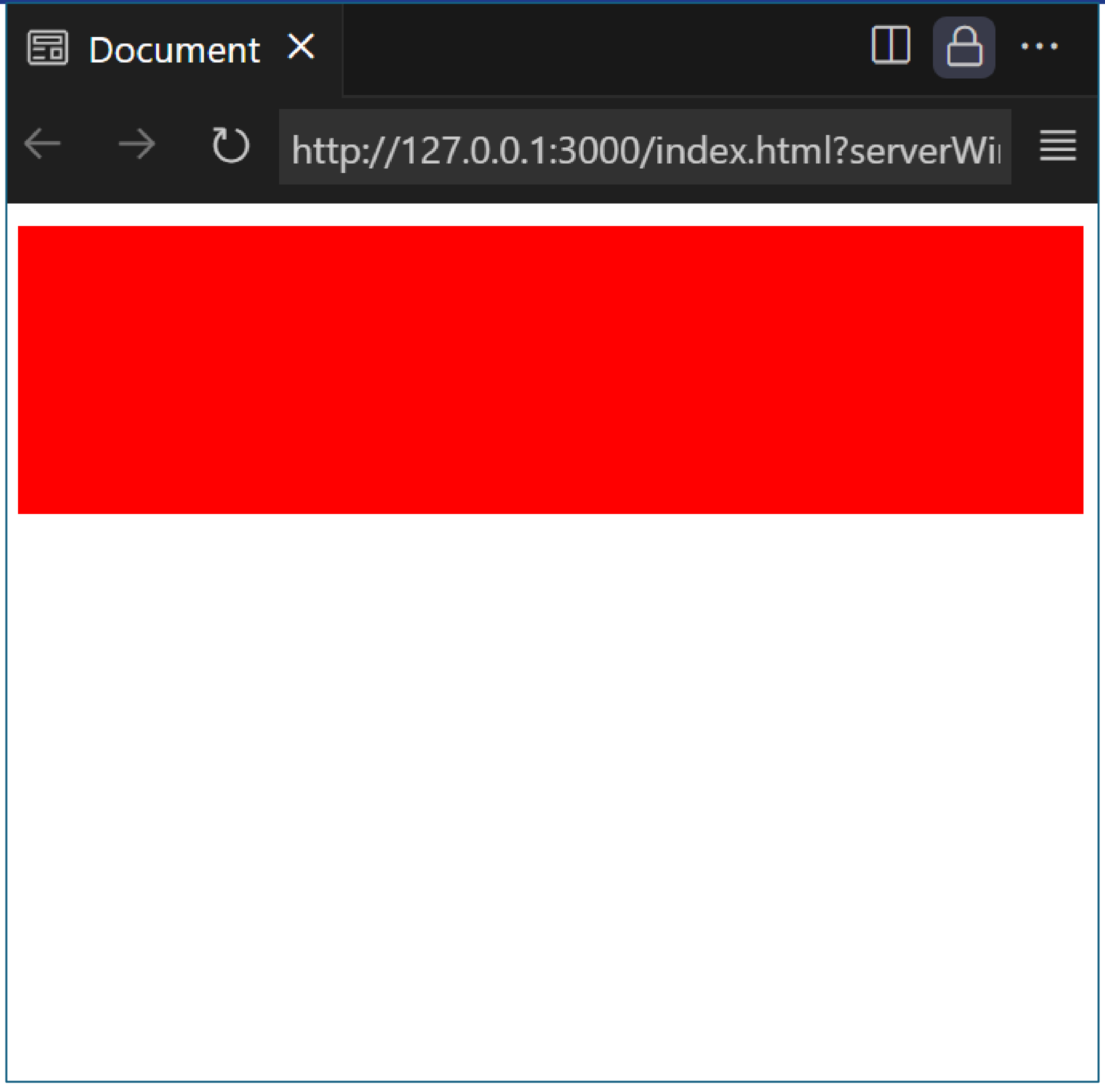
Box Model- DIV Element (width and height attributes)

```
   <style>
      #red{
         background-color: red;
        height: 100px;
        width:  100px}
   </style>
<body>
 <div id="red"></div>
</body>
```



Box Model- DIV Element (width and height attributes)

```
  <style>
     #red{
         background-color: red;
        height: 100px;
           }
  </style>
<body>
 <div id="red"></div>
</body>
```

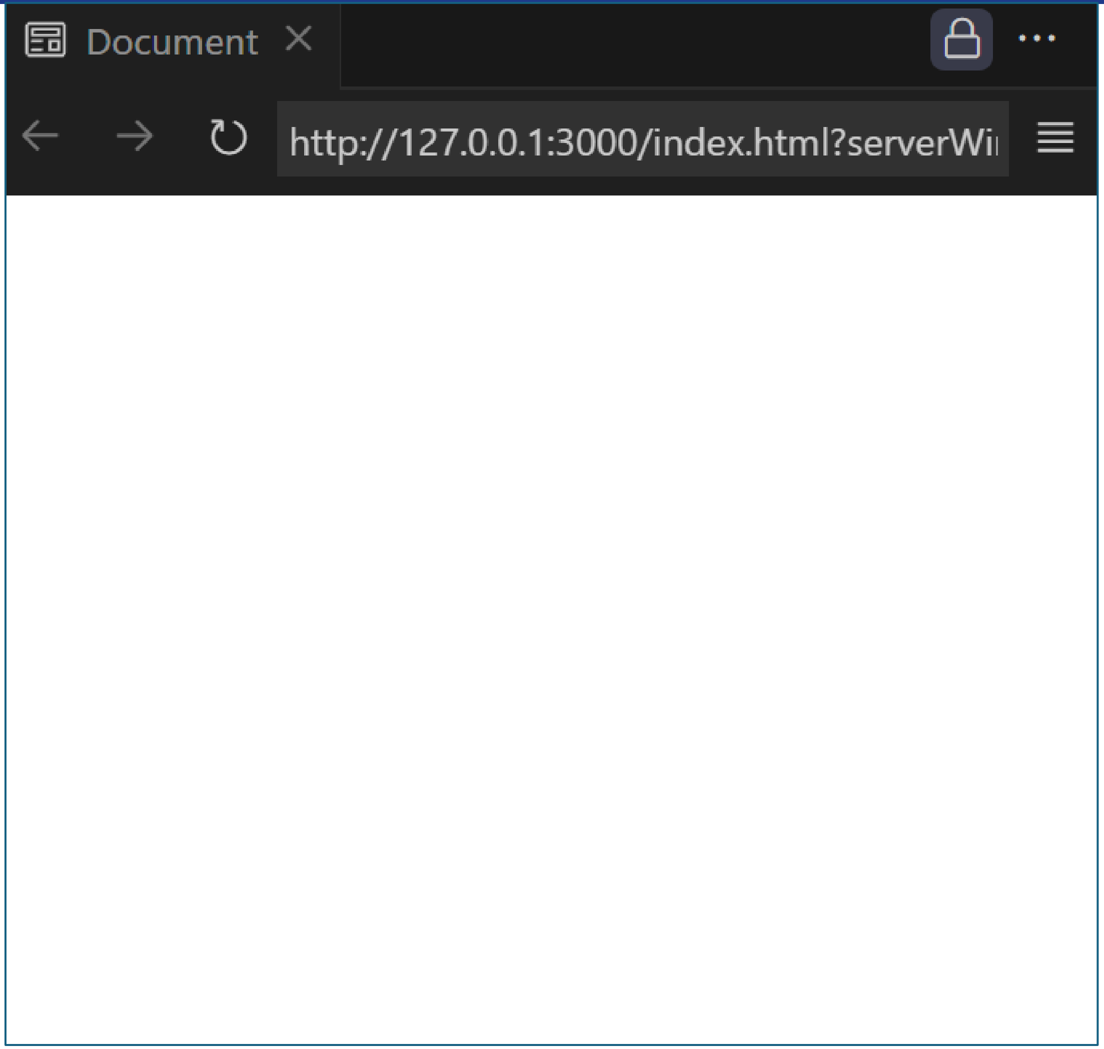


Document

http://127.0.0.1:3000/index.html?serverWi

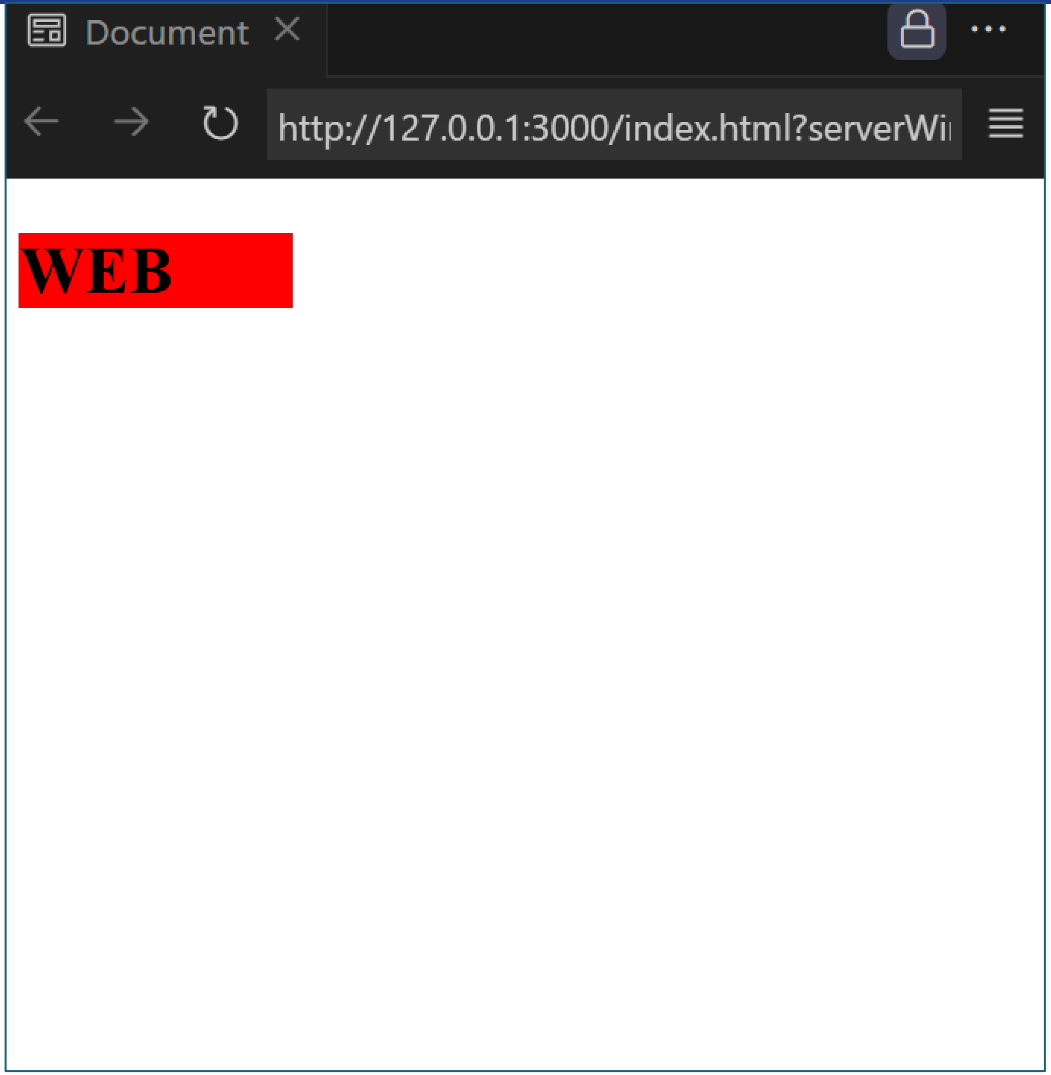
Box Model- DIV Element (width and height attributes)

```
   <style>
     #red{
         background-color: red;
        width: 100px;
           }
   </style>
<body>
 <div id="red"></div>
</body>
```



Box Model- DIV Element (width and height attributes)

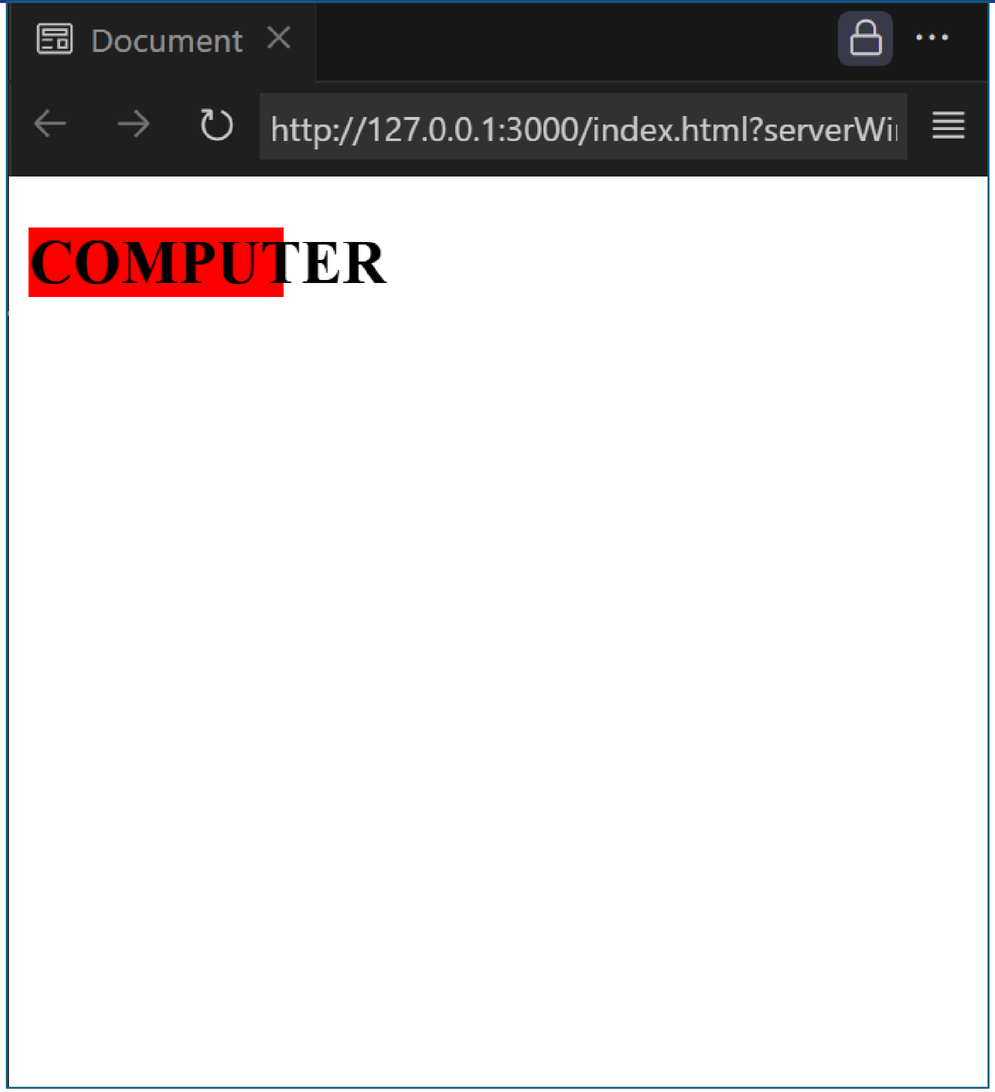
```
   <style>
      #red{
         background-color: red;
         width:100px;
         }
   </style>
<body>
 <div id="red"> <h2>WEB</h2></div>
</body>
```



WEB

Box Model- DIV Element (width and height attributes)

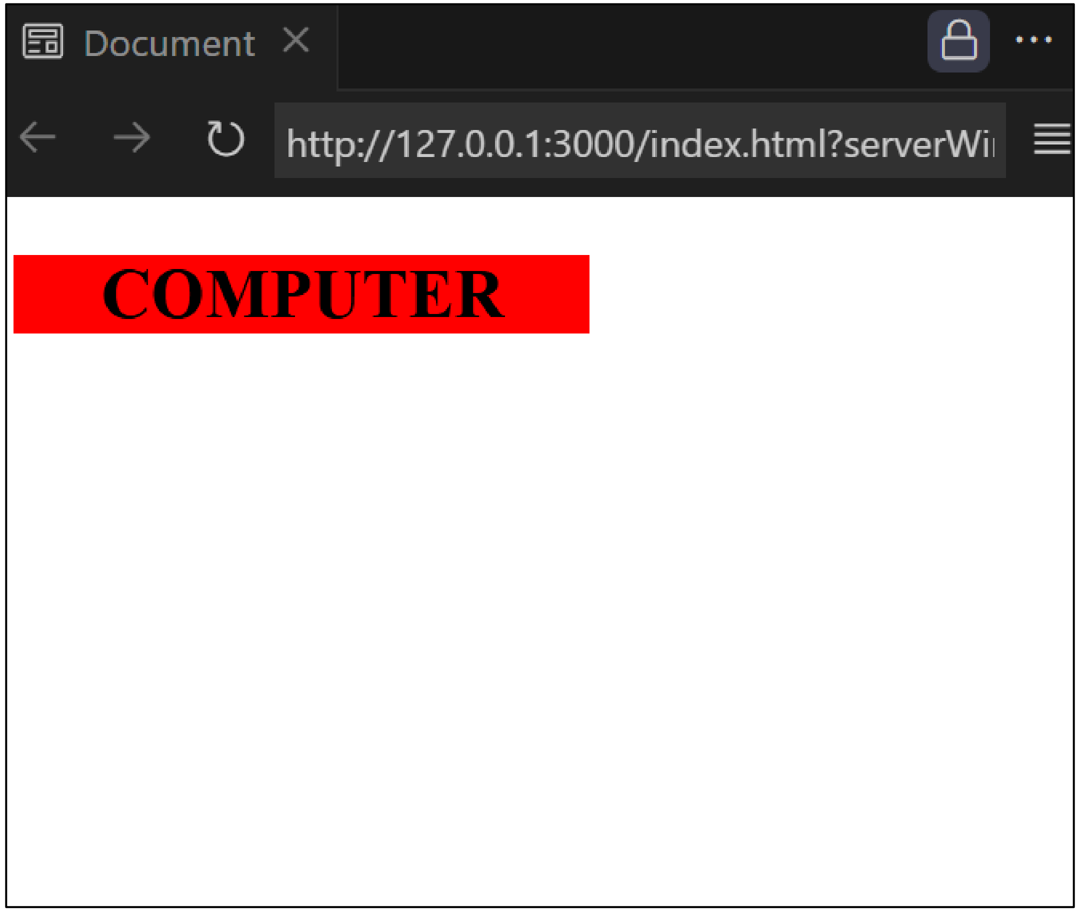
```
 <style>
      #red{
         background-color: red;
         width:100px;
         }
   </style>
<body>
  <div id="red">
          <h2>COMPUTER</h2>
  </div>
</body>
```



COMPUTER

Box Model- DIV Element (width and height attributes)

```
 <style>
      #red{
         background-color: red;
         width: 200px;
         text-align: center;
         }
   </style>
<body>
 <div id="red">
<h2>COMPUTER</h2></div>
</body>
```



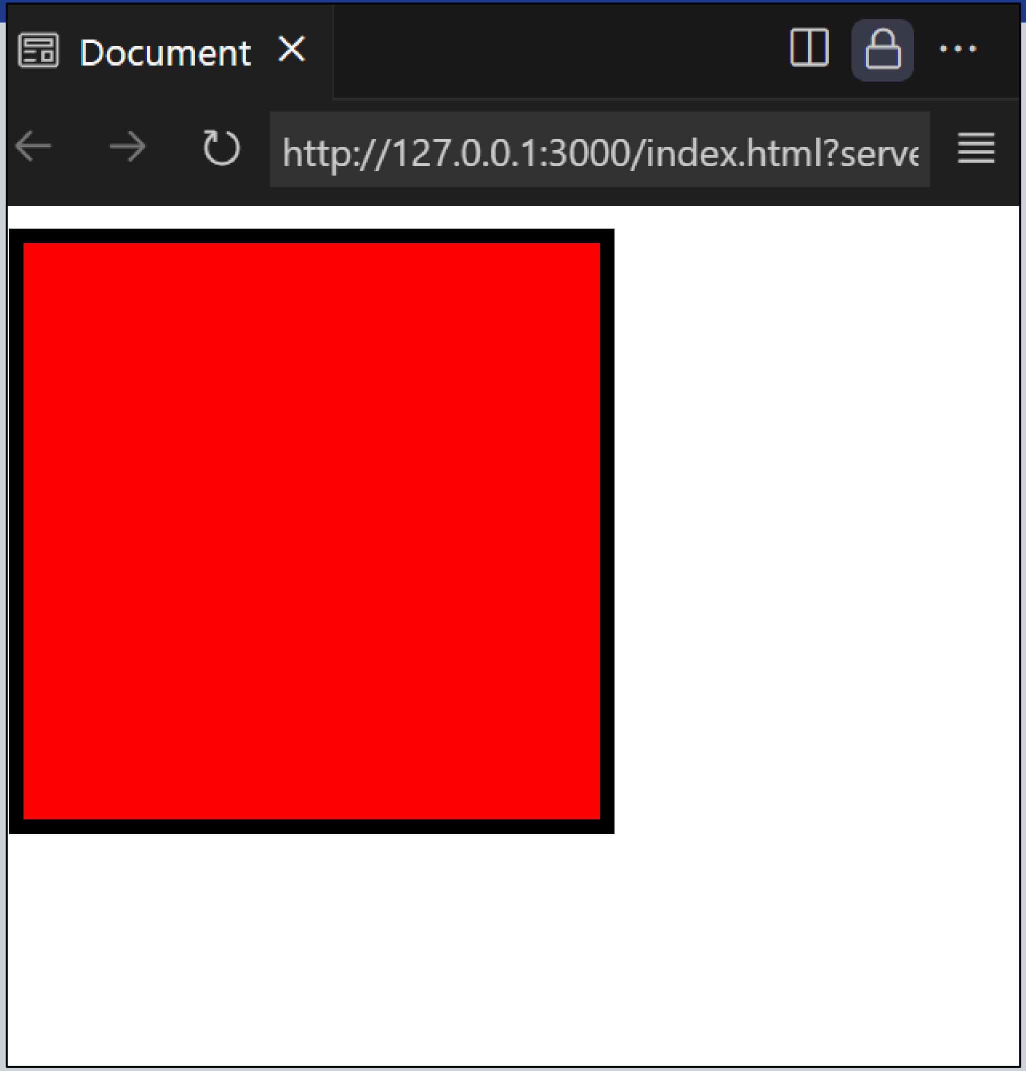
COMPUTER

BOX MODEL - BORDER

```
background-color: red;
width: 200px;
height: 200px;
border: 5px black solid;
```

Note:

- Width and height is same
- Border is an extra space around box



BOX MODEL - BORDER

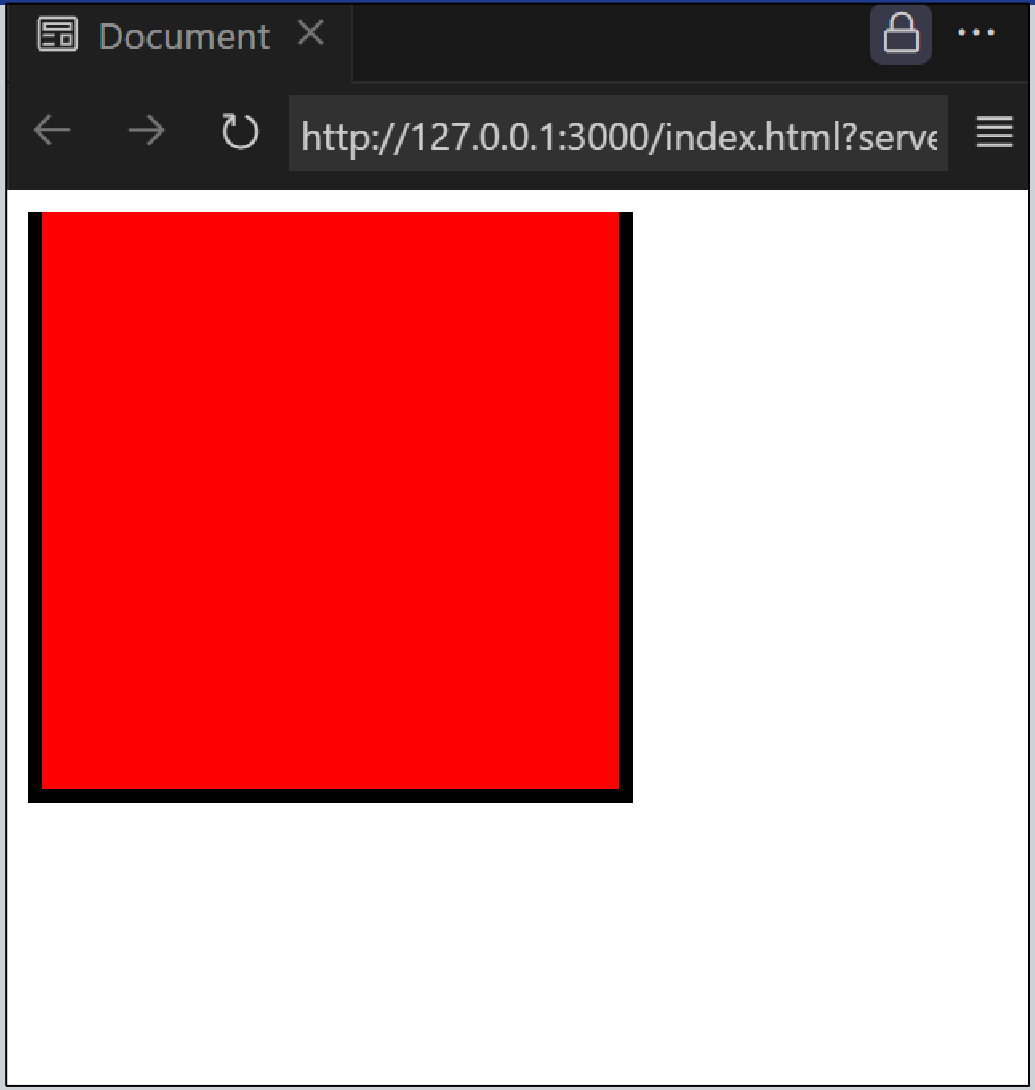
background-color: red;

width: 200px;

height: 200px;

border: 5px black solid;

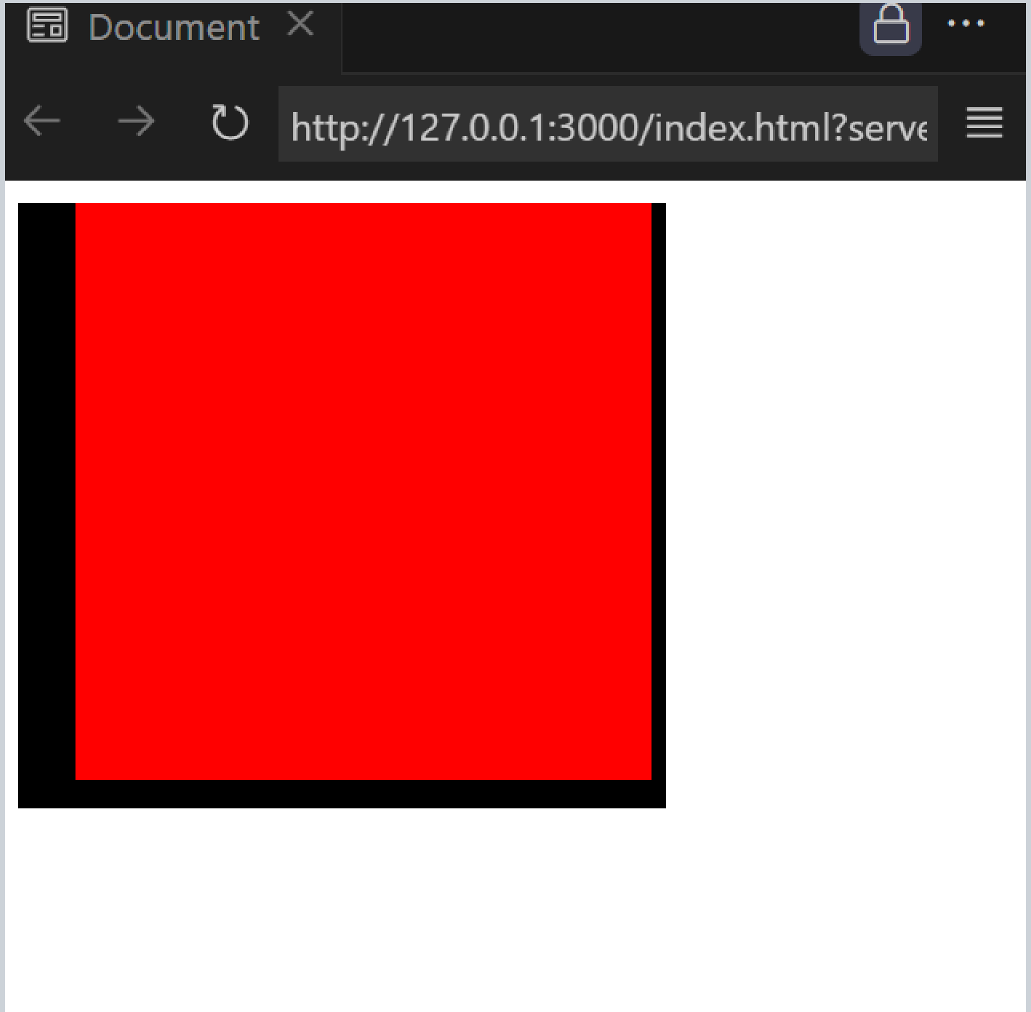
border-top: 0px



BOX MODEL - BORDER

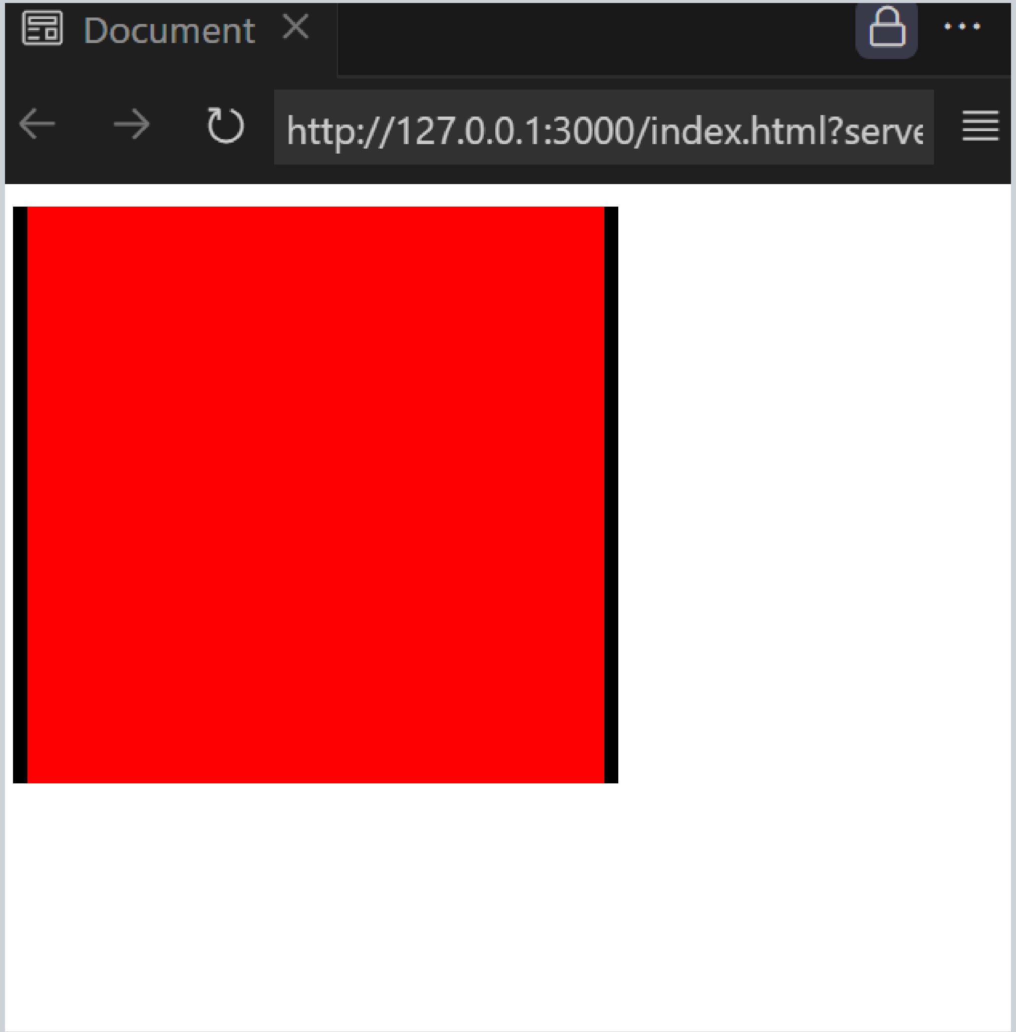
```
background-color: red;
width: 200px;
height: 200px;
border: 5px black solid;
border-width: 0px 5px 10px 20px;
```

Note: clock-wise (top right bottom left)



BOX MODEL - BORDER

```
background-color: red;
width: 200px;
height: 200px;
border: 5px black solid;
border-width: 0px 5px ;
```

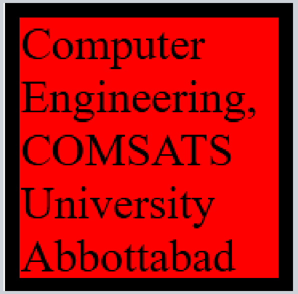


Note:

- first value→ top and bottom
- Second value→right and left

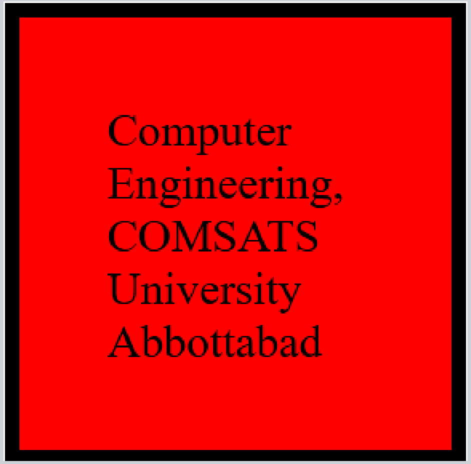
BOX MODEL - Padding

```
background-color: red;
width: 90px;
height: 90px;
border: 5px black solid;
```



Computer Engineering, COMSATS University Abbottabad

```
background-color: red;
width: 90px;
height: 90px;
border: 5px black solid;
padding:30px;
```

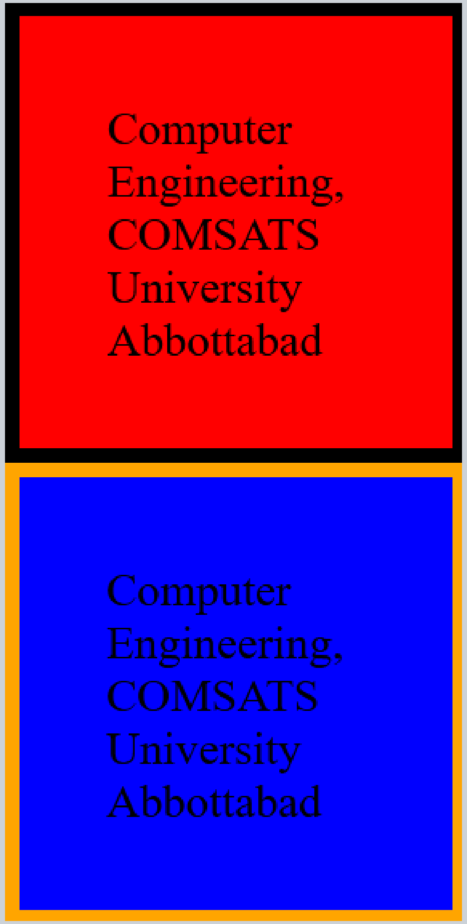


Computer Engineering, COMSATS University Abbottabad

BOX MODEL - Margin

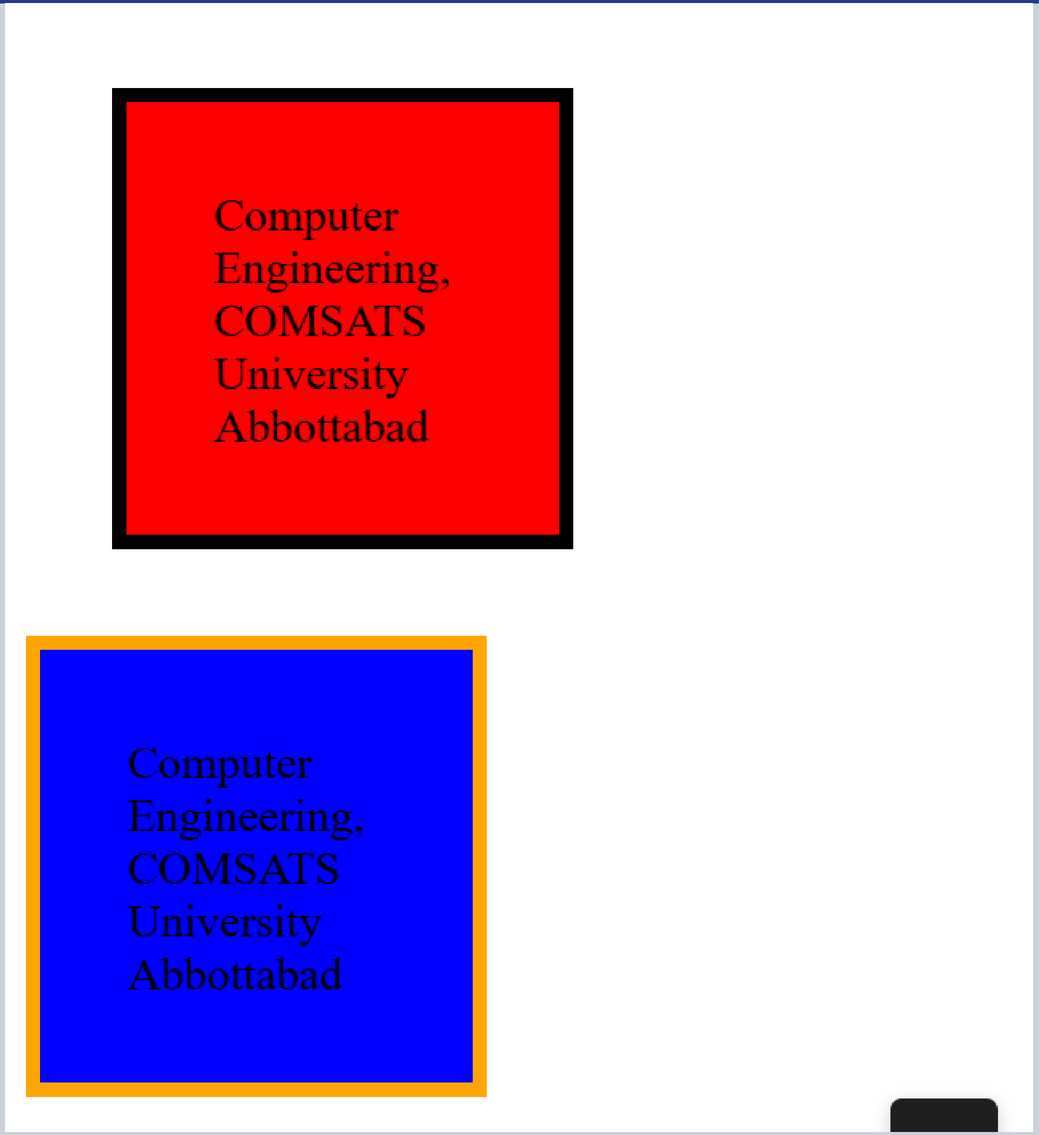
```
background-color: red;
width: 90px;
height: 90px;
border: 5px black solid;
padding:30px;
```

```
background-color: red;
width: 90px;
height: 90px;
border: 5px black solid;
padding:30px;
margin: 30px;
```



Computer Engineering, COMSATS University Abbottabad

Computer Engineering, COMSATS University Abbottabad

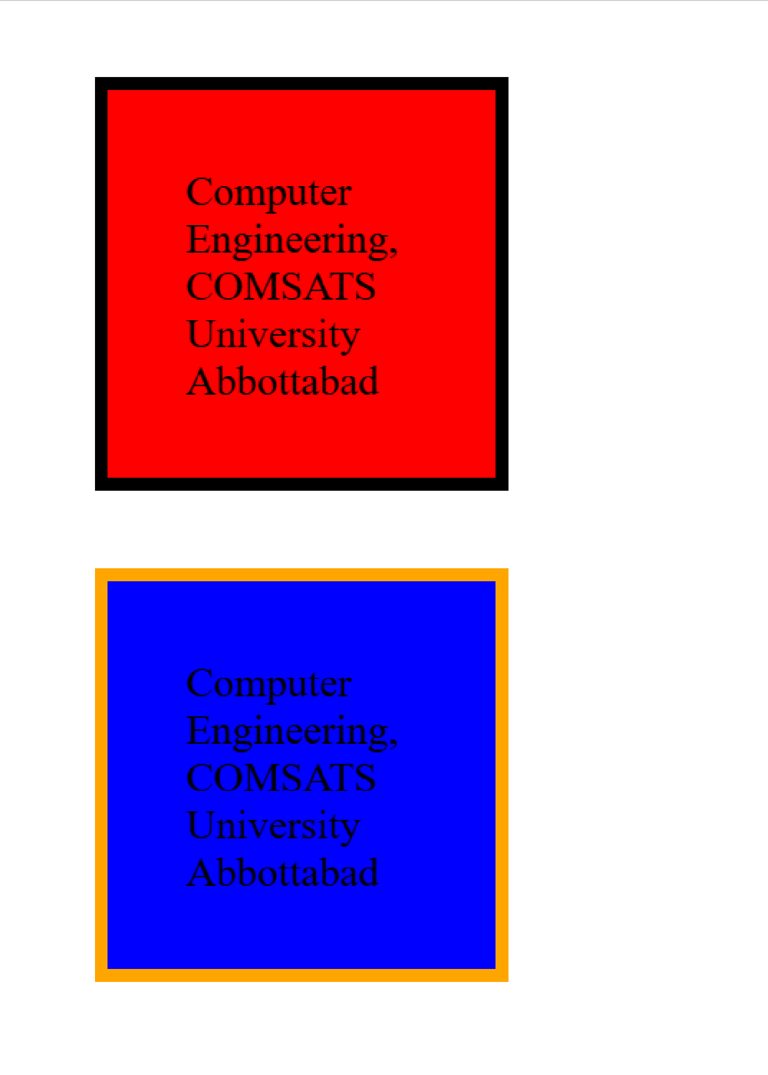


Computer Engineering, COMSATS University Abbottabad

Computer Engineering, COMSATS University Abbottabad

BOX MODEL - Margin

```
background-color: red;

width: 90px;

height: 90px;

border: 5px black solid;

padding:30px;

margin: 30px;
```



Computer Engineering, COMSATS University Abbottabad

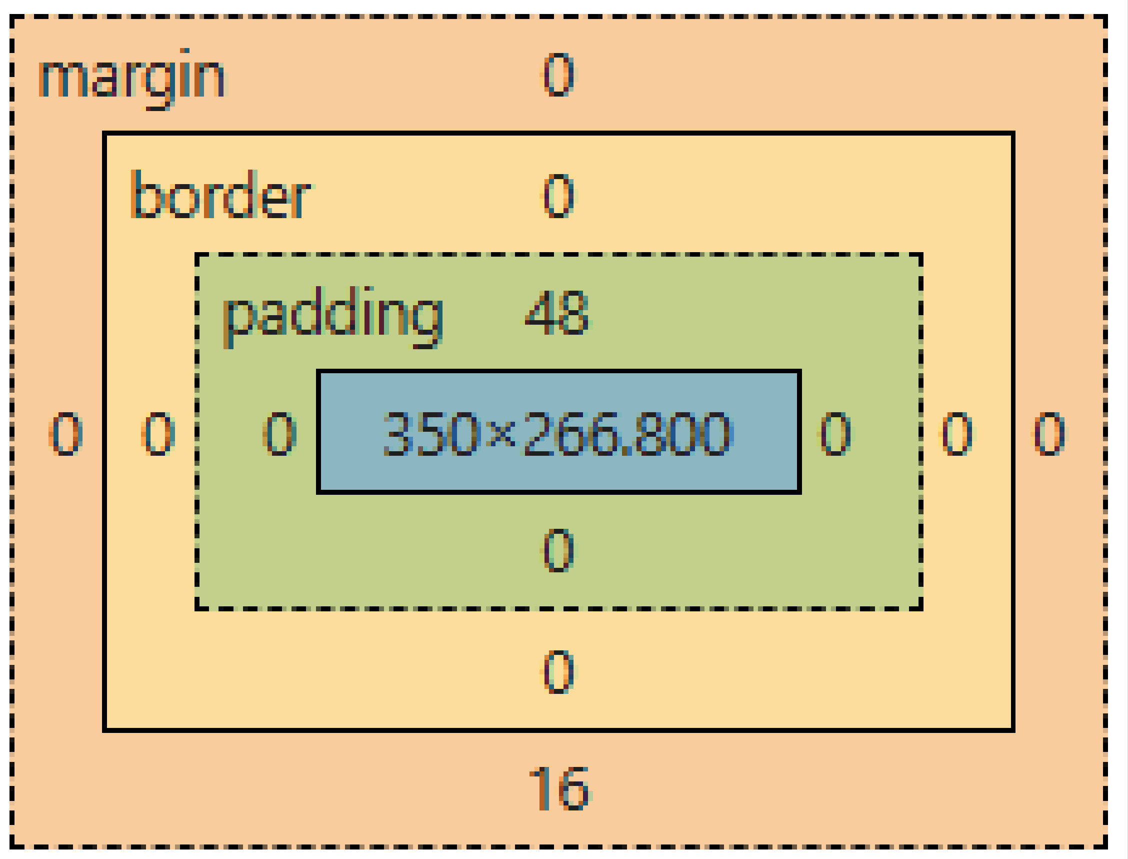
Computer Engineering, COMSATS University Abbottabad

BOX MODEL - Margin

- CONTENT
- Padding: space around the content
- Border : space around the padding
- Margin: Space around the border

https://appbrewery.github.io/box-model/

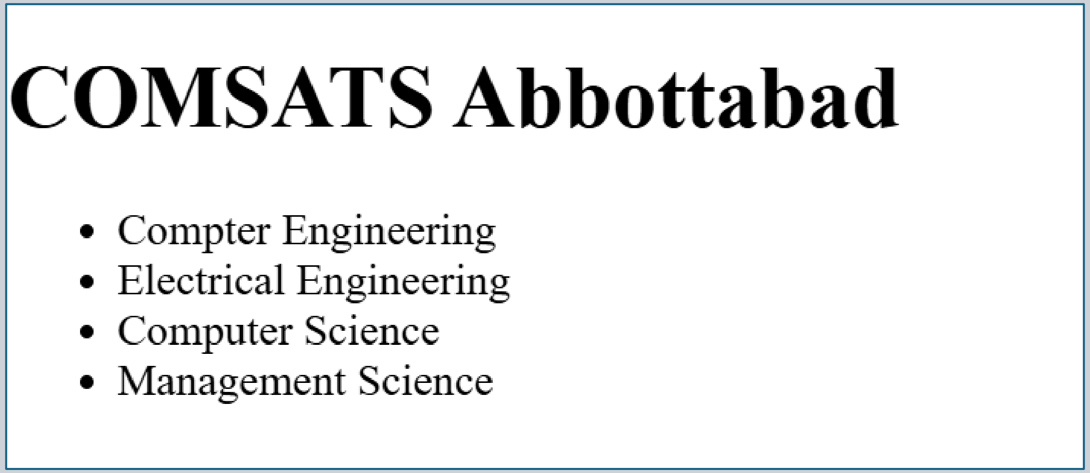
Credit: Angela Yu & App Brewery (Udemy Course)



CASCADDING –Position

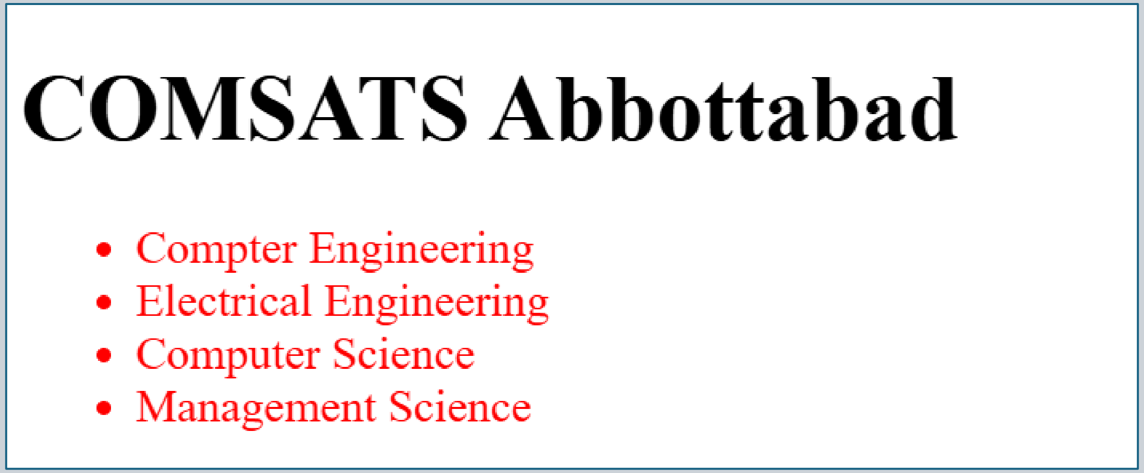
```
<h1>COMSATS Abbottabad</h1>
  <div>
   <ul>
    <li>Compter Engineering</li>
    <li>Electrical Engineering</li>
    <li>Computer Science</li>
    <li>Management Science</li>
   </ul>
  </div>
```

```
<style>
    li{color:blue;}
    li{color:red;}
  </style>
```



COMSATS Abbottabad

- Compter Engineering
- Electrical Engineering
- Computer Science
- Management Science

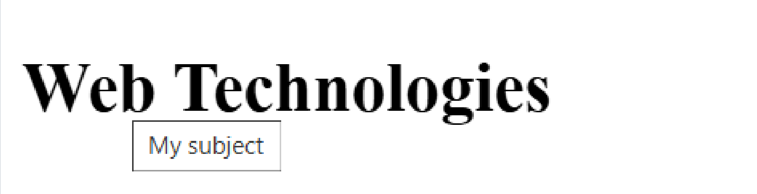


COMSATS Abbottabad

- Compter Engineering
- Electrical Engineering
- Computer Science
- Management Science

Cascading- Specificity

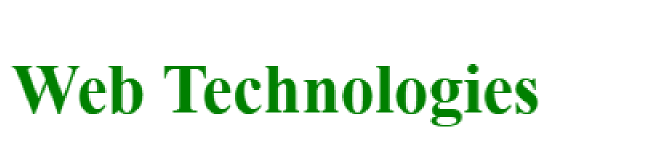
```
<h2 class="blue" id="red" title="My subject">Web Technologies</h2>
```



Web Technologies

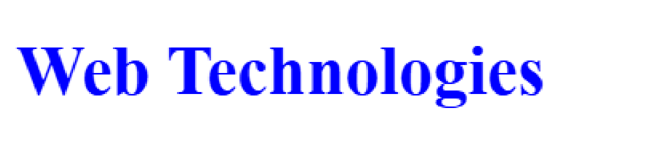
My subject

```
h2{color:green;}
```



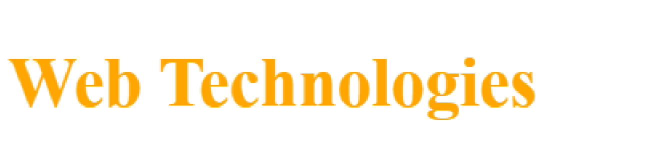
Web Technologies

```
h2{color:green;}
.blue{color:blue;}
```



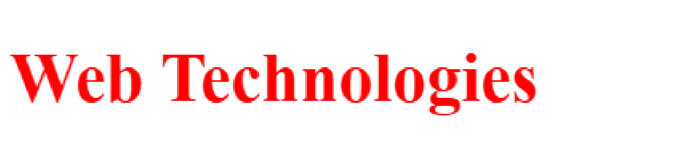
Web Technologies

```
h2{color:green;}
.blue{color:blue;}
 h2[title]{color:orange;}
```



Web Technologies

```
h2{color:green;}
.blue{color:blue;}
 h2[title]{color:orange;}
 #red{color:red;}
```



Web Technologies

Cascading- Specificity -How Specificity Works?

- Each selector gets **points** based on its **type**.
- The **higher the total**, the **stronger the rule**.
- If two rules have **equal specificity**, **the last one in the CSS file wins**.

Selector Type	Example	Specificity Score	Priority
Inline Styles	<h1 style="color:red;">	**1000**	🔺 **Highest**
ID Selector	#title {}	**100**	Very High
Class / Pseudo-Class	.heading {} or :hover	**10**	Medium
Element / Tag	p {} or h1 {}	**1**	Low
Universal Selector	* {}	**0**	Lowest

Cascading- Type

- Three Types

External

```
<link href=“style.css”/>
```

Internal

```
<style> css</style>
```

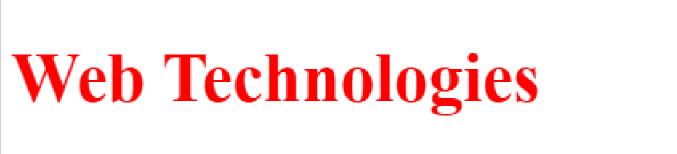
Inline

```
<tag style=“css ”/>
```

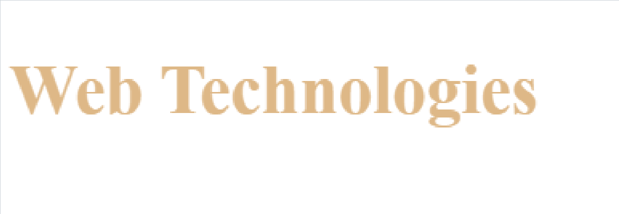
Cascading- Importance

```
<h2 style="color:indigo" class="blue" id="red" title="My subject">Web Technologies</h2>
```

```
h2{color:green;}
.blue{color:blue;}
 h2[title]{color:orange;}
 #red{color:red;}
```


Web Technologies

```
h2{color:green;
      color:burlywood !important;
    }
.blue{color:blue;}
 h2[title]{color:orange;}
 #red{color:red;}
```

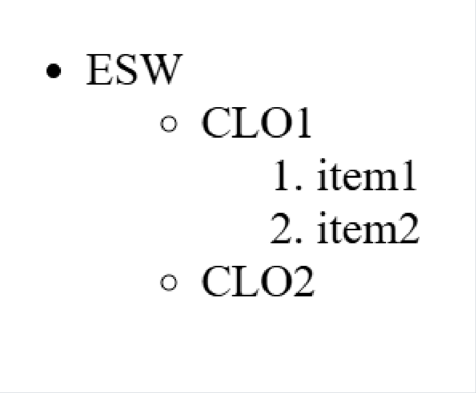

Web Technologies

Cascading- Order

Selector Type	Priority
Importance	Highest
Type	High
Specificity	Medium
Position	Lowest

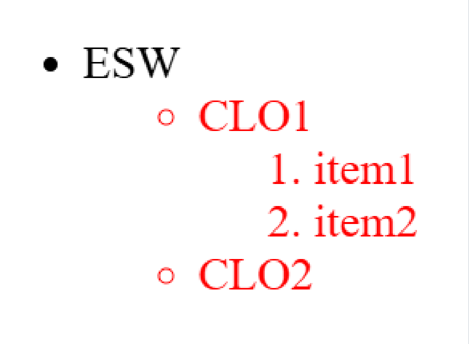
Combining Selectors (Child)

```
<ul>
    <li class="embedded">ESW
      <ul>
          <li>CLO1
              <ol>
                  <li>item1</li>
                  <li>item2</li>
              </ol>
          </li>
          <li>CLO2</li>
      </ul>
    </li>
</ul>
```



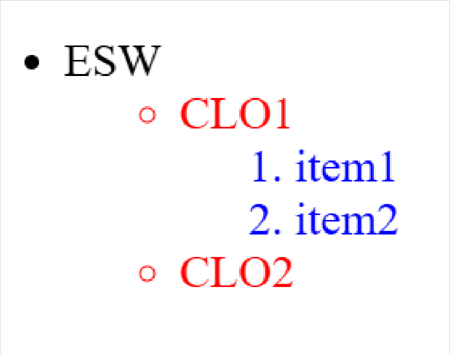
- ESW
  - CLO1
    1. item1
    2. item2
  - CLO2

```
li >ul {color:red }
```



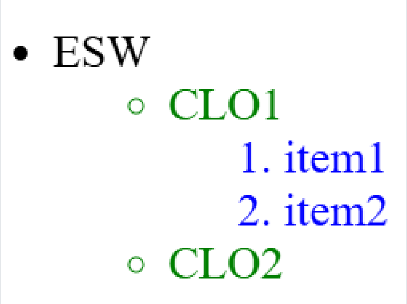
- ESW
  - CLO1
    1. item1
    2. item2
  - CLO2

```
li >ul {color:red }
li >ol {color:blue}
```



- ESW
  - CLO1
    1. item1
    2. item2
  - CLO2

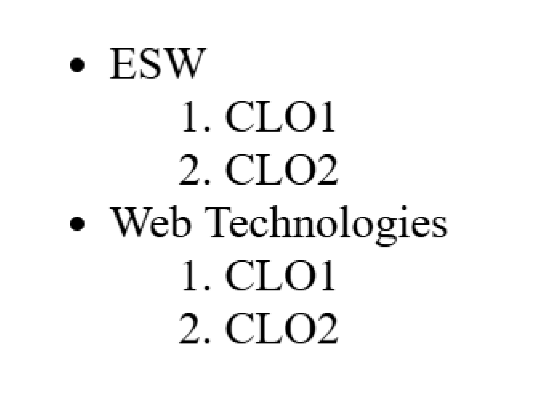
```
li >ul {color:red }
li >ol {color:blue}
li >ul >li {color:green;}
```



- ESW
  - CLO1
    1. item1
    2. item2
  - CLO2

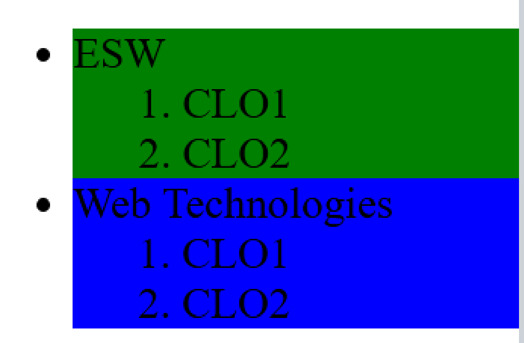
Combining Selectors (Descendent)

```
<ul>
    <li class="embedded">ESW
      <ol>
        <li>CLO1</li>
        <li>CLO2</li>
      </ol>
    </li>
    <li class="web">Web Technologies
      <ol>
        <li>CLO1</li>
        <li>CLO2</li>
      </ol>
    </li>
  </ul>
```



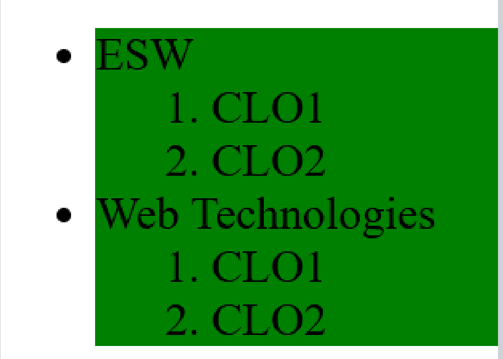
- ESW
  1. CLO1
  2. CLO2
- Web Technologies
  1. CLO1
  2. CLO2

```
.embedded{background-color: green;}
   .web{background-color:blue;}
```



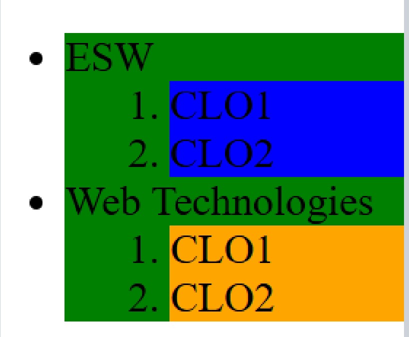
- ESW
  1. CLO1
  2. CLO2
- Web Technologies
  1. CLO1
  2. CLO2

```
li{background-color: green;}
```



- ESW
  1. CLO1
  2. CLO2
- Web Technologies
  1. CLO1
  2. CLO2

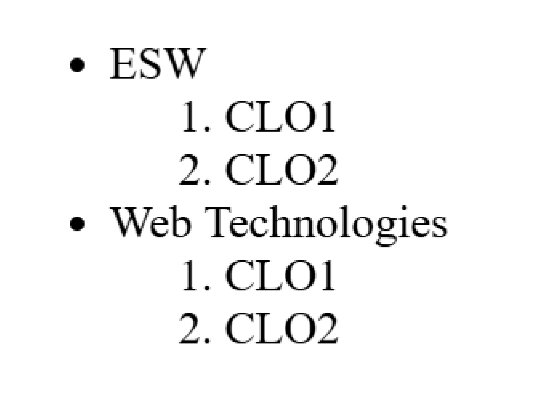
```
li{background-color: green;}
    .embedded li{background-color: blue;}
    .web li{background-color:orange;}
```



- ESW
  1. CLO1
  2. CLO2
- Web Technologies
  1. CLO1
  2. CLO2

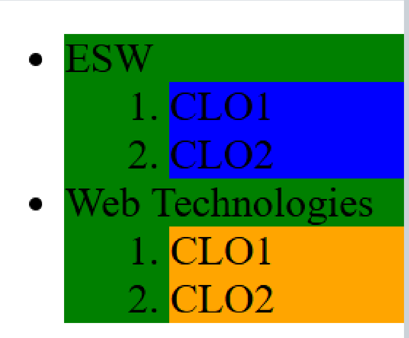
Combining Selectors (Descendent)

```html
<ul>
    <li class="embedded">ESW
      <ol>
        <li>CLO1</li>
        <li>CLO2</li>
      </ol>
    </li>
    <li class="web">Web Technologies
      <ol>
        <li>CLO1</li>
        <li>CLO2</li>
      </ol>
    </li>
  </ul>
```



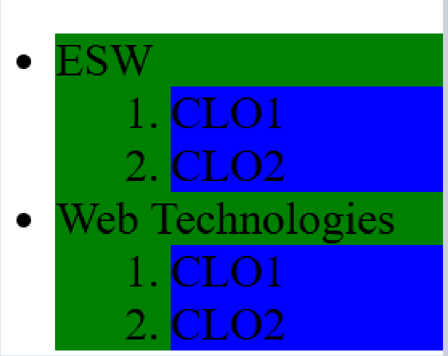
- ESW
  1. CLO1
  2. CLO2
- Web Technologies
  1. CLO1
  2. CLO2

```css
li{background-color: green;}
    .embedded li{background-color: blue;}
    .web li{background-color:orange;}
```



- ESW
  1. CLO1
  2. CLO2
- Web Technologies
  1. CLO1
  2. CLO2

```css
li{background-color: green;}
li li{background-color: blue;}
```



- ESW
  1. CLO1
  2. CLO2
- Web Technologies
  1. CLO1
  2. CLO2

Combining Selectors (Group)

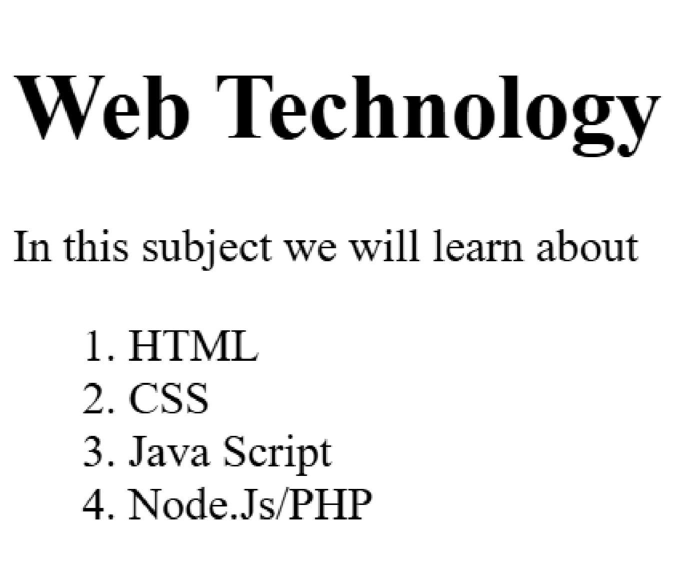
Syntax:

```
Selector , Selector { CSS Rule; }
```

```
h1,p { color:blue}
 h1,ol {background-color:yellowgreen;}
```

- CSS rules applied to both selectors
- There can be more than two selectors

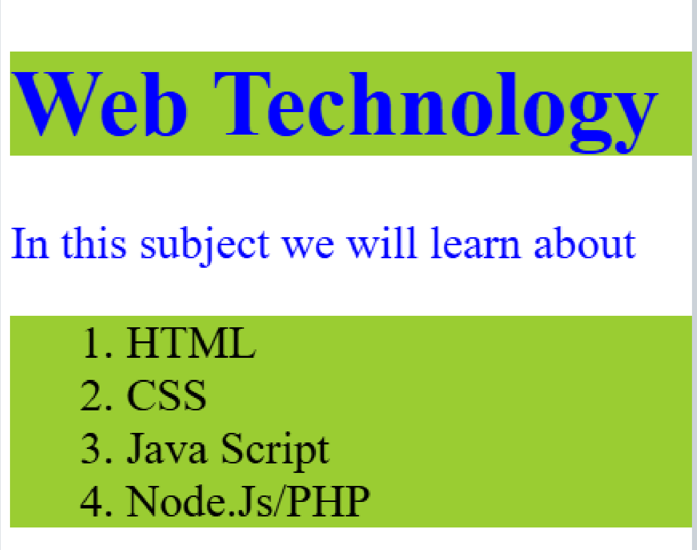
```
 <h1>Web Technology</h1>
 <p>In this subject we will learn
about</p>
  <ol>
     <li>HTML</li>
     <li>CSS</li>
     <li>Java Script</li>
     <li>Node.Js/PHP</li>
  </ol>
```



Web Technology

In this subject we will learn about

1. HTML
2. CSS
3. Java Script
4. Node.Js/PHP



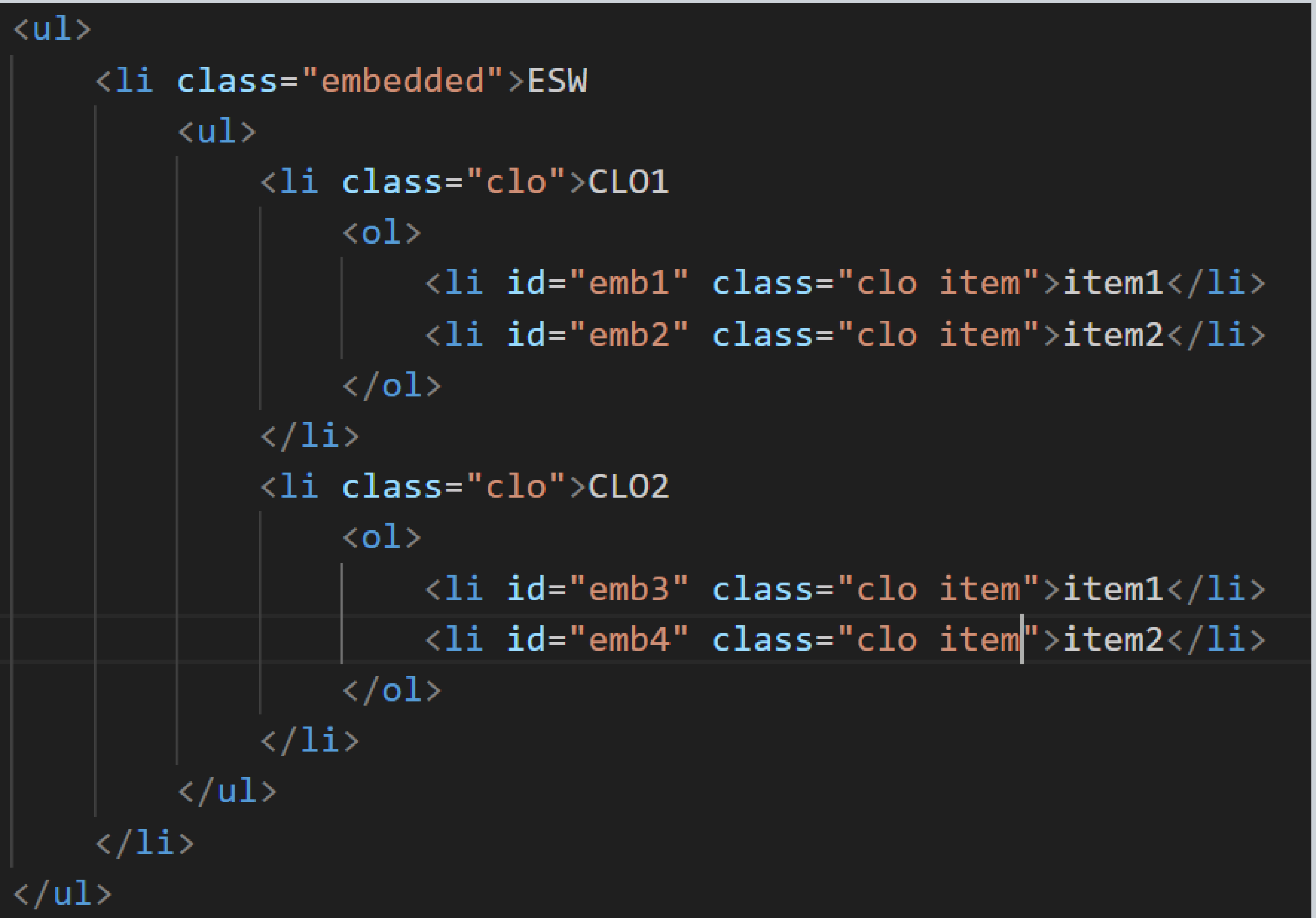
Web Technology

In this subject we will learn about

1. HTML
2. CSS
3. Java Script
4. Node.Js/PHP

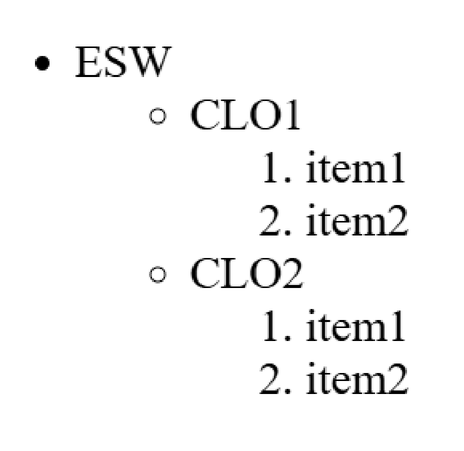
Combining Selectors (Chaining)

Syntax: SelectorSelector { CSS Rule; }

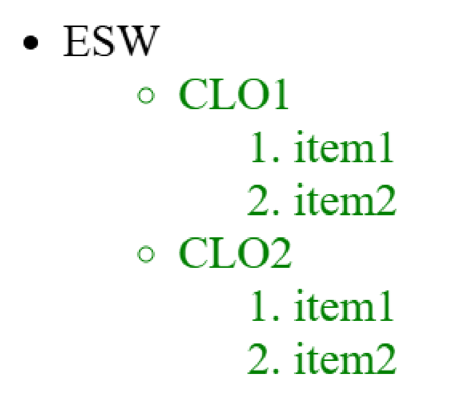


```
<ul>
    <li class="embedded">ESW
        <ul>
            <li class="clo">CLO1
                <ol>
                    <li id="emb1" class="clo item">item1</li>
                    <li id="emb2" class="clo item">item2</li>
                </ol>
            </li>
            <li class="clo">CLO2
                <ol>
                    <li id="emb3" class="clo item">item1</li>
                    <li id="emb4" class="clo item">item2</li>
                </ol>
            </li>
        </ul>
    </li>
</ul>
```

li.clo{color:green;}



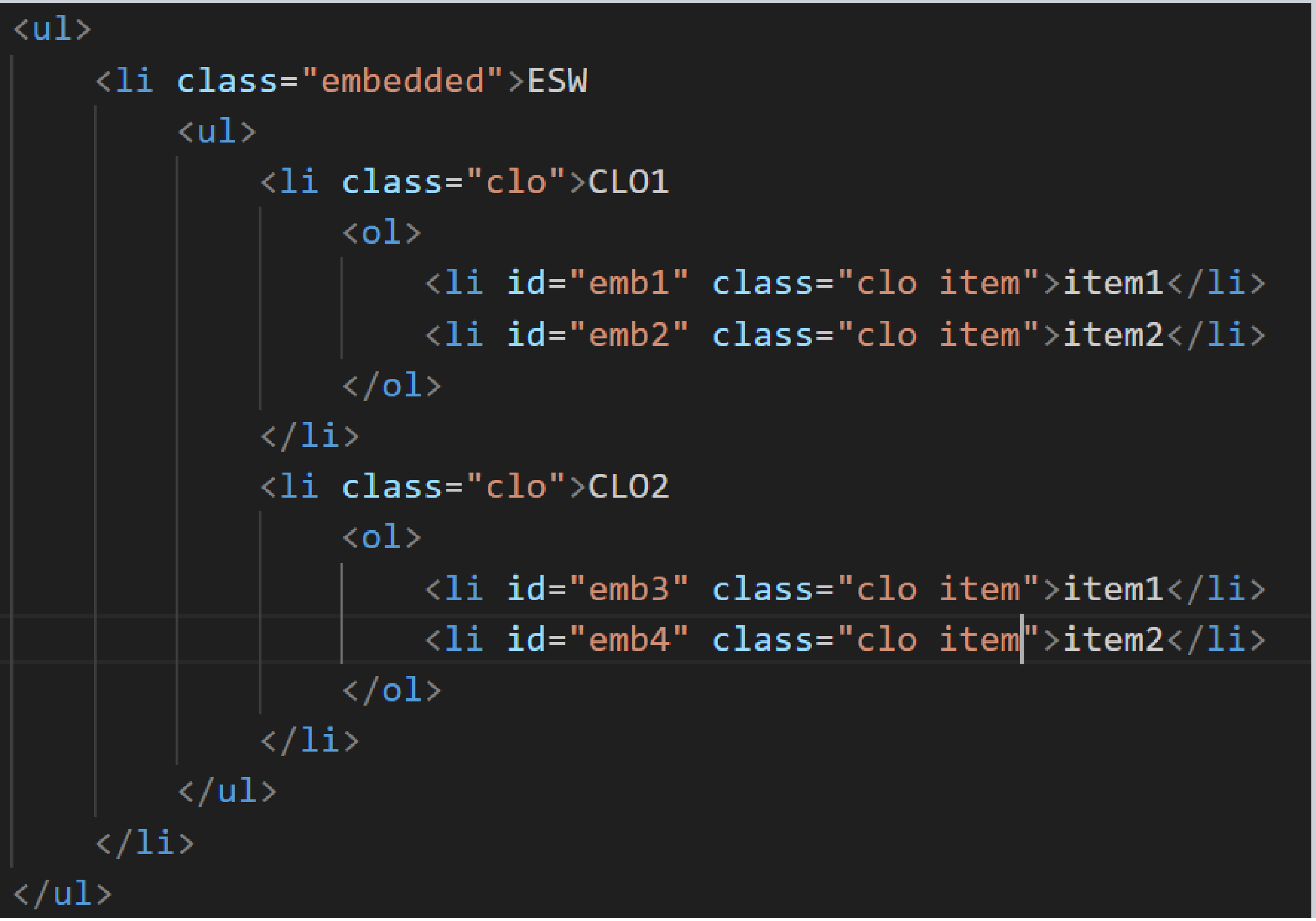
- ESW
  - ○ CLO1
    1. item1
    2. item2
  - ○ CLO2
    1. item1
    2. item2



- ESW
  - ○ CLO1
    1. item1
    2. item2
  - ○ CLO2
    1. item1
    2. item2

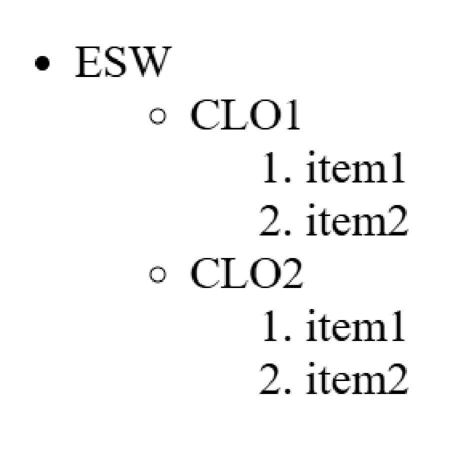
Combining Selectors (Chaining)

Syntax: SelectorSelector { CSS Rule; }

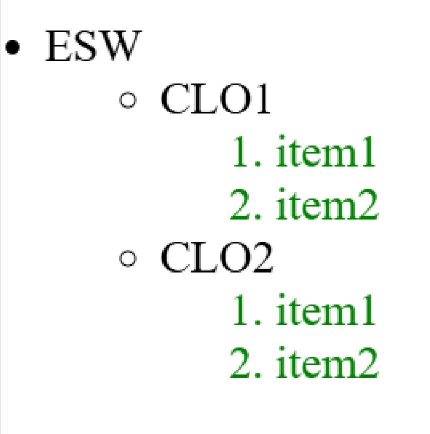


```
<ul>
    <li class="embedded">ESW
        <ul>
            <li class="clo">CLO1
                <ol>
                    <li id="emb1" class="clo item">item1</li>
                    <li id="emb2" class="clo item">item2</li>
                </ol>
            </li>
            <li class="clo">CLO2
                <ol>
                    <li id="emb3" class="clo item">item1</li>
                    <li id="emb4" class="clo item">item2</li>
                </ol>
            </li>
        </ul>
    </li>
</ul>
```

li.clo.item{color:green;}



- ESW
  - CLO1
    1. item1
    2. item2
  - CLO2
    1. item1
    2. item2

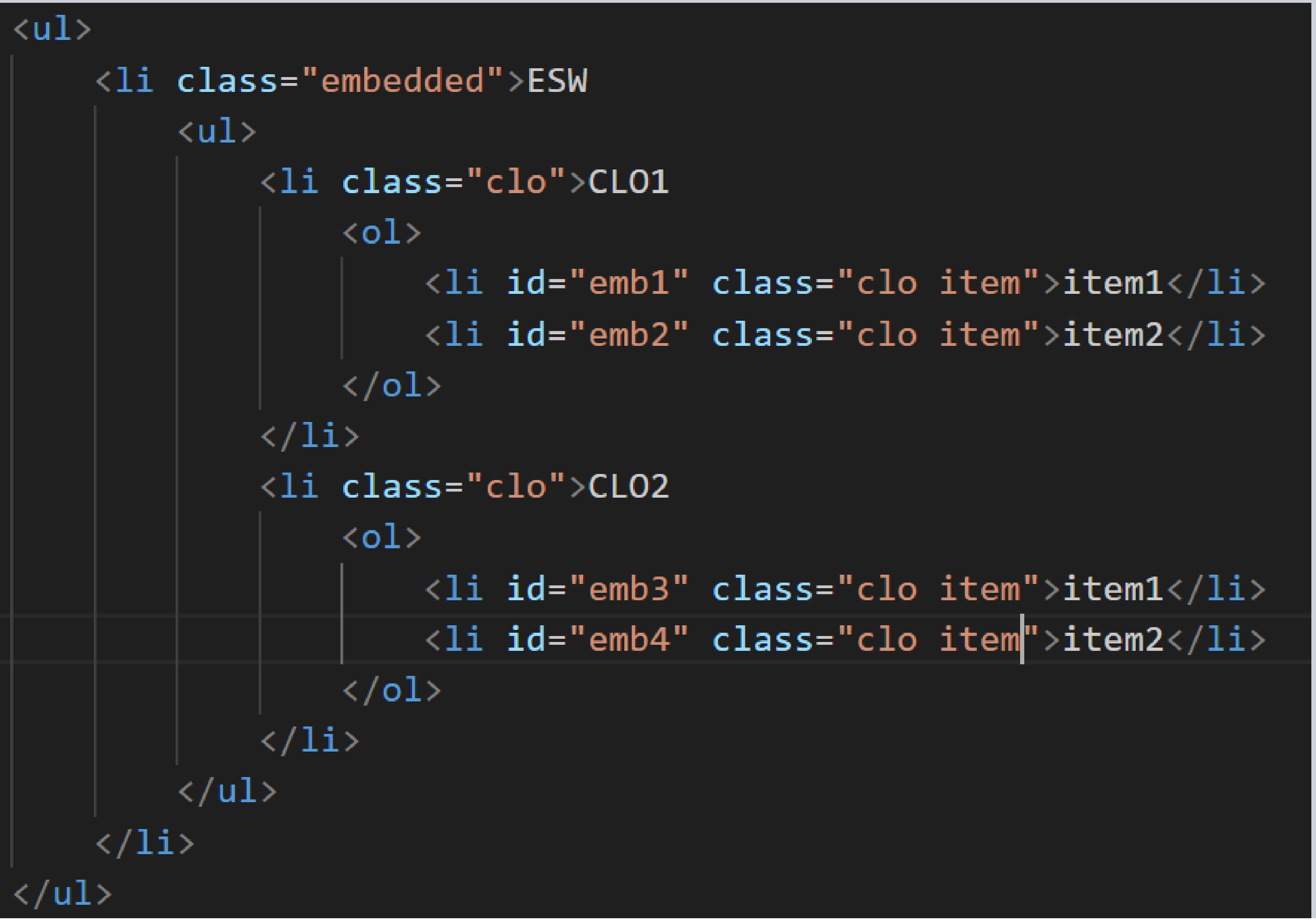


- ESW
  - CLO1
    1. item1
    2. item2
  - CLO2
    1. item1
    2. item2

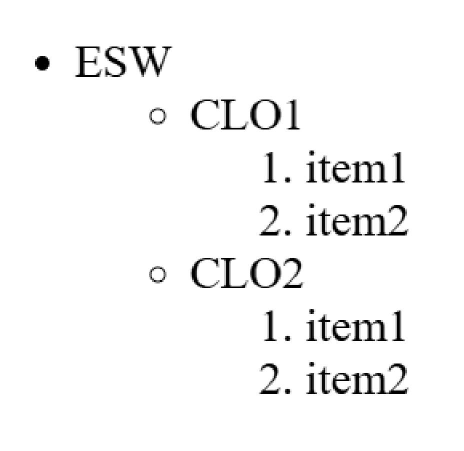
Combining Selectors (Chaining)

Syntax: SelectorSelector { CSS Rule; }

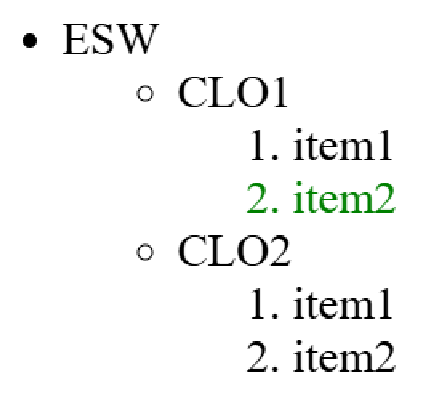
li.clo.item#emb2{color:green;}



```
<ul>
    <li class="embedded">ESW
        <ul>
            <li class="clo">CLO1
                <ol>
                    <li id="emb1" class="clo item">item1</li>
                    <li id="emb2" class="clo item">item2</li>
                </ol>
            </li>
            <li class="clo">CLO2
                <ol>
                    <li id="emb3" class="clo item">item1</li>
                    <li id="emb4" class="clo item">item2</li>
                </ol>
            </li>
        </ul>
    </li>
</ul>
```



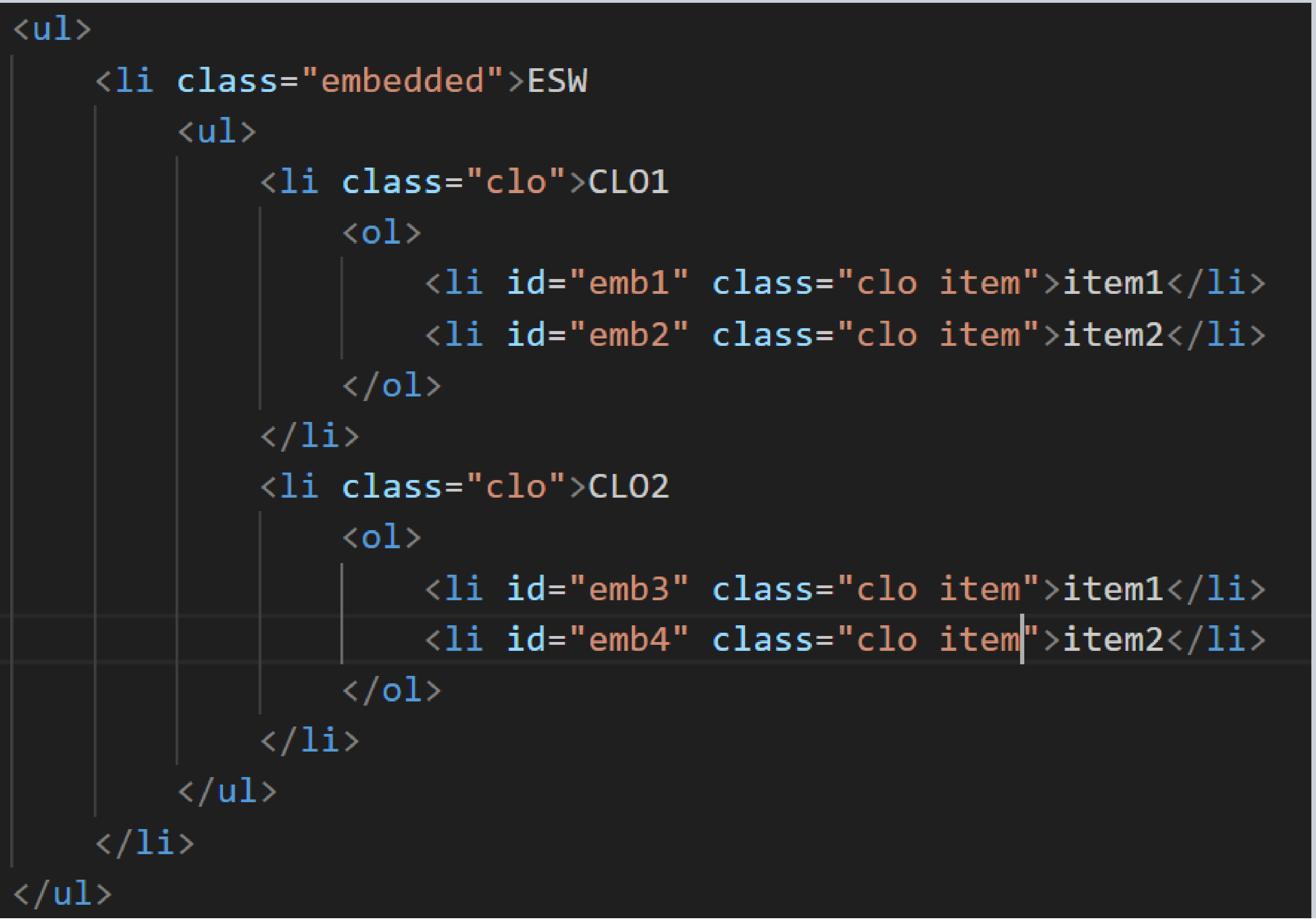
- ESW
  - ○ CLO1
    1. item1
    2. item2
  - ○ CLO2
    1. item1
    2. item2



- ESW
  - ○ CLO1
    1. item1
    2. item2
  - ○ CLO2
    1. item1
    2. item2

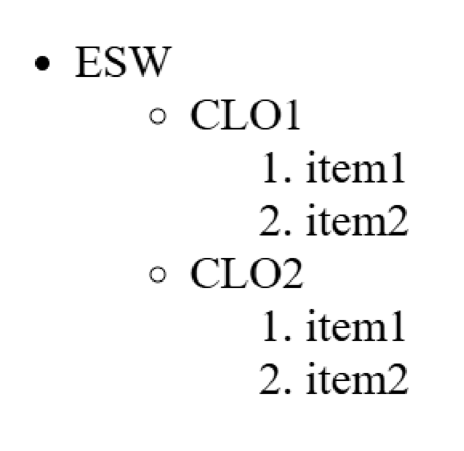
Combine Combining Selectors

Any way to combine two or more selectors

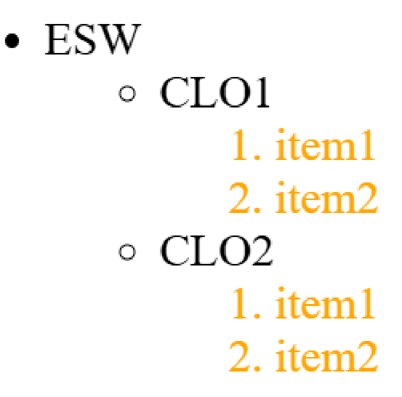


```
<ul>
    <li class="embedded">ESW
        <ul>
            <li class="clo">CLO1
                <ol>
                    <li id="emb1" class="clo item">item1</li>
                    <li id="emb2" class="clo item">item2</li>
                </ol>
            </li>
            <li class="clo">CLO2
                <ol>
                    <li id="emb3" class="clo item">item1</li>
                    <li id="emb4" class="clo item">item2</li>
                </ol>
            </li>
        </ul>
    </li>
</ul>
```

ol> li.clo{color:orange}



- ESW
    - CLO1
        1. item1
        2. item2
    - CLO2
        1. item1
        2. item2



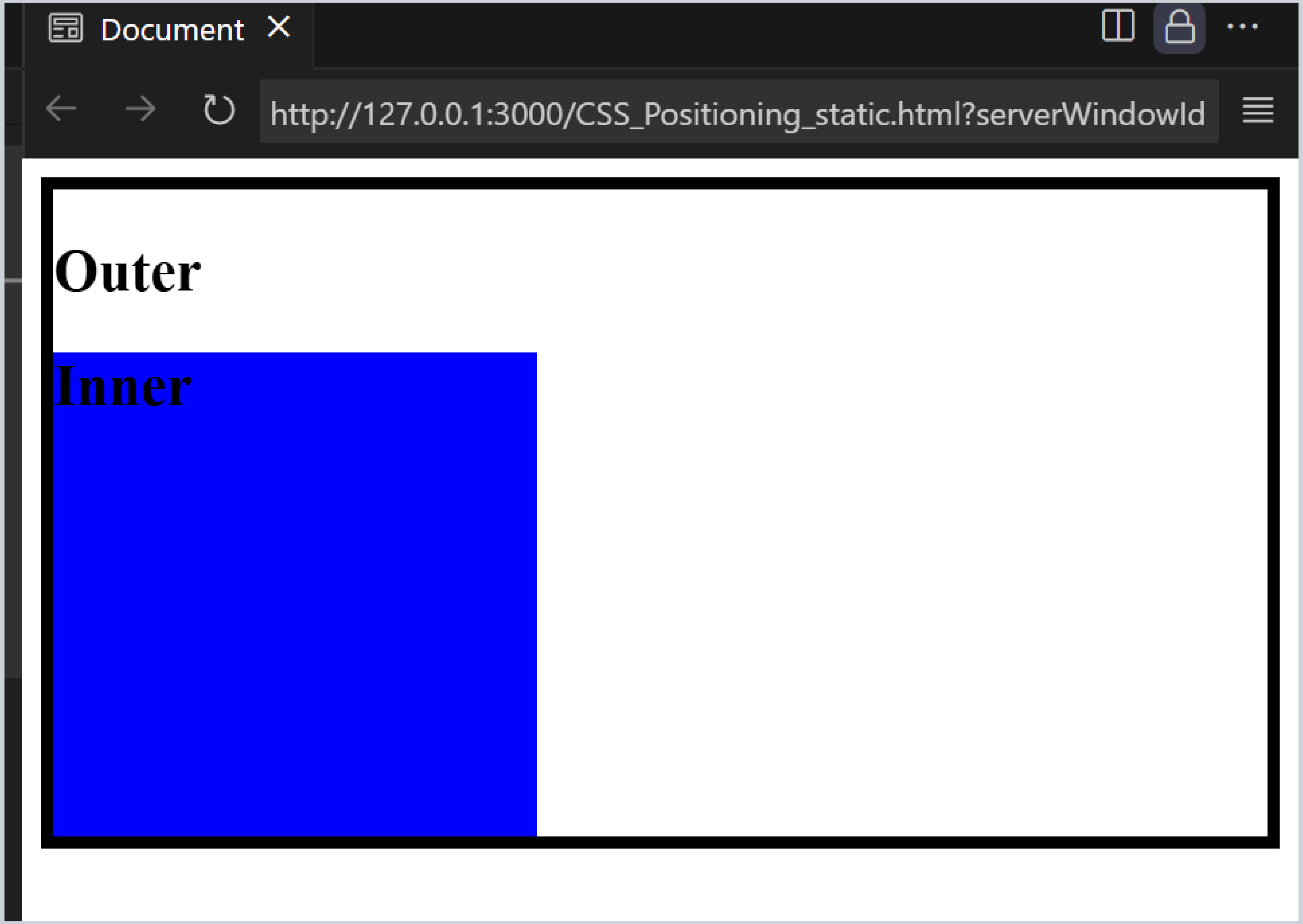
- ESW
    - CLO1
        1. item1
        2. item2
    - CLO2
        1. item1
        2. item2

CSS Positioning - Static

- Default Position (Normal flow of HTML elements)

```
<div class="outer">
    <h2>Outer</h2>
        <div class="inner">
            <h2>Inner</h2>
        </div>
  </div>
```

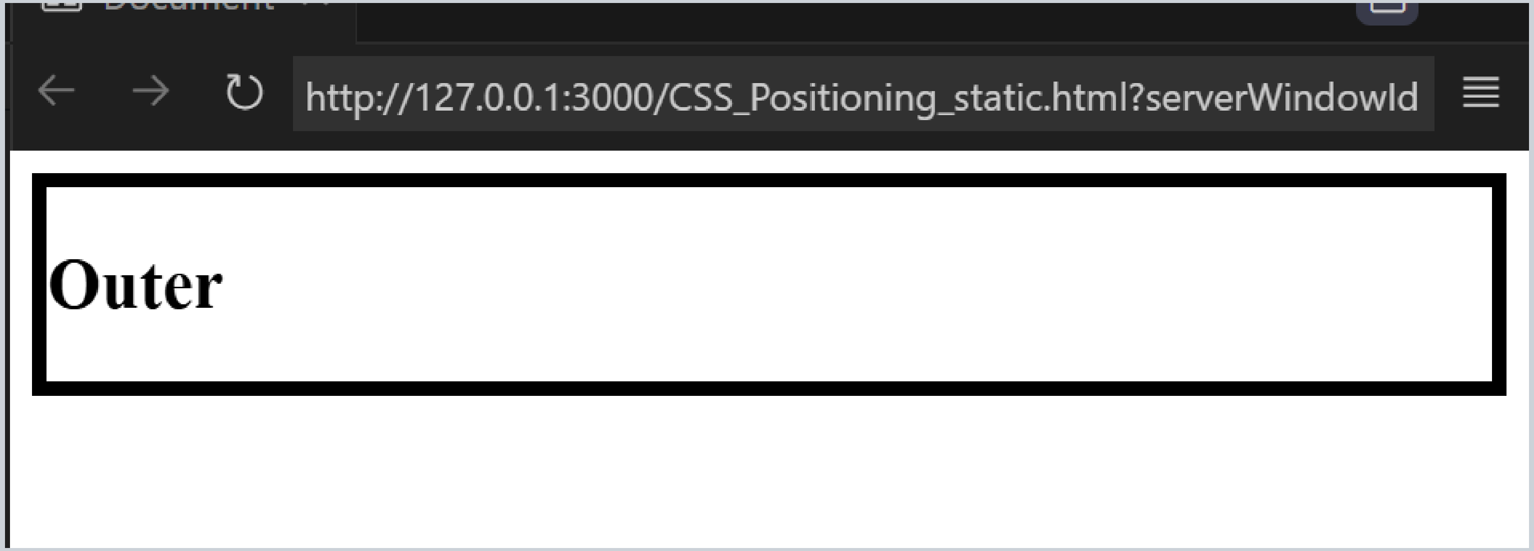
```
.outer{
        border: 5px solid black;
    }
    .inner{
        background-color: blue;
        height: 200px;
        width:200px;
    }
```


Document
http://127.0.0.1:3000/CSS_Positioning_static.html?serverWindowId
Outer
Inner

CSS Positioning - Static

```
<div class="outer">
    <h2>Outer</h2>
        <div class="inner">
        </div>
  </div>
```

```
.outer{
      border: 5px solid black;
      position: static;
    }
    .inner{
      background-color: blue;
      }
```



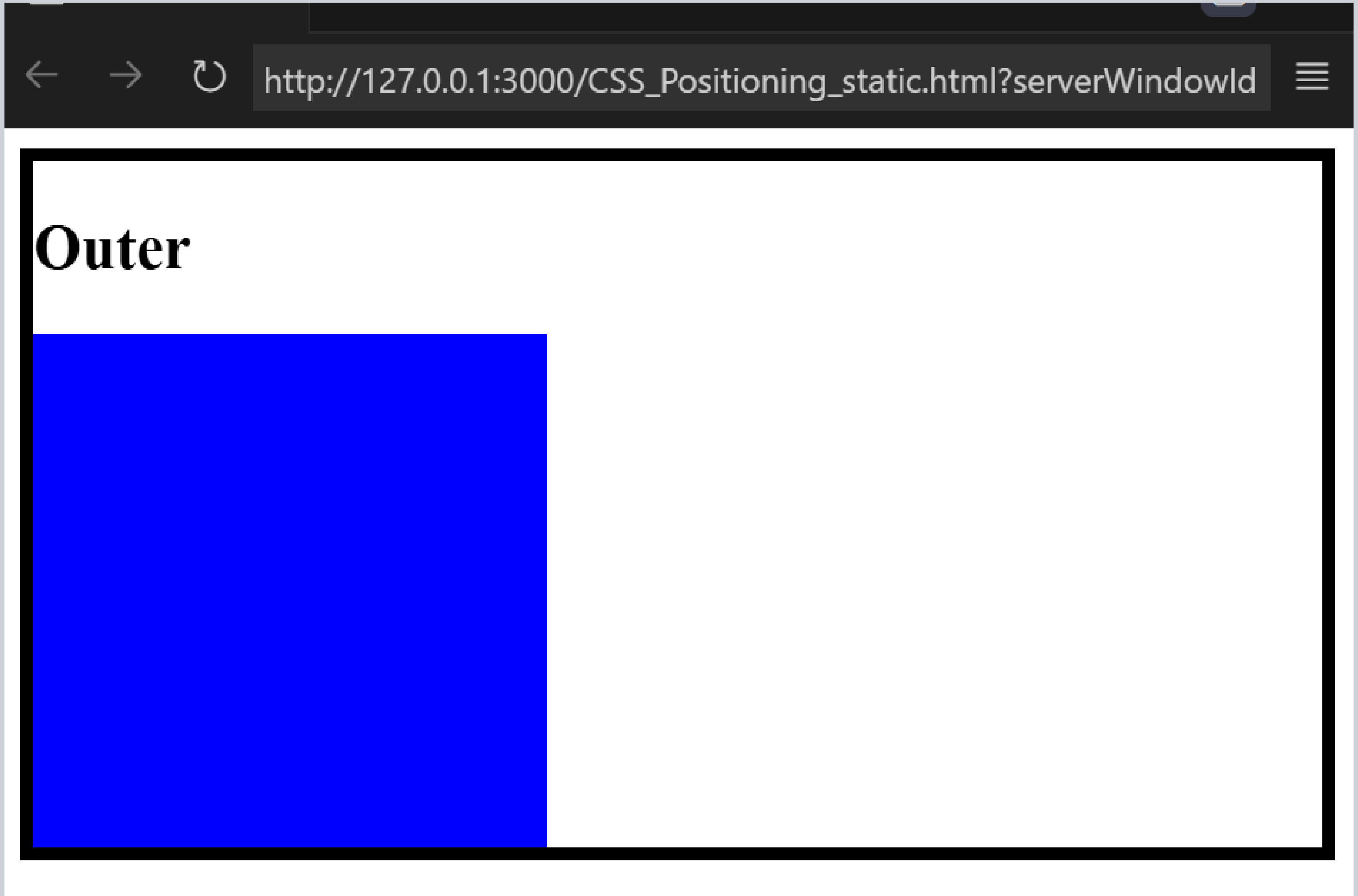
http://127.0.0.1:3000/CSS_Positioning_static.html?serverWindowId

Outer

CSS Positioning - Static

```
<div class="outer">
     <h2>Outer</h2>
          <div class="inner">
          </div>
  </div>
```

```
.outer{
      border: 5px solid black;
      position:static;
   }
   .inner{
      background-color: blue;
      position: static;
      height: 200px;
      width: 200px;
   }
```



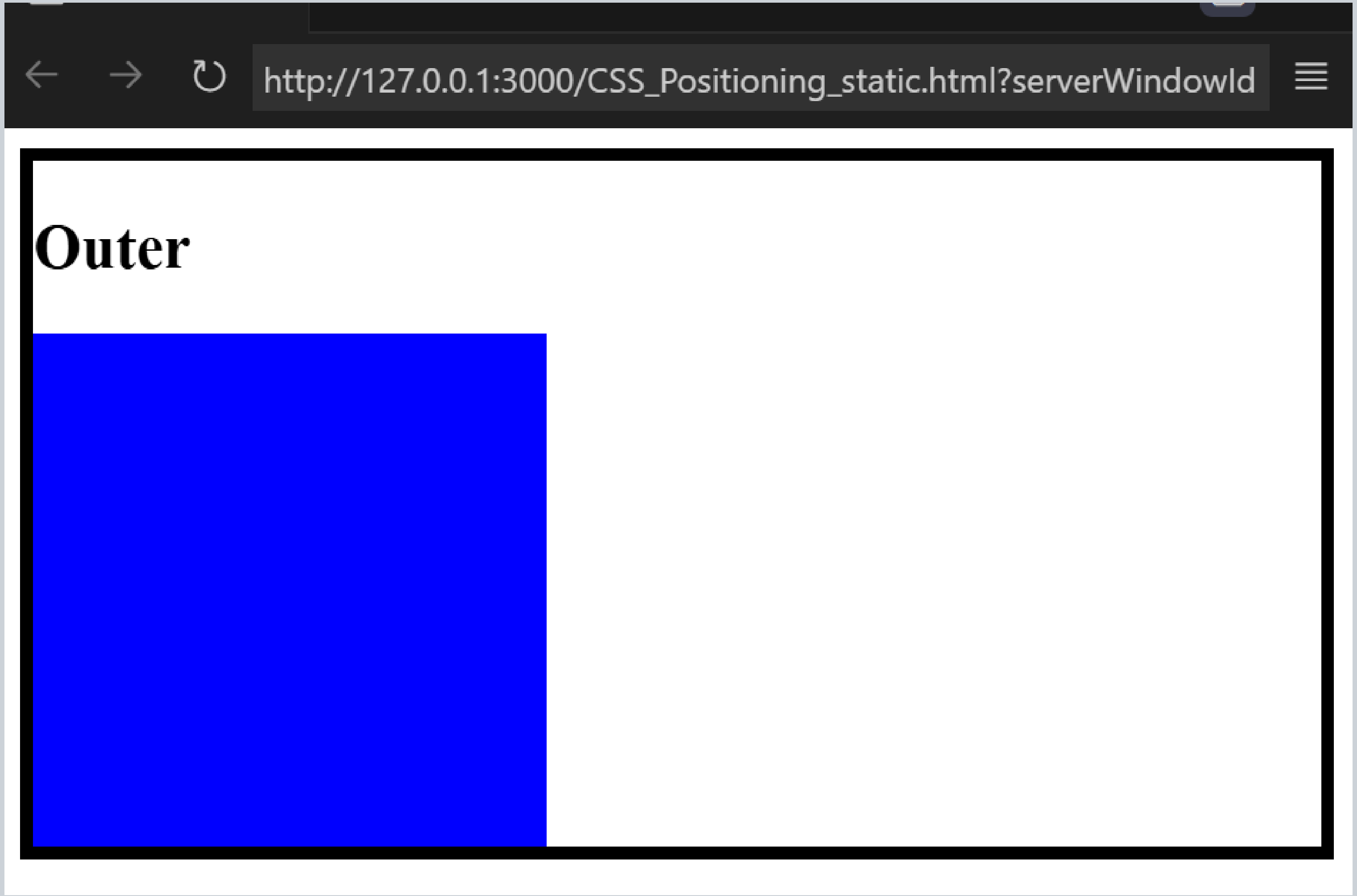
http://127.0.0.1:3000/CSS_Positioning_static.html?serverWindowId

Outer

CSS Positioning - Relative

Relative to static location

```
.outer{
        border: 5px solid black;
        position: relative;
    }
    .inner{
        background-color: blue;
        position: static;
        height: 200px;
        width: 200px;
    }
```



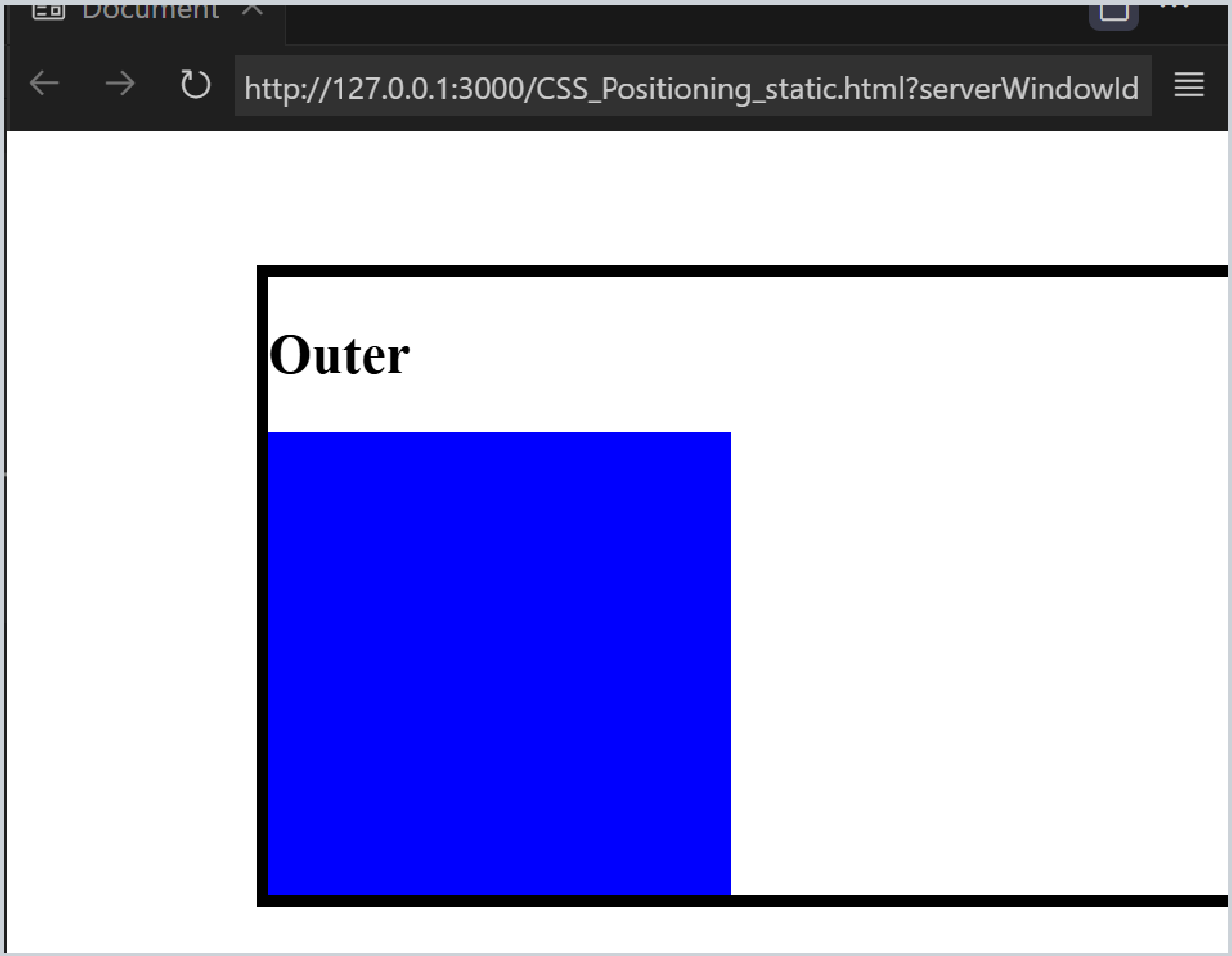
http://127.0.0.1:3000/CSS_Positioning_static.html?serverWindowId

Outer

CSS Positioning - Relative

Relative to static location

```
.outer{
        border: 5px solid black;
        position:relative;
        top:50px;
        left:100px;

    }
    .inner{
        background-color: blue;
        position: static;
        height: 200px;
        width: 200px;
    }
```

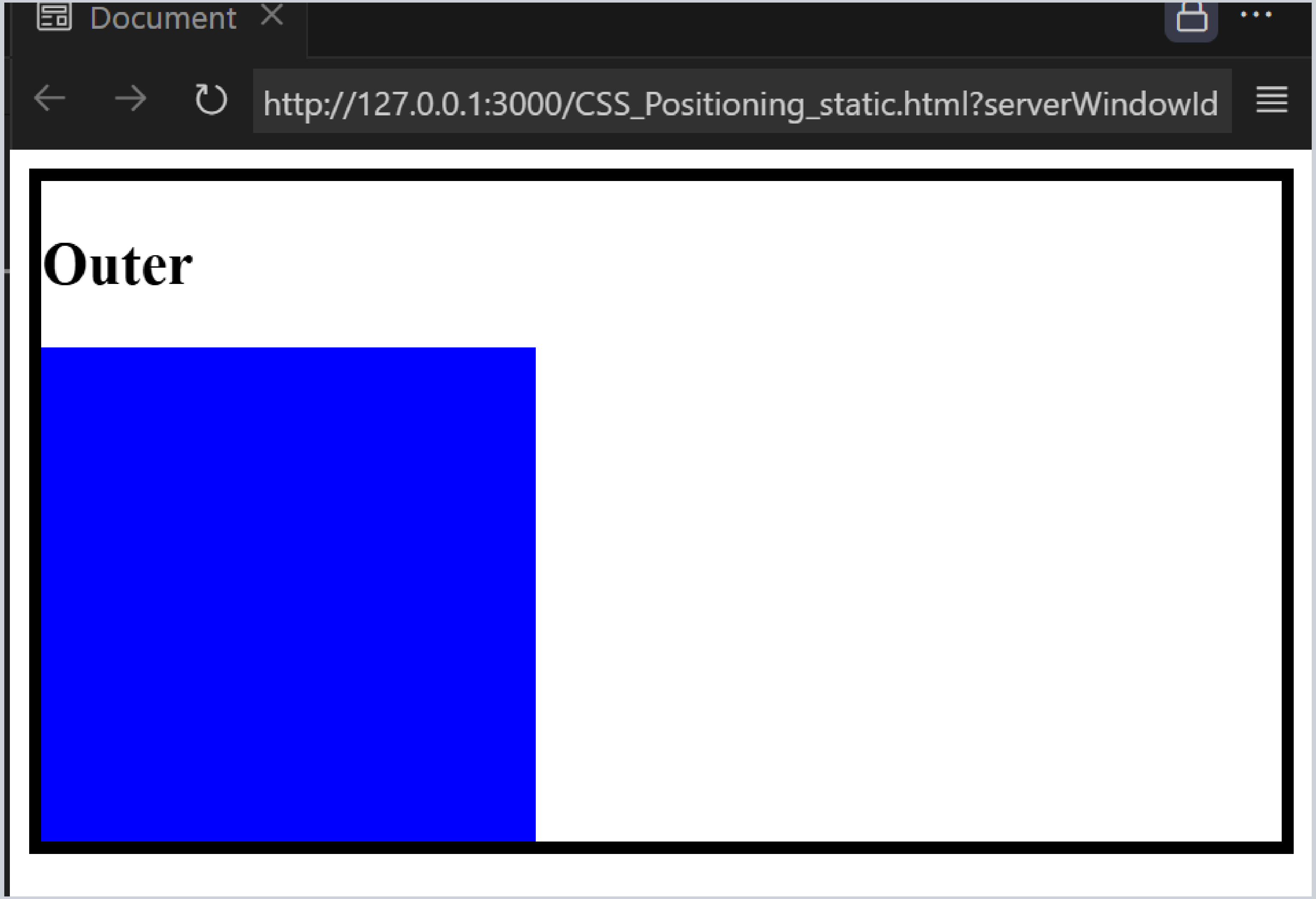


Document
http://127.0.0.1:3000/CSS_Positioning_static.html?serverWindowId
Outer

CSS Positioning - Relative

Relative to static location

```
.outer{
        border: 5px solid black;
        position:static;
    }
    .inner{
        background-color: blue;
        position: relative;
        height: 200px;
        width: 200px;
    }
```

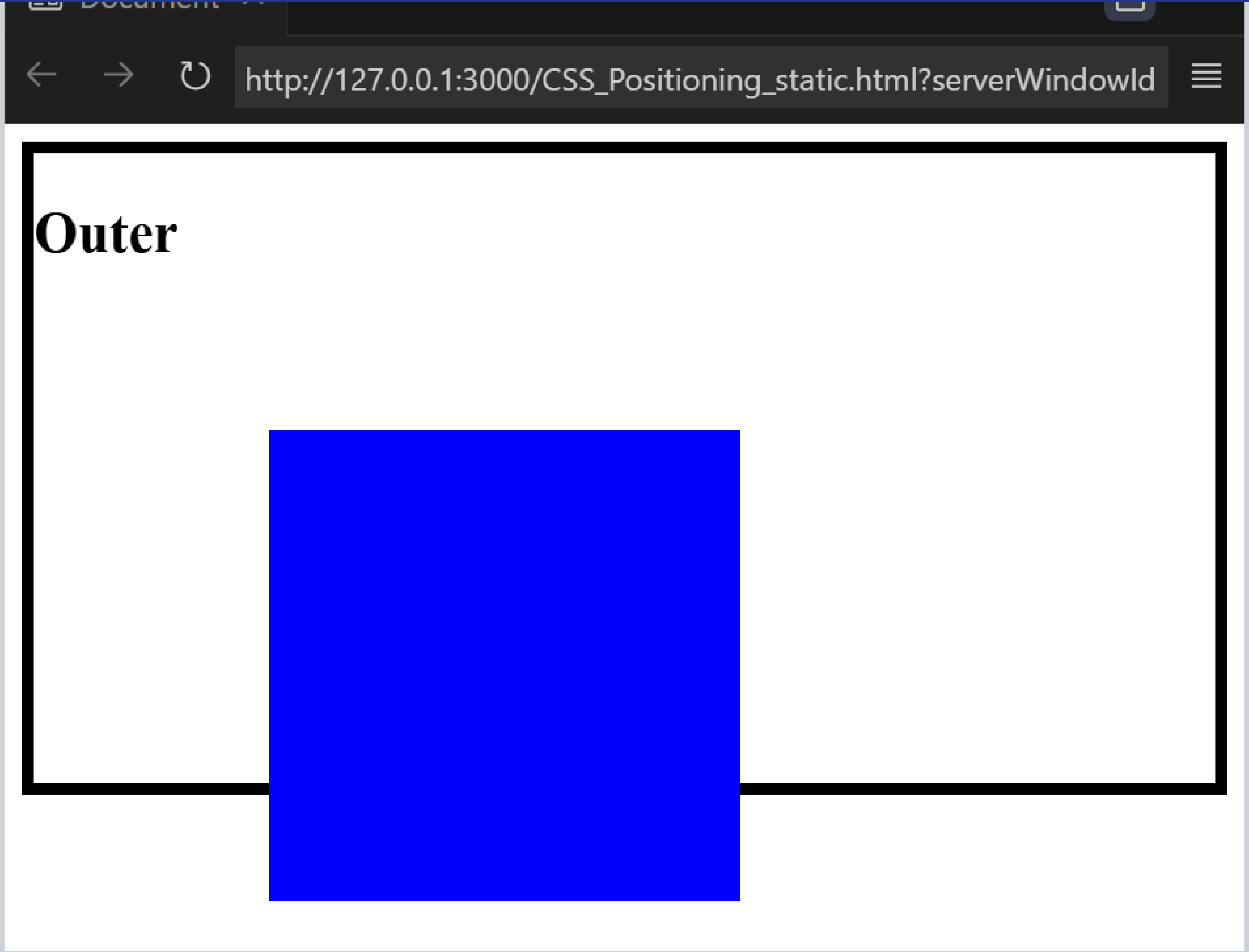


Document
http://127.0.0.1:3000/CSS_Positioning_static.html?serverWindowId
Outer

CSS Positioning - Relative

Relative to static location

```
.outer{
        border: 5px solid black;
        position:static;
    }
    .inner{
        background-color: blue;
        position: relative;
        height: 200px;
        width: 200px;
        top: 50px;
        left:100px;
    }
```

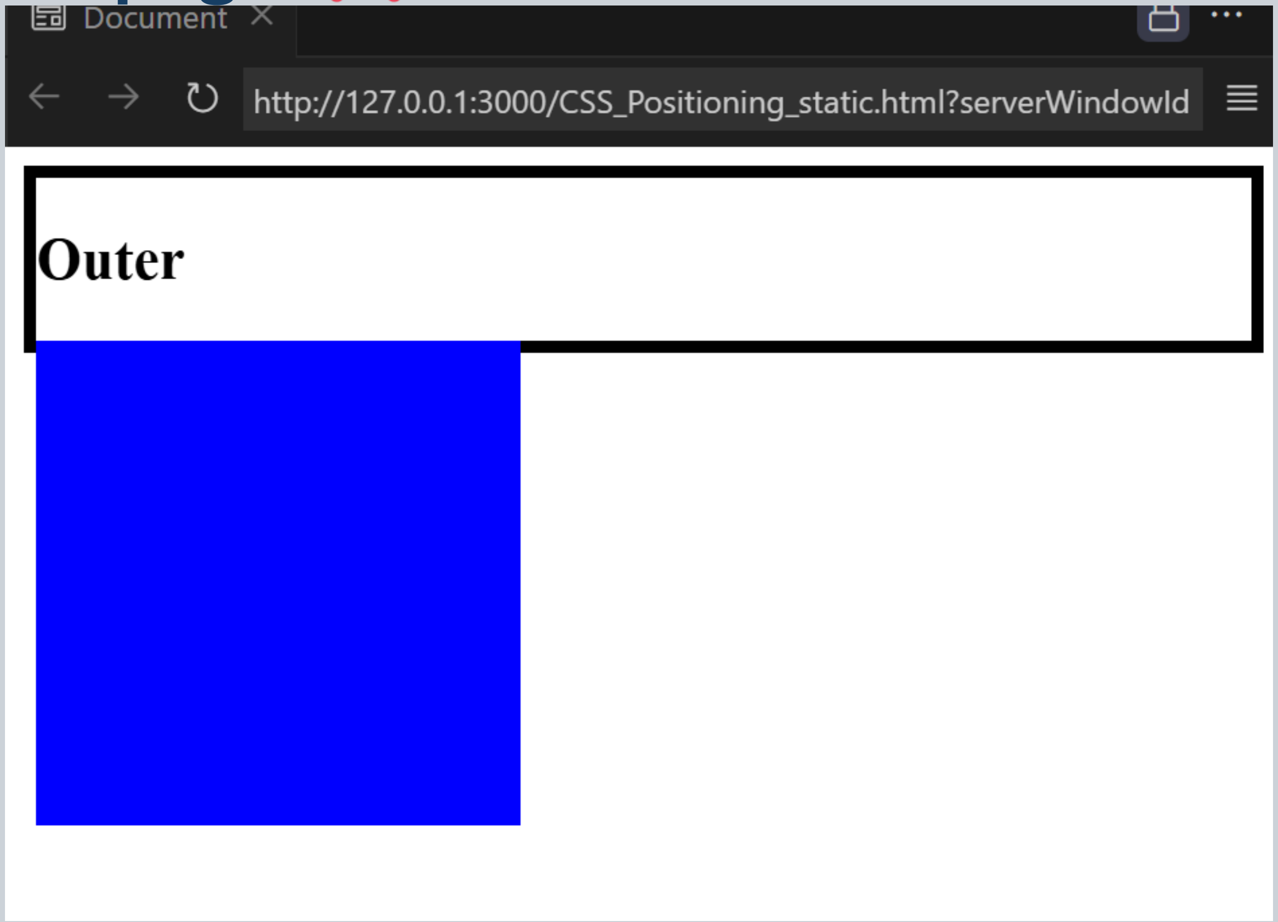


http://127.0.0.1:3000/CSS_Positioning_static.html?serverWindowId

CSS Positioning - absolute

- Relative to nears positioned ancestor ✅
- Or Relative to top left corner of webpage. ❌

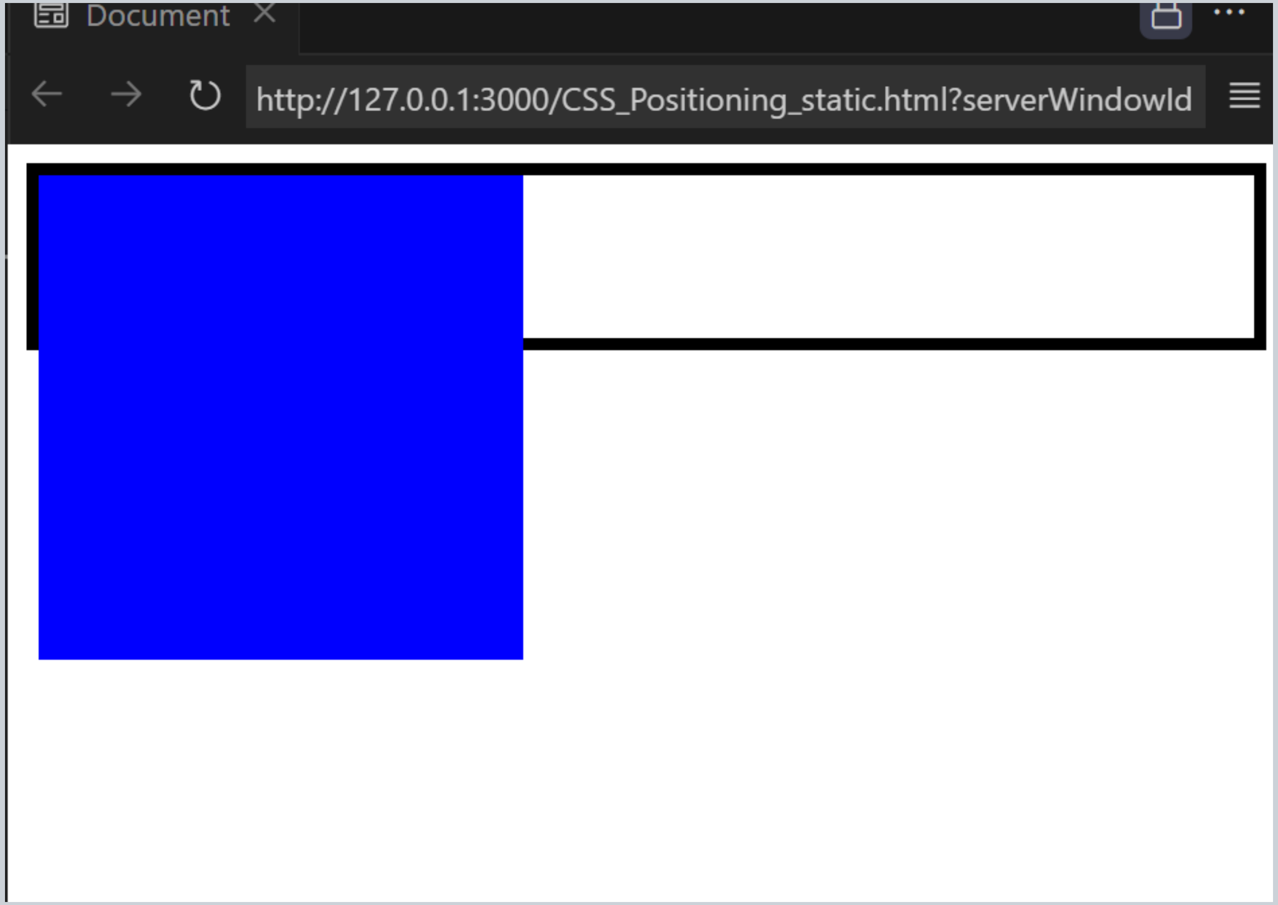
```
.outer{
        border: 5px solid black;
        position: relative;
    }
    .inner{
        background-color: blue;
        position: absolute;
        height:200px;
        width:200px;
    }
```


Document
http://127.0.0.1:3000/CSS_Positioning_static.html?serverWindowId
Outer

CSS Positioning - absolute

- Relative to nears positioned ancestor ✅
- Or Relative to top left corner of webpage. ❌

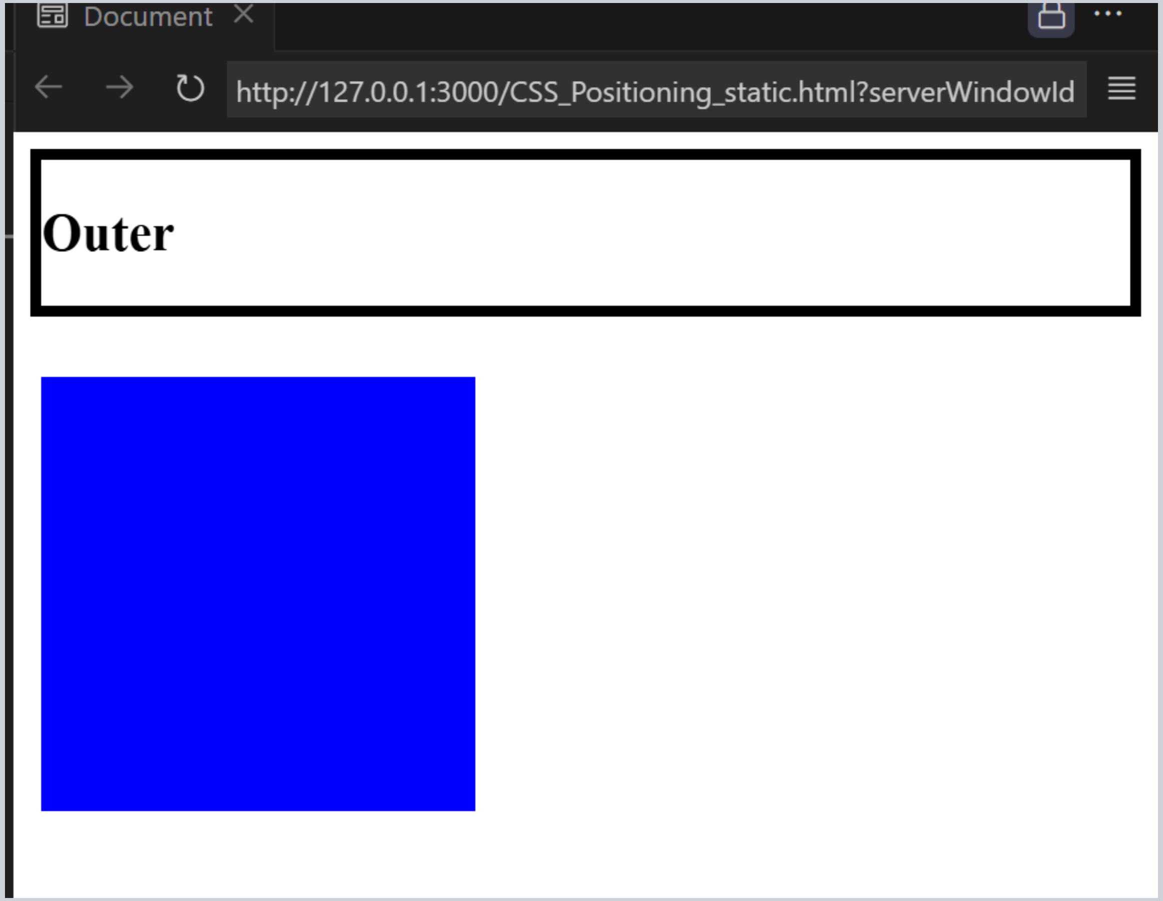
```
.outer{
        border: 5px solid black;
        position: relative;
    }
    .inner{
        background-color: blue;
        position: absolute;
        height:200px;
        width:200px;
        top: 0px;
    }
```


Document
http://127.0.0.1:3000/CSS_Positioning_static.html?serverWindowId

CSS Positioning - absolute

- Relative to nears positioned ancestor ✅
- Or Relative to top left corner of webpage. ❌

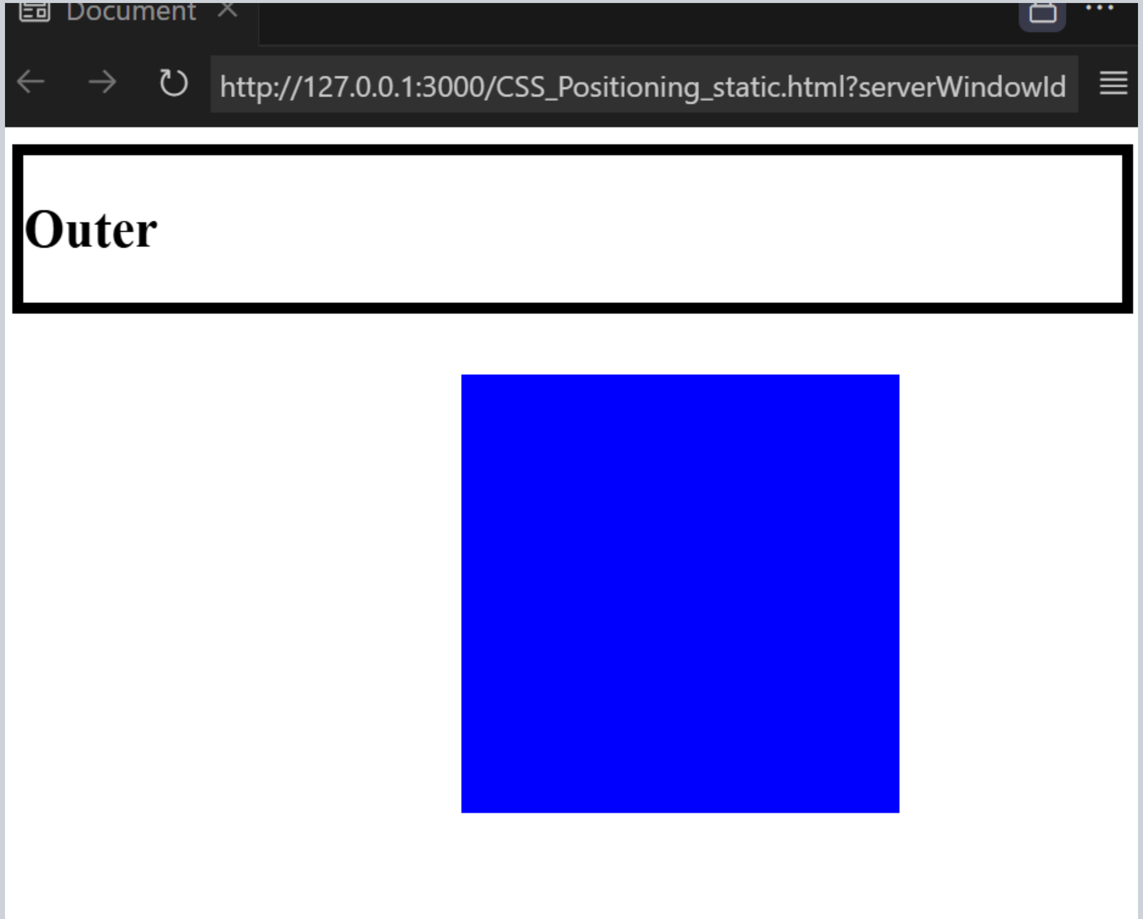
```
.outer{
        border: 5px solid black;
        position: relative;
    }
    .inner{
        background-color: blue;
        position: absolute;
        height:200px;
        width:200px;
        top: 100px;
    }
```


Document
http://127.0.0.1:3000/CSS_Positioning_static.html?serverWindowId
Outer

CSS Positioning - absolute

- **Relative to nears positioned ancestor** ✅
- **Or Relative to top left corner of webpage.** ❌

```
.outer{
        border: 5px solid black;
        position: relative;
     }
     .inner{
        background-color: blue;
        position: absolute;
        height:200px;
        width:200px;
        top: 100px;
        left:200px;
     }
```



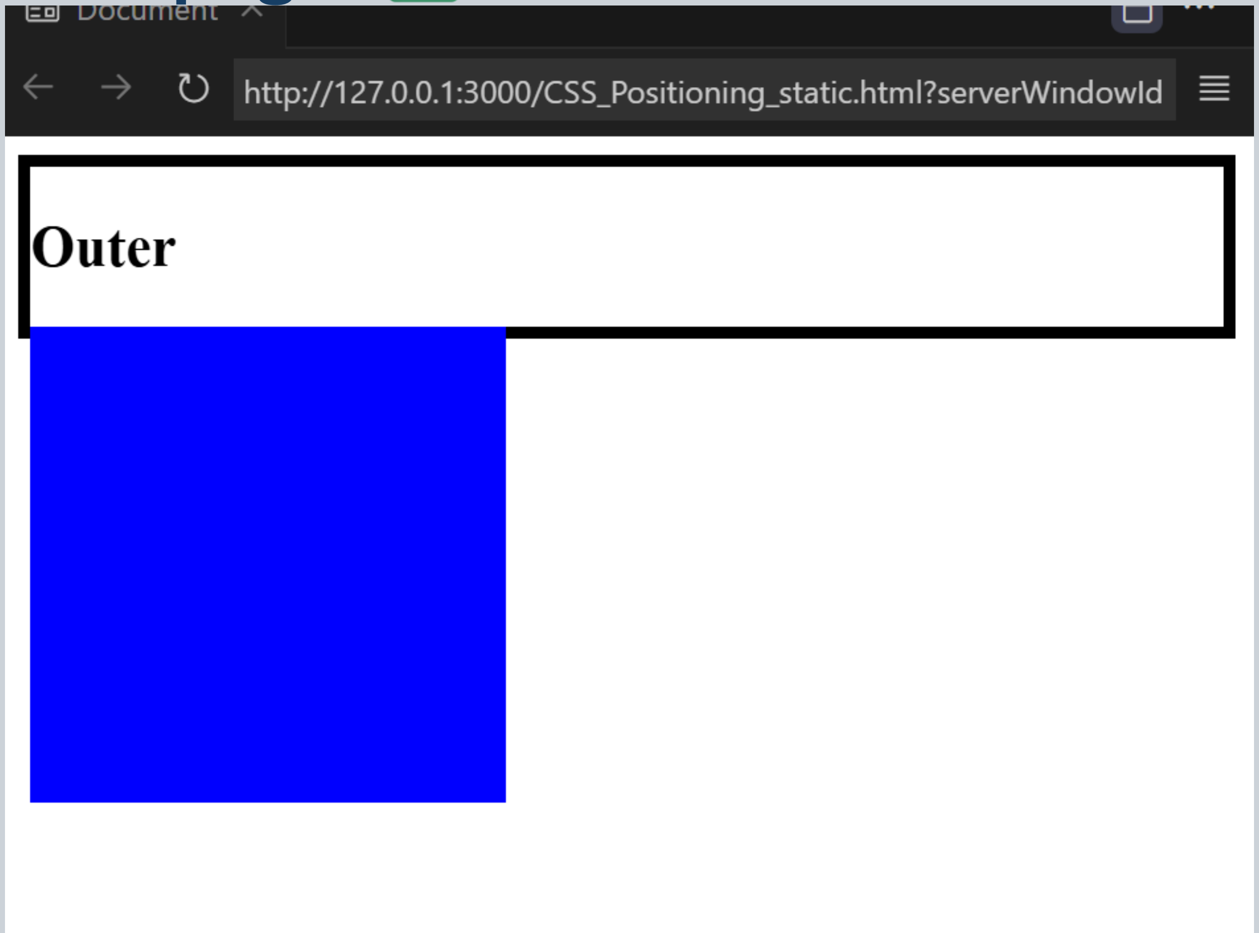
http://127.0.0.1:3000/CSS_Positioning_static.html?serverWindowId

Outer

CSS Positioning - absolute

- Relative to nears positioned ancestor ❌
- Or Relative to top left corner of webpage. ✅

```
.outer{
        border: 5px solid black;
    }
    .inner{
        background-color: blue;
        position: absolute;
        height:200px;
        width:200px;
    }
```



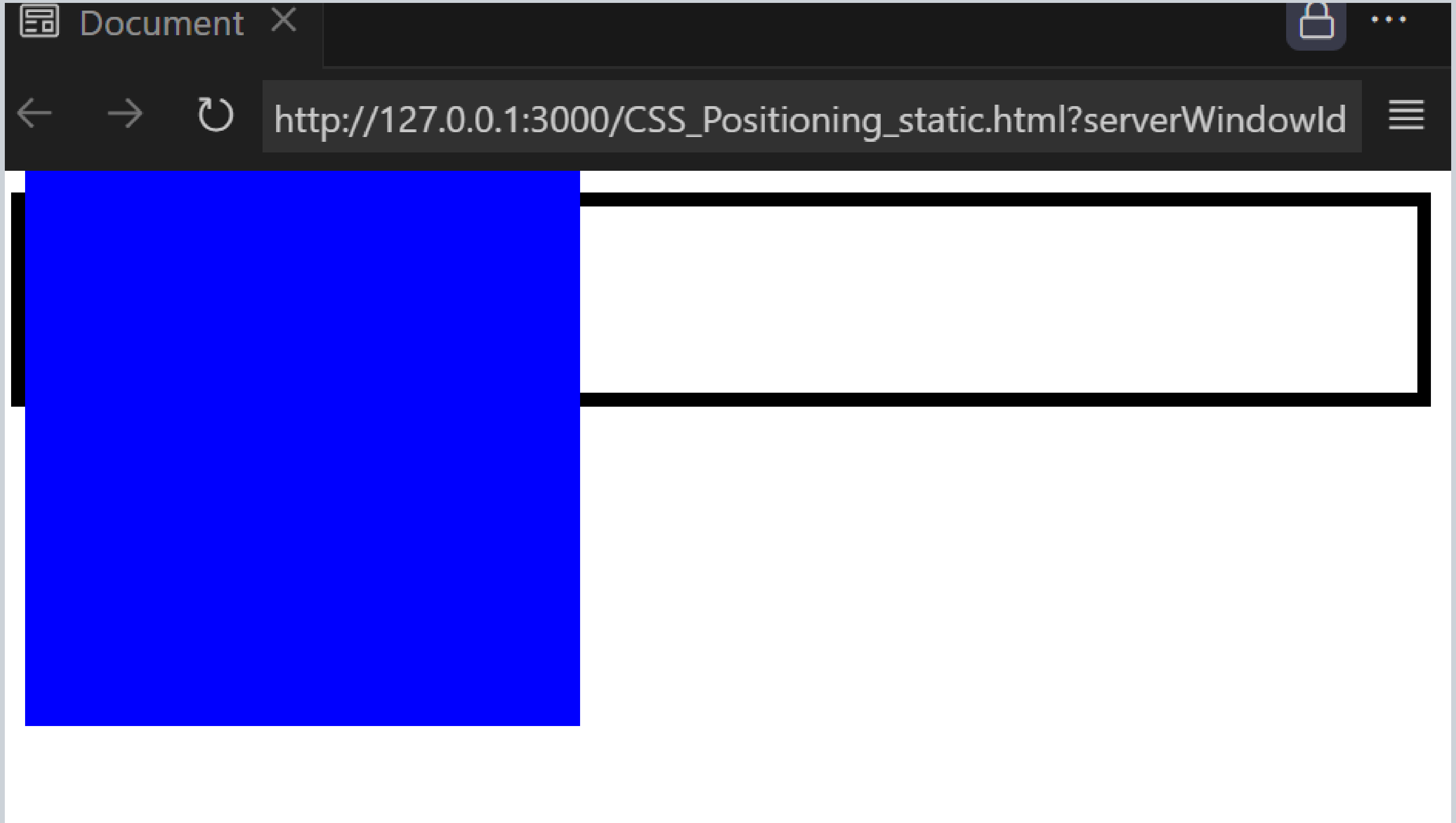
Document
http://127.0.0.1:3000/CSS_Positioning_static.html?serverWindowId

Outer

CSS Positioning - absolute

- Relative to nears positioned ancestor ❌
- Or Relative to top left corner of webpage. ✅

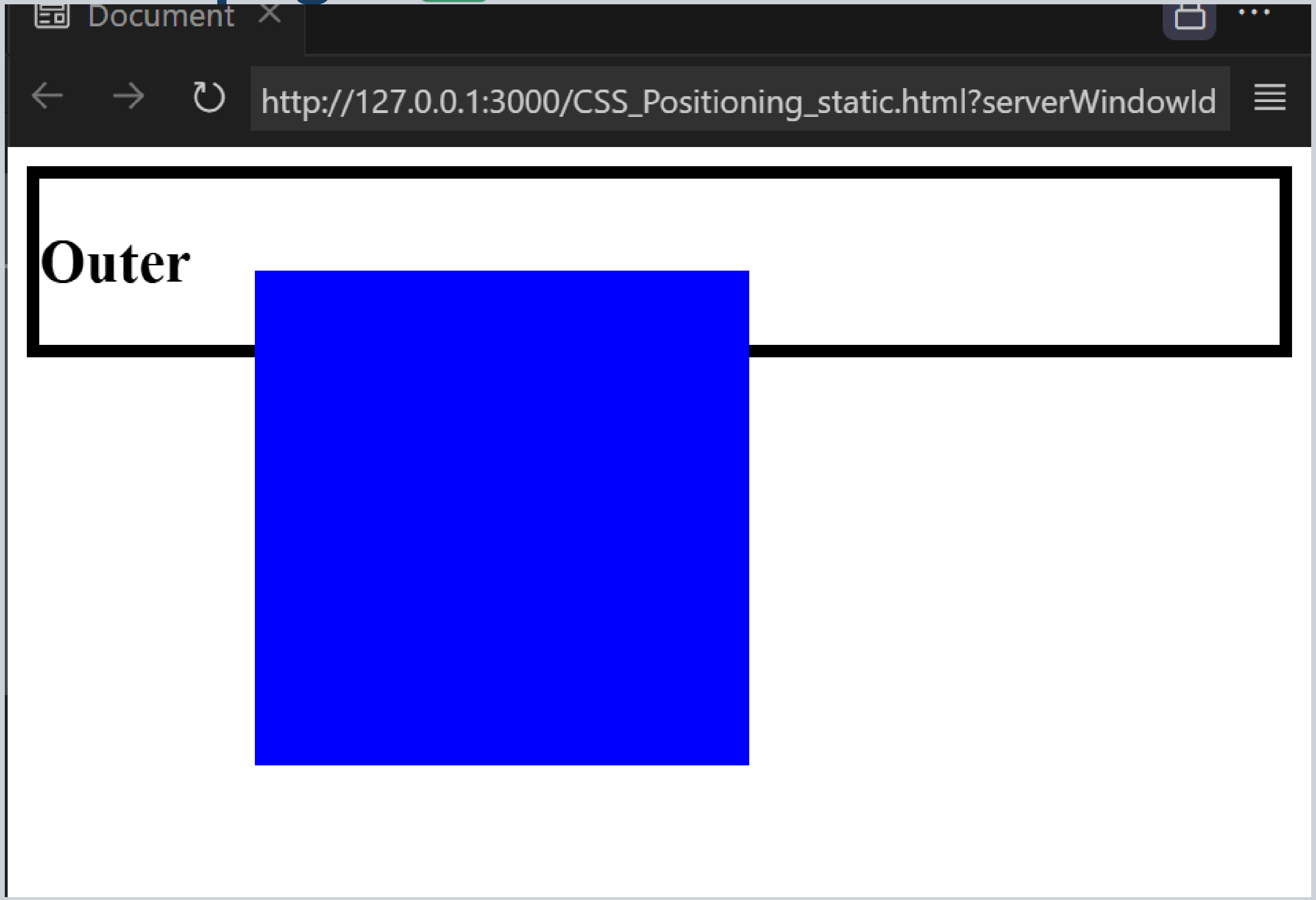
```
.outer{
        border: 5px solid black;
    }
    .inner{
        background-color: blue;
        position: absolute;
        height:200px;
        width:200px;
        top:0px;
    }
```


Document
http://127.0.0.1:3000/CSS_Positioning_static.html?serverWindowId

CSS Positioning - absolute

- Relative to nears positioned ancestor ❌
- Or Relative to top left corner of webpage. ✅

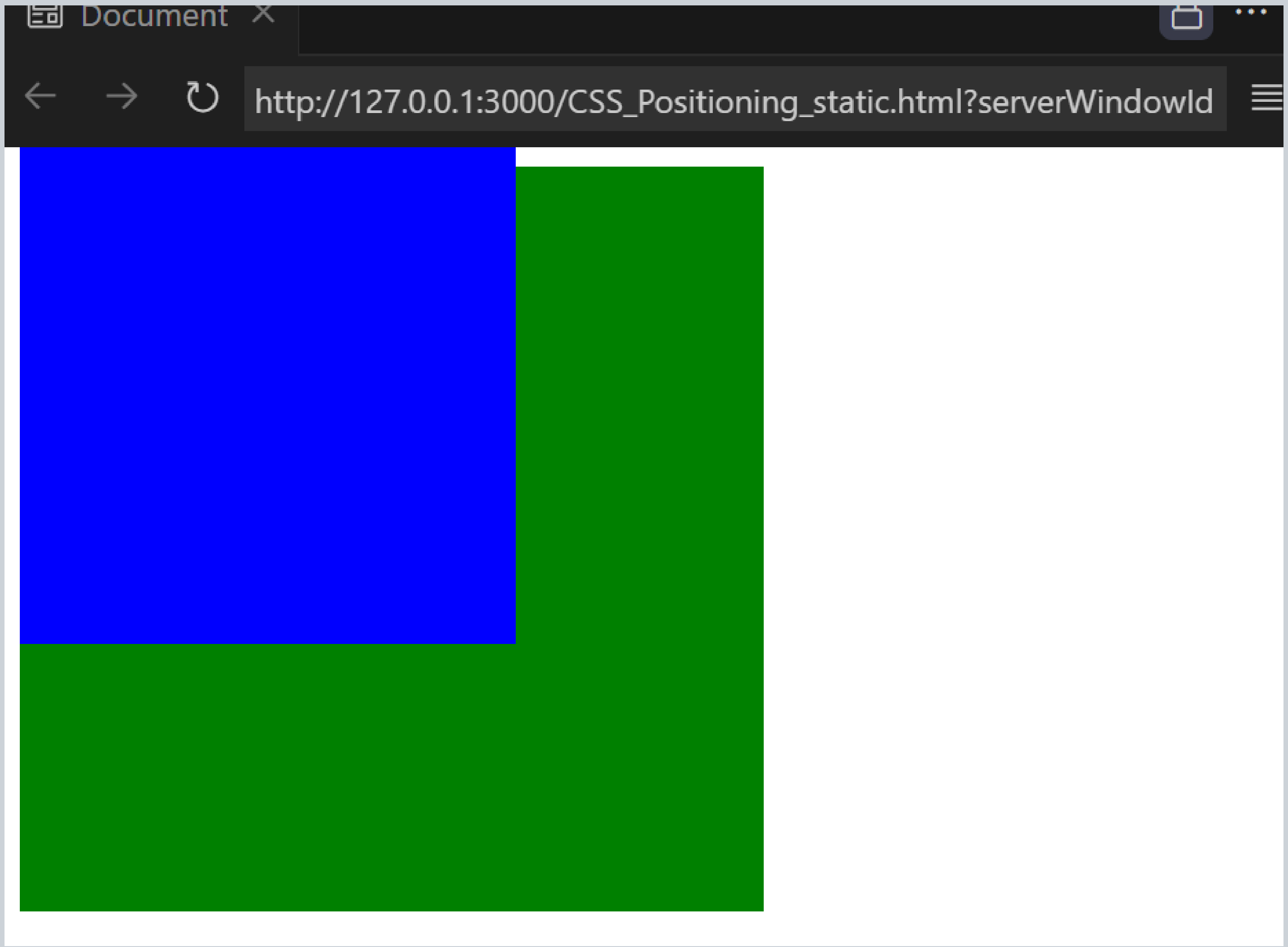
```
. .outer{
        border: 5px solid black;
     }
     .inner{
        background-color: blue;
        position: absolute;
        height:200px;
        width:200px;
        top:50px;
        left:100px;
     }
```


Document
http://127.0.0.1:3000/CSS_Positioning_static.html?serverWindowId
Outer

CSS Positioning – absolute (Z-index)

- **Default z index=0;**

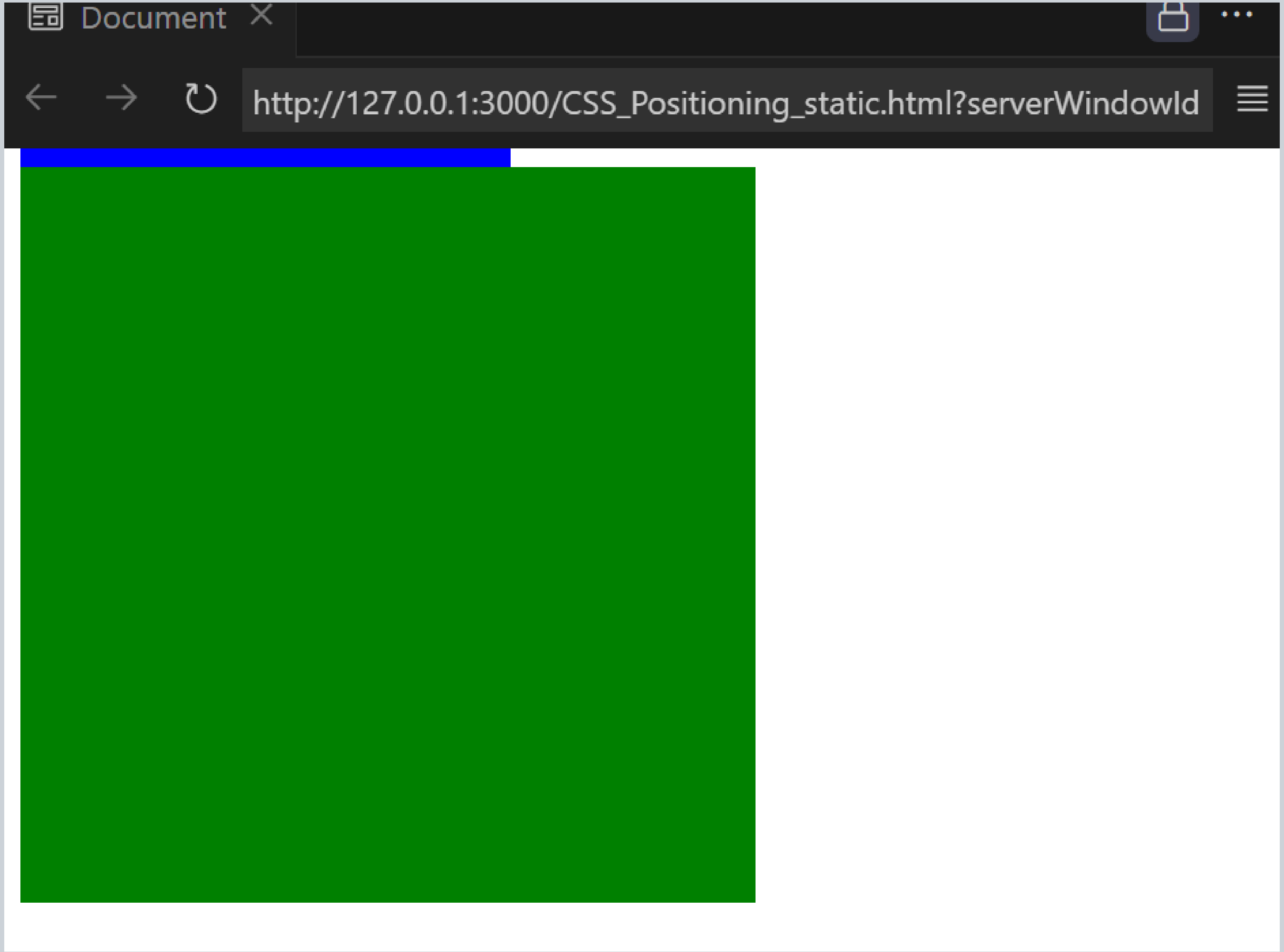
```
.outer{
      background-color: green;
      height: 300px;
      width: 300px;
  }
  .inner{
      background-color: blue;
      position: absolute;
      height:200px;
      width:200px;
      top:0px;
  }
```


Document
http://127.0.0.1:3000/CSS_Positioning_static.html?serverWindowId

CSS Positioning – absolute (Z-index)

- **Default z index= (negative);**

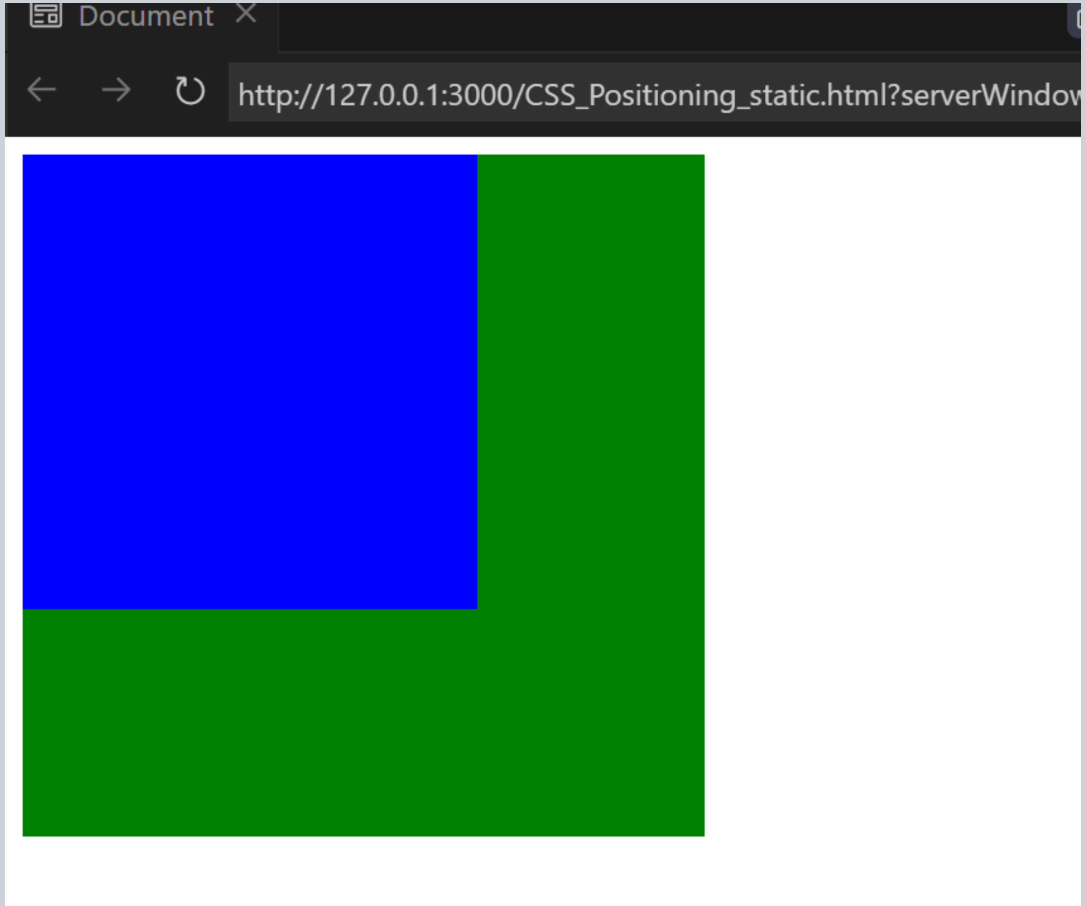
```
.outer{
        background-color: green;
        height: 300px;
        width: 300px;
    }
    .inner{
        background-color: blue;
        position: absolute;
        height:200px;
        width:200px;
        top:0px;
        z-index: -10;
    }
```


Document
http://127.0.0.1:3000/CSS_Positioning_static.html?serverWindowId

CSS Positioning - fixed

- Relative to top left corner
  - Always there (scrolling does not affect it)

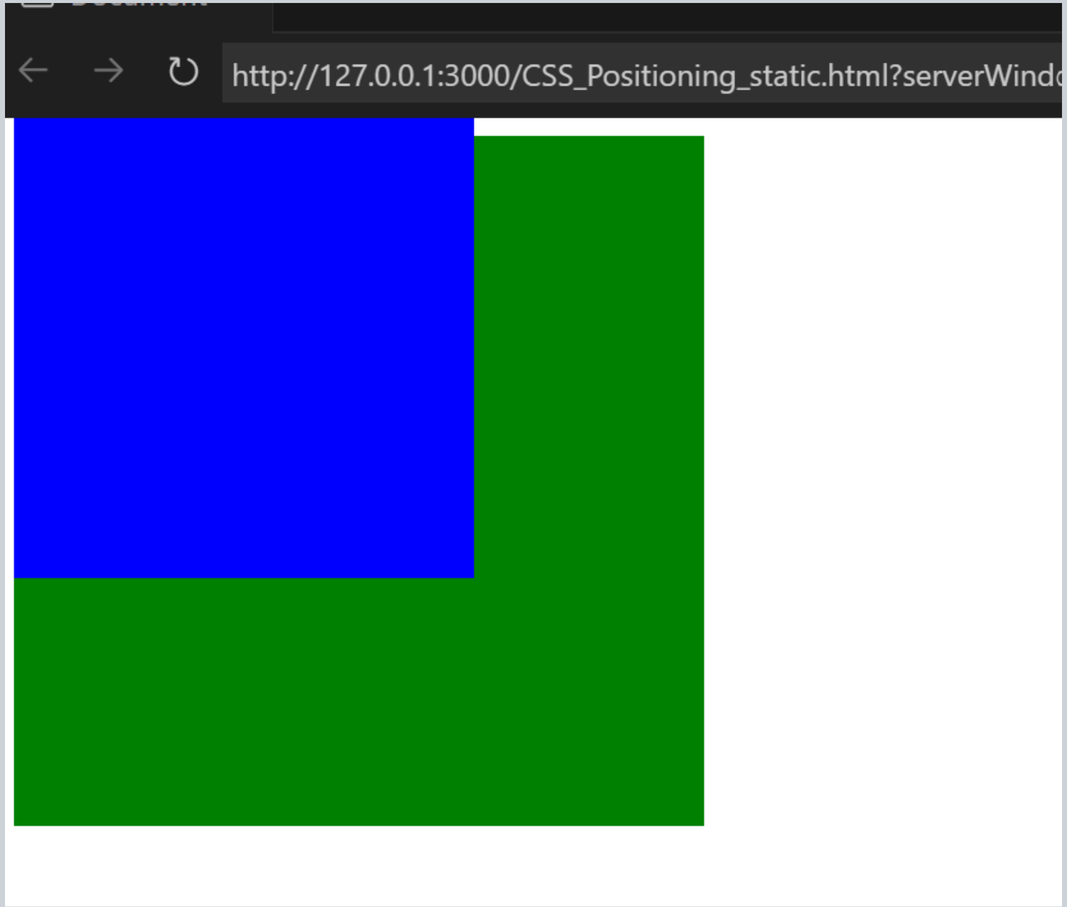
```
.outer{
      background-color: green;
      height: 300px;
      width: 300px;
    }
    .inner{
      background-color: blue;
      position: fixed;
      height:200px;
      width:200px;
    }
```


Document
http://127.0.0.1:3000/CSS_Positioning_static.html?serverWindow

CSS Positioning - fixed

- Relative to top left corner
  - Always there (scrolling does not affect it)

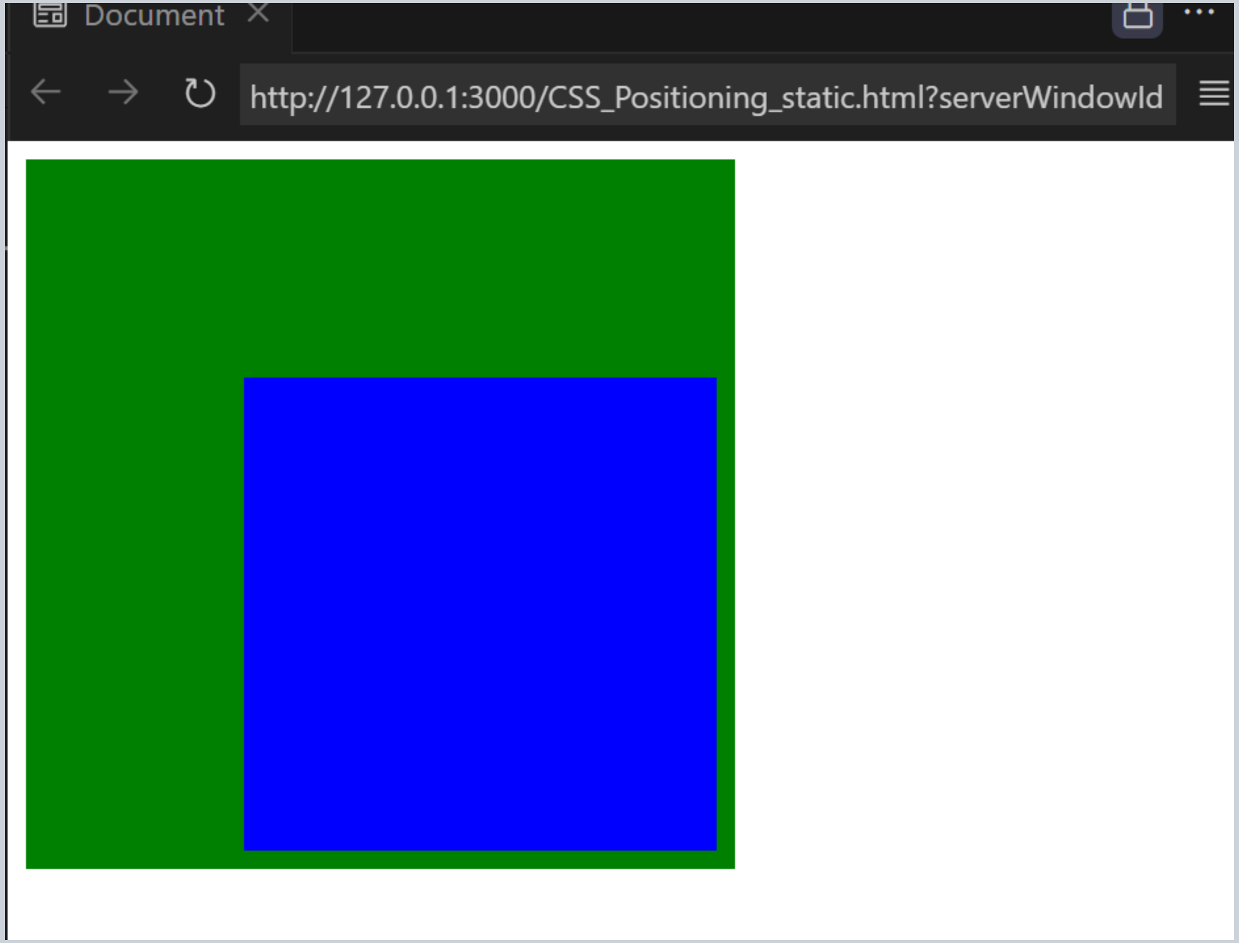
```
.outer{
        background-color: green;
        height: 300px;
        width: 300px;
     }
     .inner{
        background-color: blue;
        position: fixed;
        height:200px;
        width:200px;
        top:0px;
     }
```


http://127.0.0.1:3000/CSS_Positioning_static.html?serverWind

CSS Positioning - fixed

- Relative to top left corner
  - Always there (scrolling does not affect it.)

```
.outer{
        background-color: green;
        height: 300px;
        width: 300px;
    }
    .inner{
        background-color: blue;
        position: fixed;
        height:200px;
        width:200px;
        top:100px;
        left:100px;
    }
```

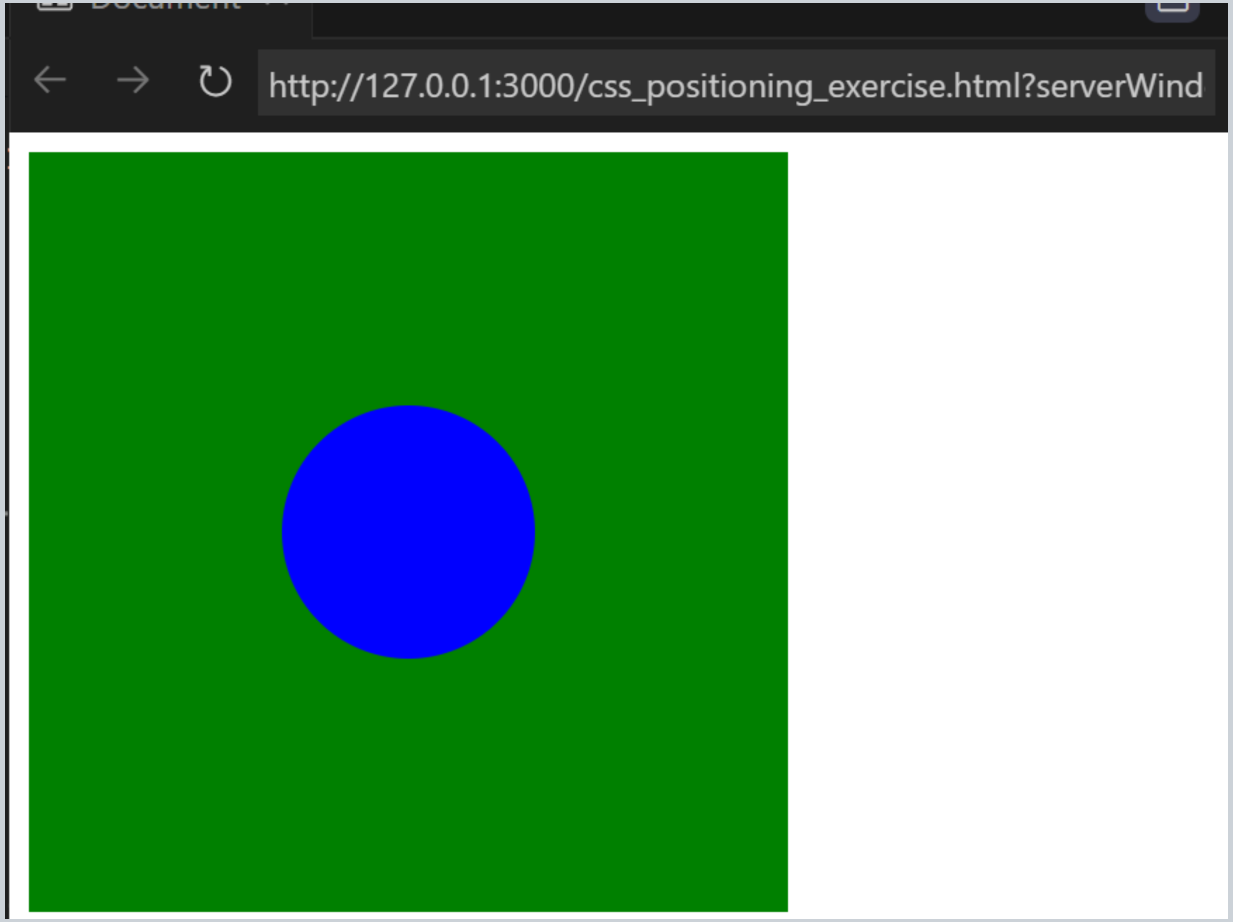

Document
http://127.0.0.1:3000/CSS_Positioning_static.html?serverWindowId

CSS Positioning – important link

- An amazing online webpage to understand the CSS positioning
- https://appbrewery.github.io/css-positioning/
- Credit: Angela Yu & App Brewery (Udemy Course)

CSS Positioning – Practice

- **Practice Problems:**



http://127.0.0.1:3000/css_positioning_exercise.html?serverWind