

Internet và Công nghệ Web

Chương 3 NGÔN NGỮ CSS

(tiếp theo)





Nội dung

1. Box Model - Mô hình hộp
2. Vị trí: thuộc tính position
3. Thuộc tính display
4. Float và Clear
5. Bài tập



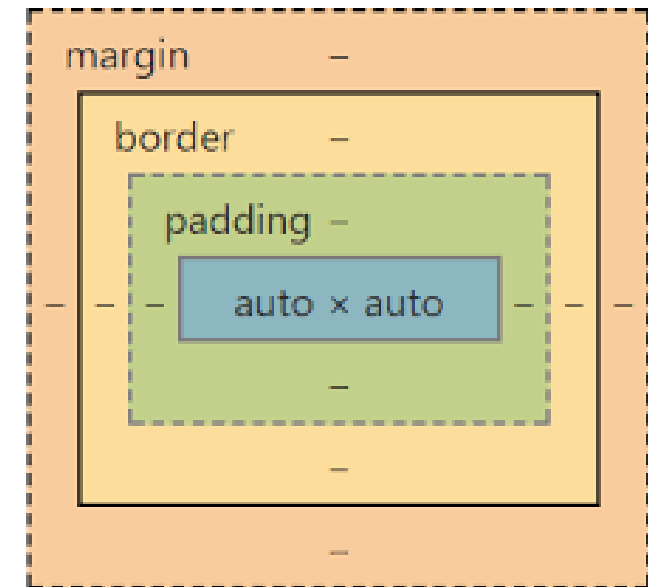
Nhắc lại

CSS Selector	CSS	HTML
Tag name	<pre>h1 { color: red; }</pre>	<code><h1>Công nghệ Thông tin</h1></code>
Class attribute	<pre>.large { font-size: 16pt; }</pre>	<code><p class="large">...</code>
Tag và Class	<pre>p.large { ... }</pre>	<code><p class="large">...</code>
Element id	<pre>#p36 { font-weight: bold; }</pre>	<code><p id="p36">...</code>



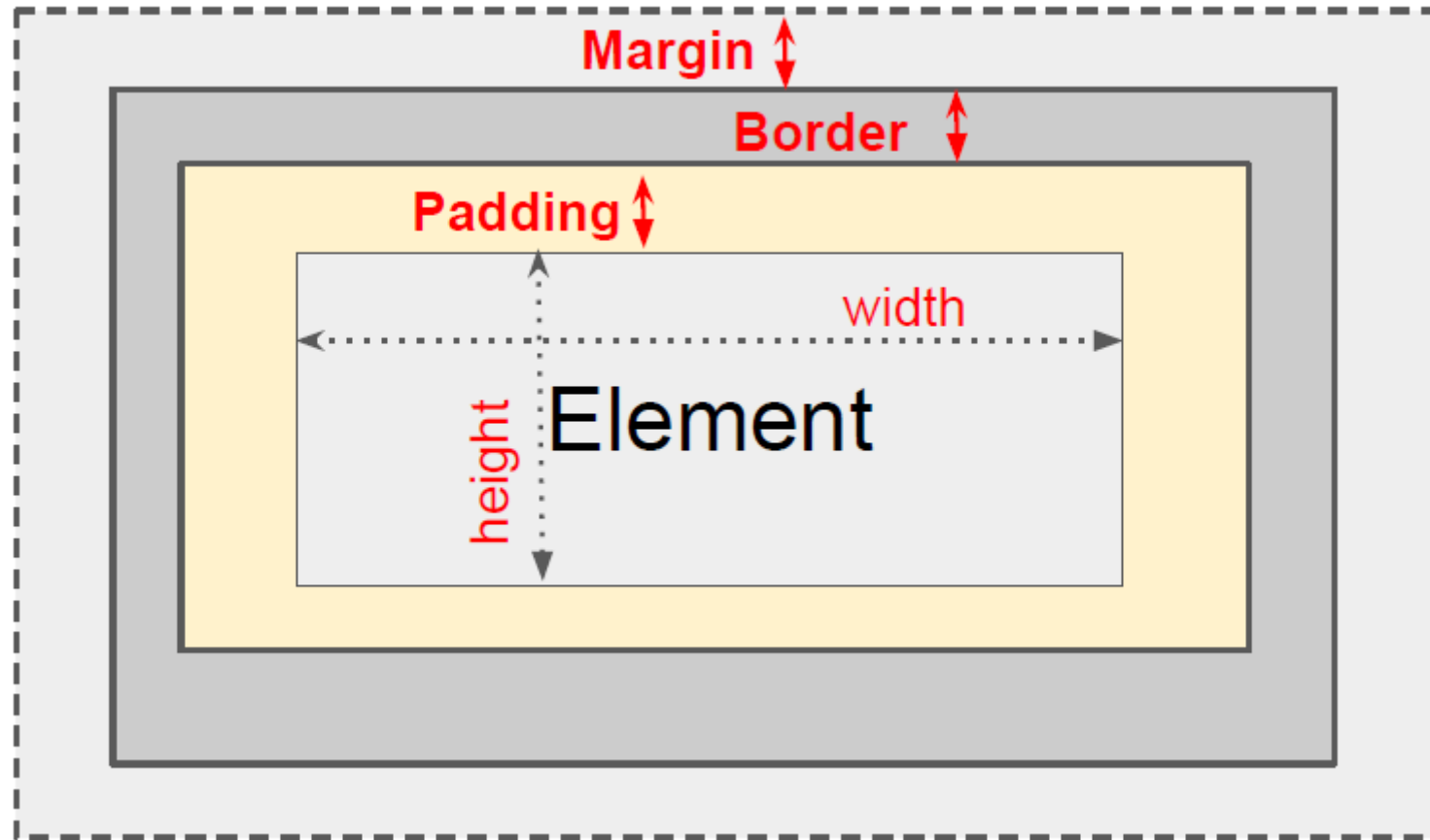
Box Model

- **Box model** quy định cấu trúc cơ bản của mỗi phần tử HTML được hiển thị và kiểm soát về kích thước, vị trí trên trang web.
- Mô hình hộp chia một phần tử HTML thành 4 thành phần chính:
 - Content – Nội dung
 - Padding – Khoảng cách lề bên trong
 - Border – Đường viền
 - Margin – Cách lề
- Kiểm soát kích thước của khối bằng cách điều chỉnh giá trị của các thuộc tính.





Box Model



Total element width = width + left padding + right padding + left border + right border + left margin + right margin



Box Model: Đơn vị

Tuyệt đối - Absolute

- 2px pixels
- 1mm millimeters
- 2cm centimeters
- 0.2in inches
- 3pt printer point 1/72 inch

Dựa vào các phần tử khác - Relative

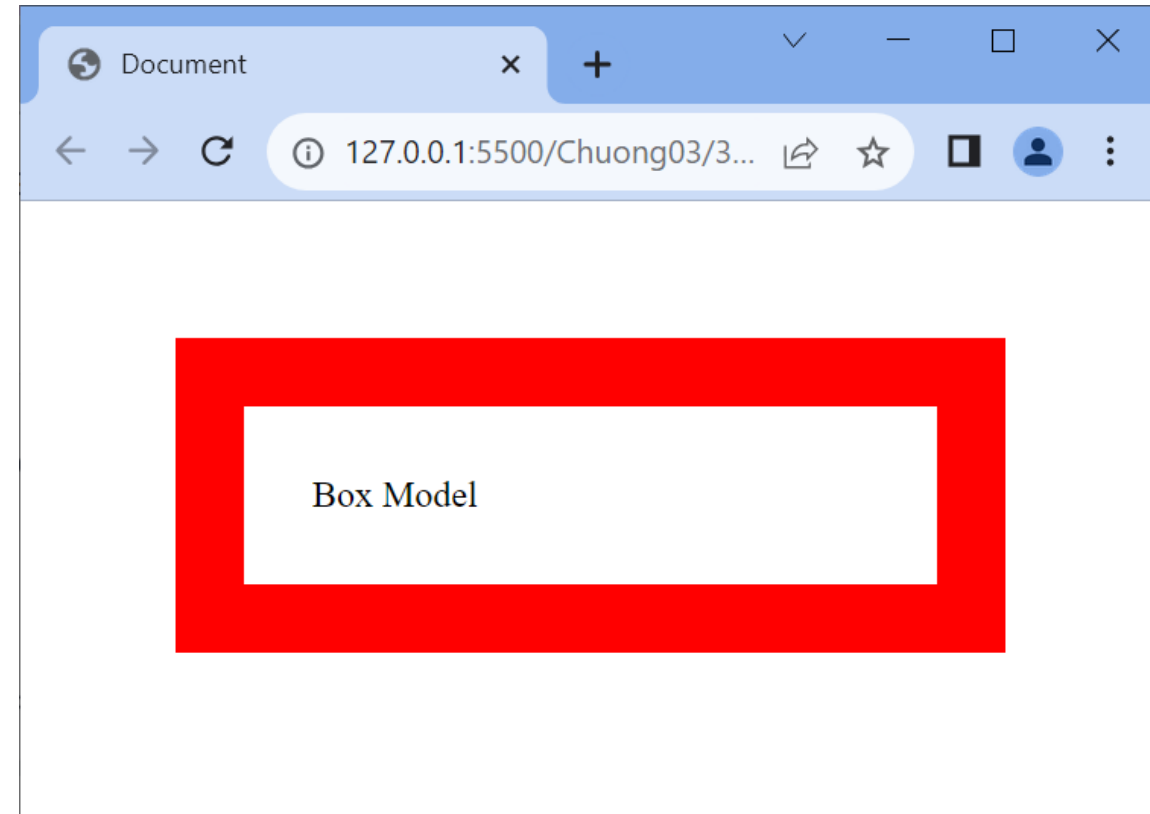
- 2em 2 times the element's current font size
- 3rem 3 times the **root** element's current font size



Box Model

Ví dụ

```
.block0 {  
  padding: 30px;  
  border: 30px solid red;  
  margin: 60px;  
}
```



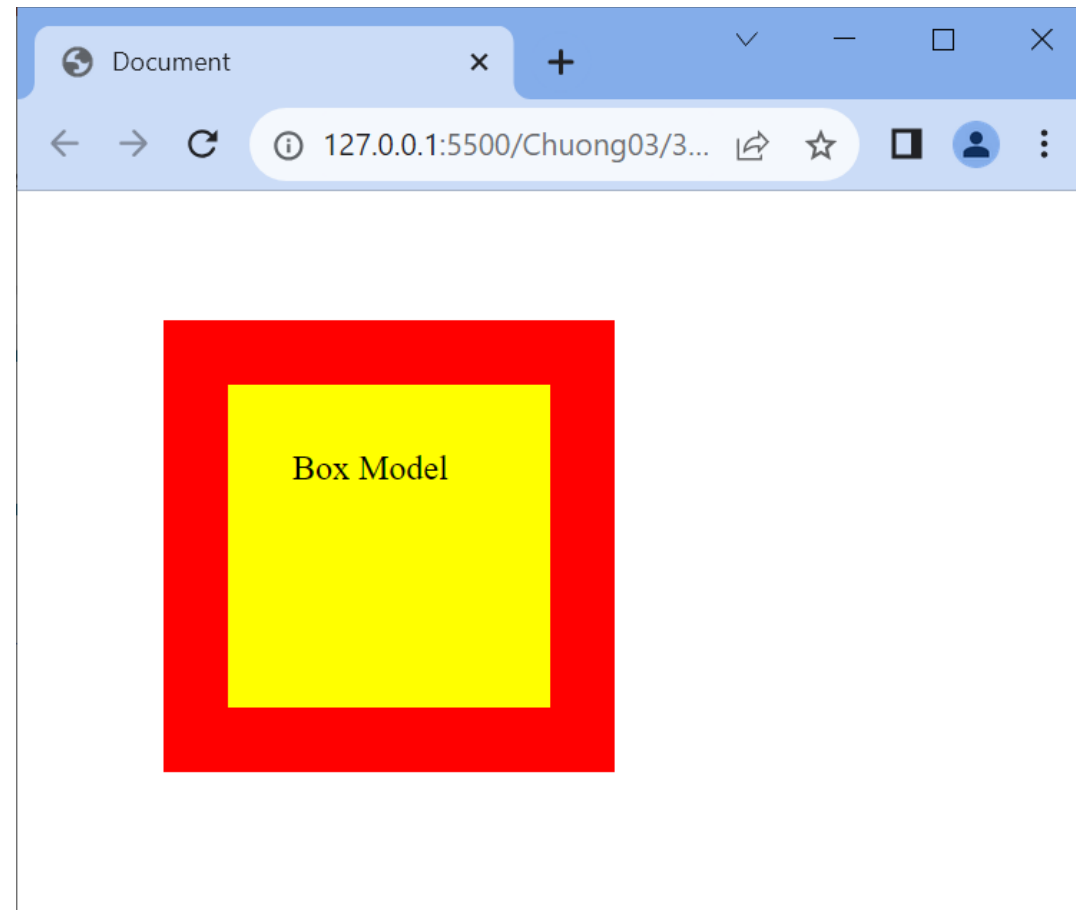


Box Model

Kích thước

- Thuộc tính width, height
- Nếu ko có width thì width lấy theo cha

```
.block01 {  
  width: 90px;  
  height: 90px;  
  background-color: yellow;  
  
  padding: 30px;  
  border: 30px solid red;  
  margin: 60px;  
}
```

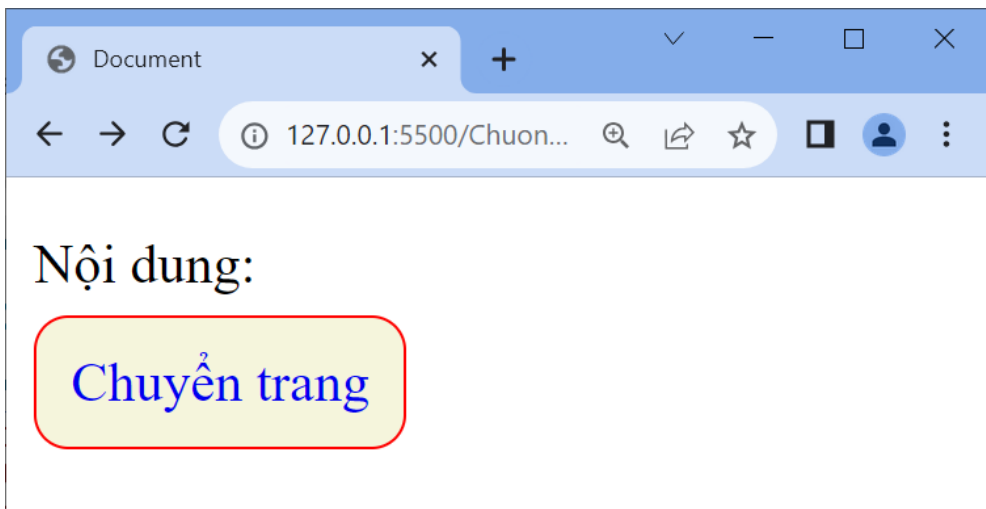




Box Model: Bo tròn góc

Bo tròn góc

- Thuộc tính **border-radius**



```
<!DOCTYPE html>
<html lang="en">
<head>
  <style>
    .block{
      background-color: beige;
      padding: 10px;
      border: 1px solid red;
      text-decoration: none;
      border-radius: 10px;
    }
  </style>
</head>
<body>
  <p>Nội dung: </p>
  <a class="block" href="http://">Chuyển trang</a>
</body>
</html>
```



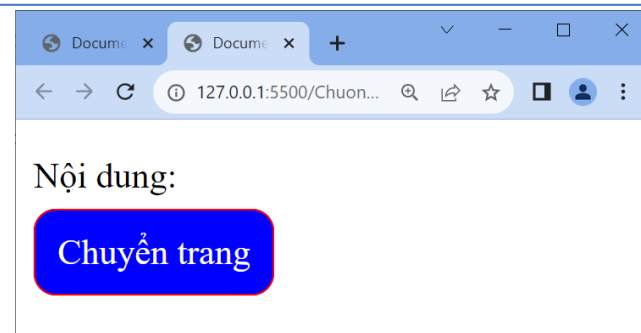
Box Model: Tạo hiệu ứng

Hiệu ứng

- Thuộc tính **hover** để tạo hiệu ứng tương tác khi con trỏ chuột nằm trên một phần tử HTML.
- Ví dụ

```
p:hover, a:hover {  
    background-color: yellow;  
}
```

```
<style>  
    .block{  
        background-color: beige;  
        padding: 10px;  
        border: 1px solid red;  
        text-decoration: none;  
        border-radius: 10px;  
    }  
    .block:hover{  
        background-color: blue;  
        color: white;  
    }  
</style>
```





Thuộc tính position: static, relative, absolute, fixed

Vị trí

- Thuộc tính **position** kiểm soát **vị trí** của khối.
- Các giá trị thường được sử dụng
 1. **static**
 2. **relative**
 3. **absolute**
 4. **fixed**

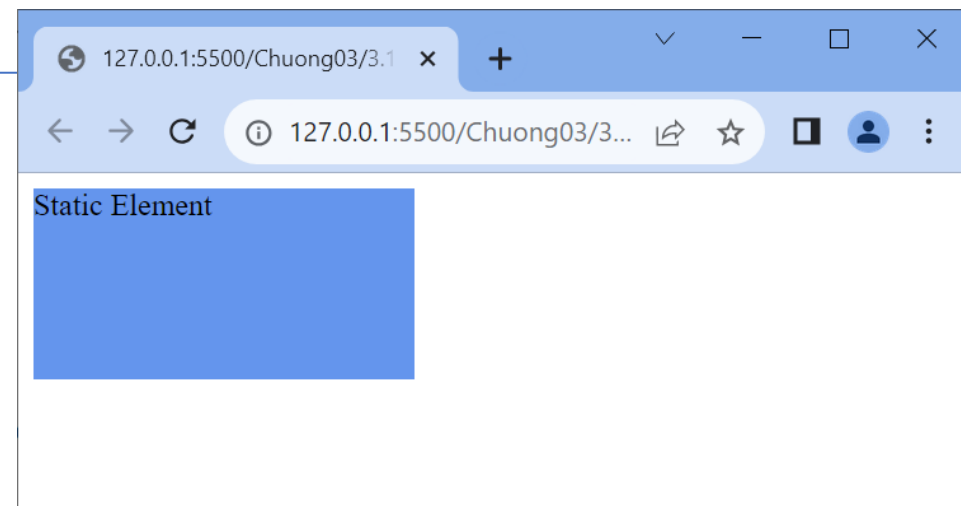


position: static, relative, absolute, fixed

position: static

- **static**: Giá trị mặc định cho position, và không bị ảnh hưởng bởi các thuộc tính top, right, bottom, và left.

```
<head>
  <style>
    .static-element {
      position: static;
      width: 200px;
      height: 100px;
      background-color: cornflowerblue;
    }
  </style>
</head>
<body>
  <div class="static-element">Static Element</div>
</body>
```





position: static, relative, absolute, fixed

position: relative

- **relative**: đặt vị trí dựa trên vị trí ban đầu, sử dụng top, right, bottom, và left để điều chỉnh vị trí của phần tử so với vị trí ban đầu

```
.relative-container {  
  position: relative;  
  width: 300px;  
  height: 200px;  
  background-color: cadetblue;  
  top: 50px;  
  left: 50px;  
}
```

```
.relative-element {  
  position: relative;  
  top: 20px;  
  left: 20px;  
  width: 200px;  
  height: 100px;  
  background-color: white;  
}
```

```
<div class="relative-container">  
  <div class="relative-element">Relative Element</div>  
  <div class="relative-element">Relative Element 2</div>  
</div>
```

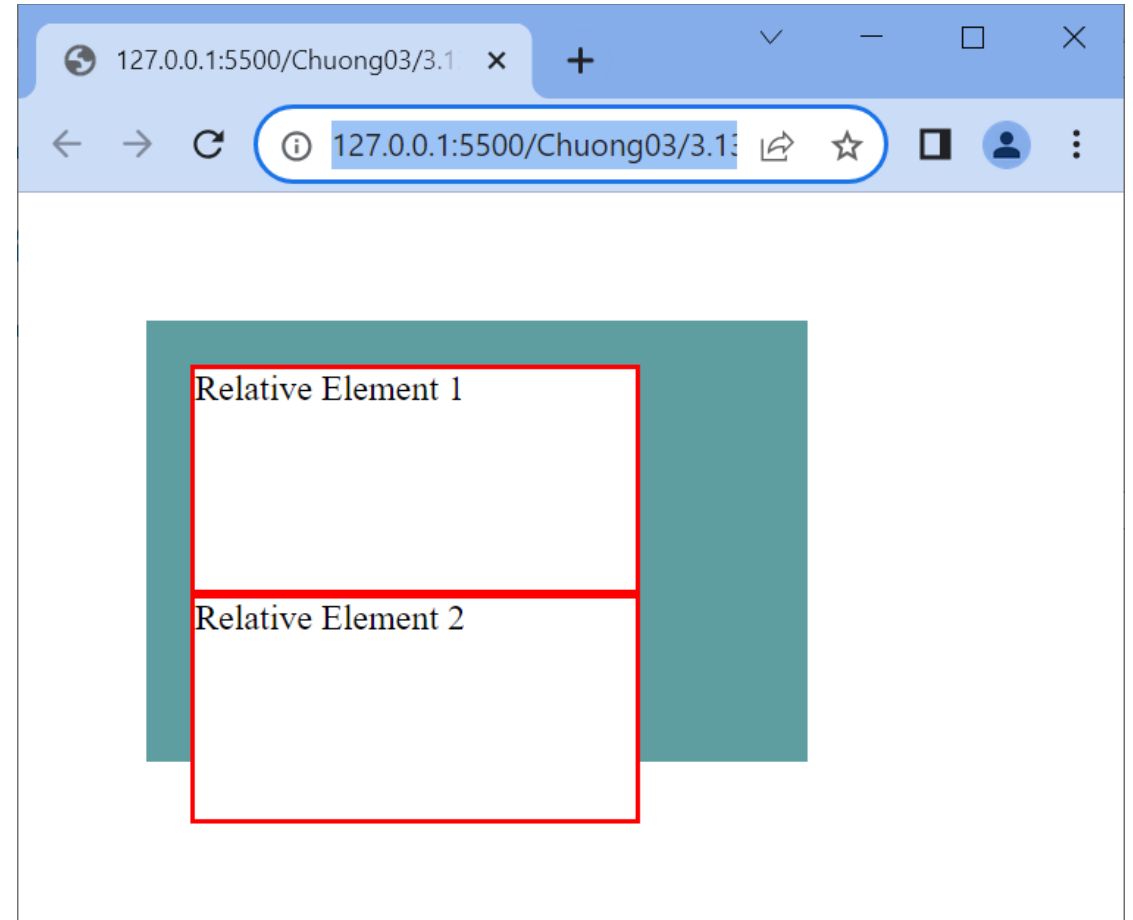


position: static, relative, absolute, fixed

position: relative

```
.relative-container {  
  position: relative;  
  width: 300px;  
  height: 200px;  
  background-color: cadetblue;  
  top: 50px;  
  left: 50px;  
}
```

```
.relative-element {  
  position: relative;  
  top: 20px;  
  left: 20px;  
  width: 200px;  
  height: 100px;  
  background-color: white;  
}
```





position: static, relative, absolute, fixed

position: absolute

- Di chuyển phần tử tuyệt đối so với vị trí của phần tử cha gần nhất không phải là static.

```
.absolute-container {  
  position: relative;  
  top: 50px;  
  left: 100px;  
  width: 400px;  
  height: 200px;  
  background-color: cornflowerblue;  
}
```

```
.absolute-element {  
  position: absolute;  
  top: 50px;  
  left: 100px;  
  width: 200px;  
  height: 100px;  
  background-color: wheat;  
}
```

```
<div class="absolute-container">  
  <div class="absolute-element">Absolute Element 1</div>  
  <div class="absolute-element">Absolute Element 2</div>  
</div>
```

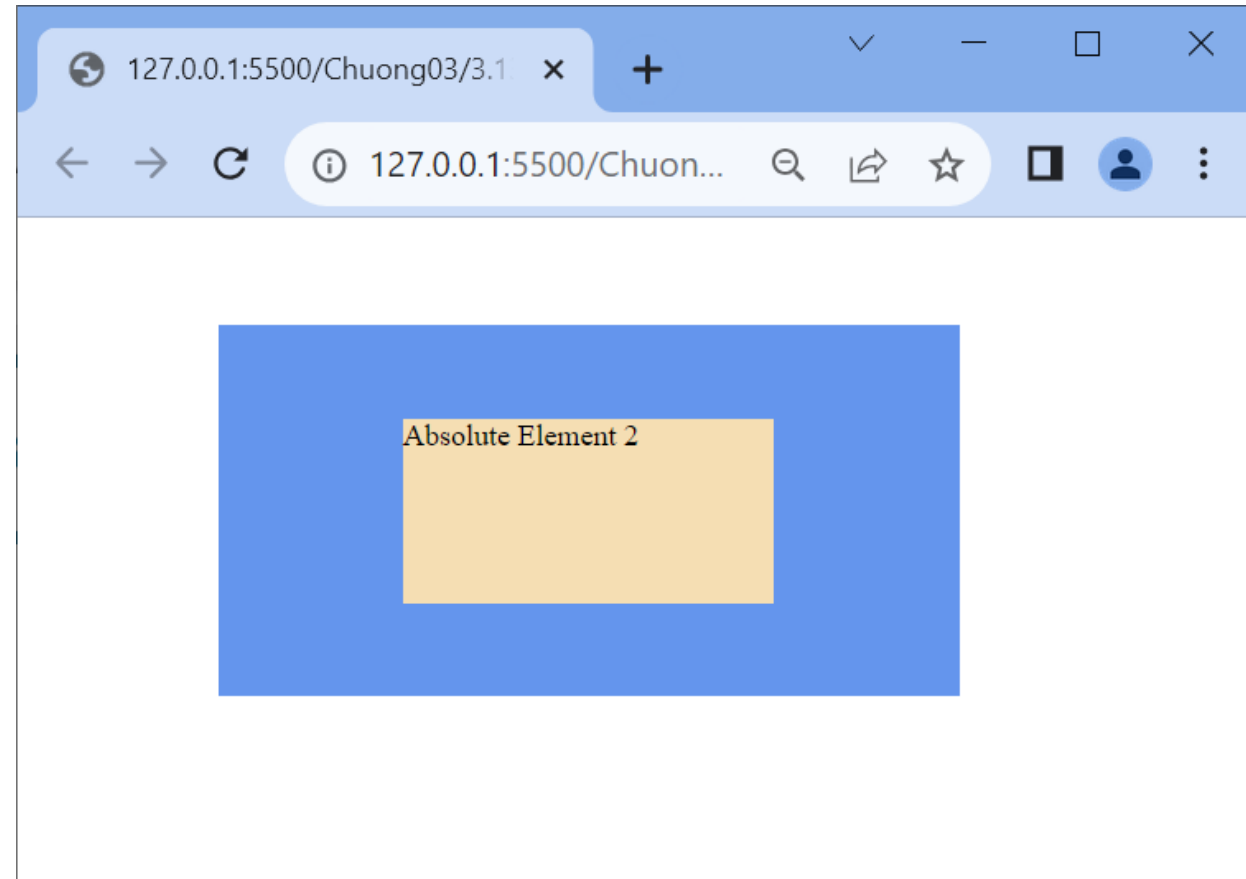


position: static, relative, absolute, fixed

position: absolute

```
.absolute-container {  
  position: relative;  
  top: 50px;  
  left: 100px;  
  width: 400px;  
  height: 200px;  
  background-color: cornflowerblue;  
}
```

```
.absolute-element {  
  position: absolute;  
  top: 50px;  
  left: 100px;  
  width: 200px;  
  height: 100px;  
  background-color: wheat;  
}
```





position: static, relative, absolute, fixed

position: fixed

- Cố định ở một vị trí cụ thể trên trang và không bị ảnh hưởng bởi cuộn trang (scrolling) hoặc các phần tử khác.

```
.fixed-element {  
    position: fixed;  
    top: 100px;  
    left: 100px;  
    width: 200px;  
    height: 100px;  
    background-color: yellow;  
    color: red;  
}
```

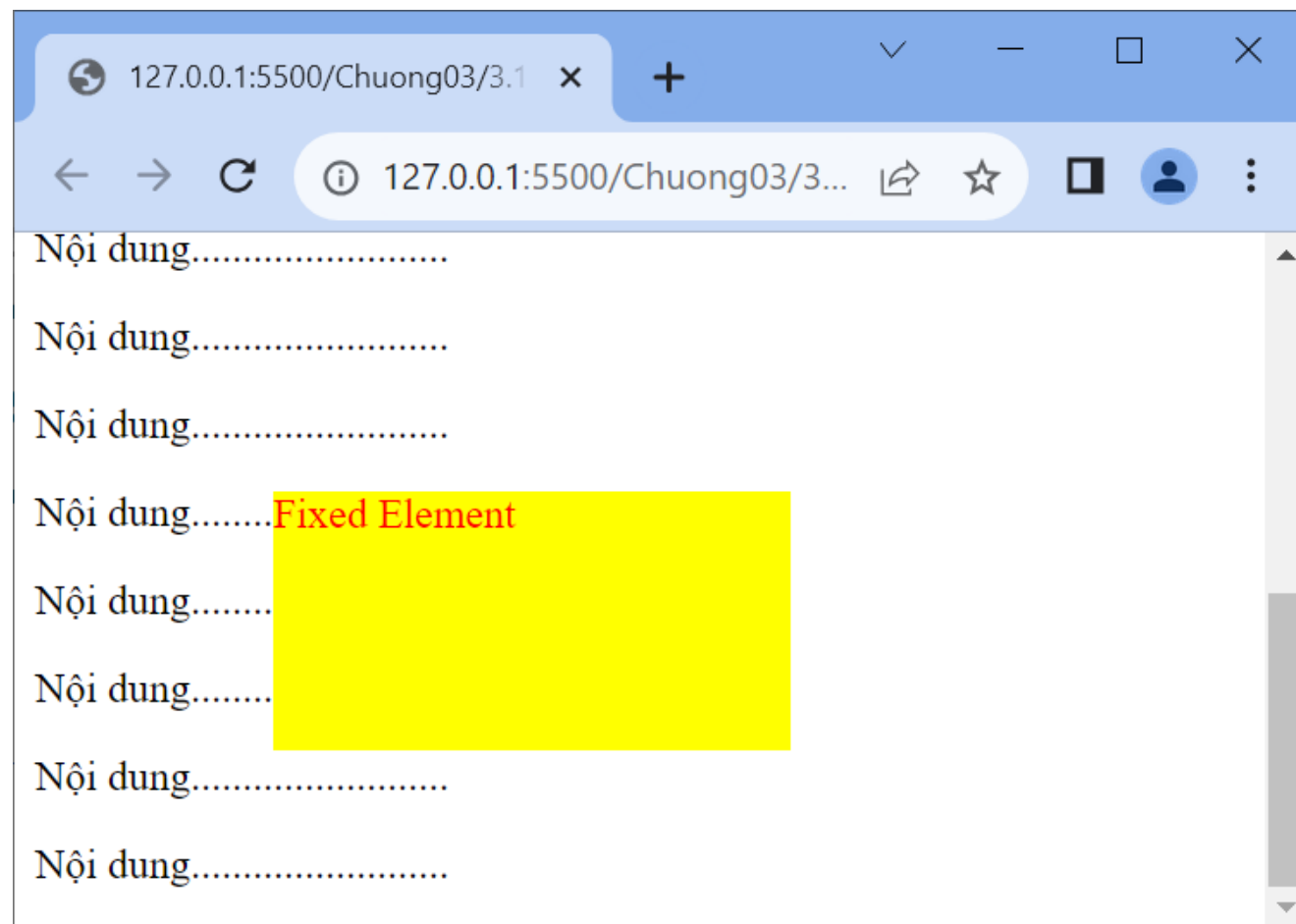
```
<body>  
    <div class="fixed-element">Fixed Element</div>  
    <p>Nội dung.....</p>  
    <p>Nội dung.....</p>  
    <p>Nội dung.....</p>  
    <p>Nội dung.....</p>  
    <p>Nội dung.....</p>  
    <p>Nội dung.....</p>  
    <p>Nội dung.....</p>  
    <p>Nội dung.....</p>  
    <p>Nội dung.....</p>  
    <p>Nội dung.....</p>  
    <p>Nội dung.....</p>  
    <p>Nội dung.....</p>  
    <p>Nội dung.....</p>  
    <p>Nội dung.....</p>  
    <p>Nội dung.....</p>  
    <p>Nội dung.....</p>  
</body>
```



position: static, relative, absolute, fixed

position: fixed

```
.fixed-element {  
    position: fixed;  
    top: 100px;  
    left: 100px;  
    width: 200px;  
    height: 100px;  
    background-color: yellow;  
    color: red;  
}
```





Thuộc tính display

Display Property - Thuộc tính hiển thị

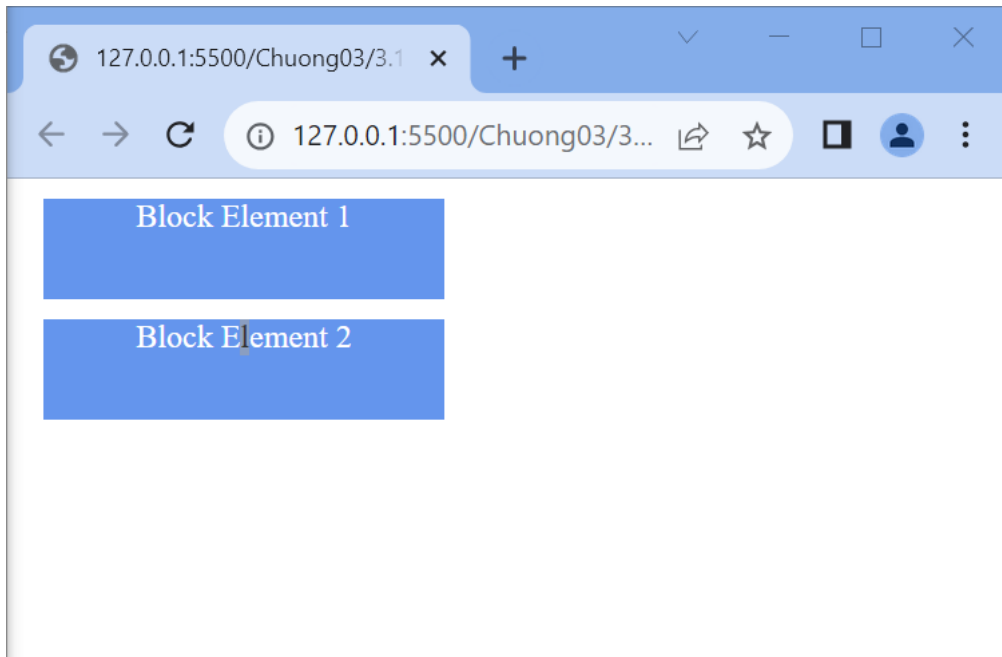
- Thuộc tính **display** cho phép kiểm soát cách phần tử hiển thị trên trang.
- Các giá trị phổ biến bao gồm
 1. block
 2. inline
 3. inline-block
 4. flex



display: block, inline, inline-block, flex

display: block

- Hiển thị trên một dòng riêng biệt và chiếm toàn bộ chiều rộng của phần tử cha



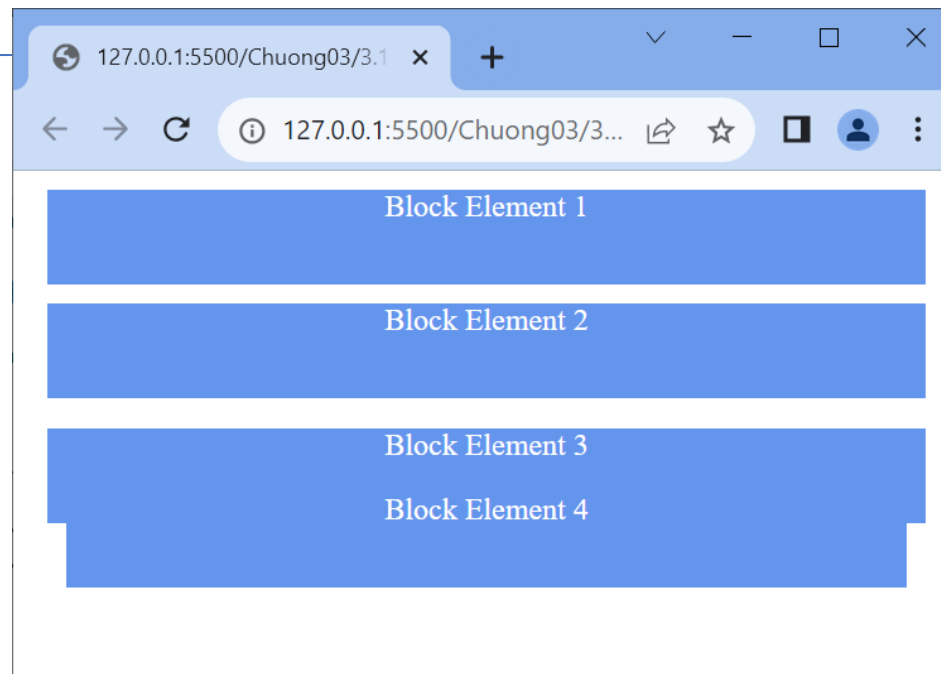
```
<!DOCTYPE html>
<html>
<head>
  <style>
    .block-element {
      display: block;
      width: 200px;
      height: 50px;
      background-color: cornflowerblue;
      color: white;
      text-align: center;
      margin: 10px;
    }
  </style>
</head>
<body>
  <div class="block-element">Block Element 1</div>
  <div class="block-element">Block Element 2</div>
</body>
</html>
```



display: block, inline, inline-block, flex

display: block

```
<body>
  <div class="block-element">Block Element 1</div>
  <div class="block-element">Block Element 2</div>
  <div class="block-element">
    <p>Block Element 3</p>
    <div class="block-element">Block Element 4</div>
  </div>
</body>
```

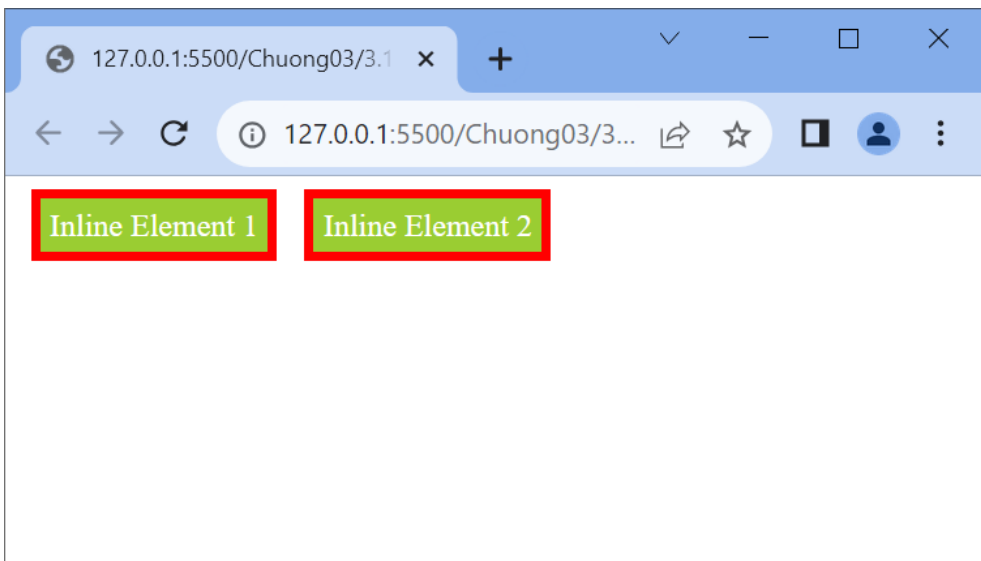




display: block, inline, inline-block, flex

display: inline

- Hiện thị trên cùng một dòng với văn bản xung quanh.
- Rộng cao: mặc định



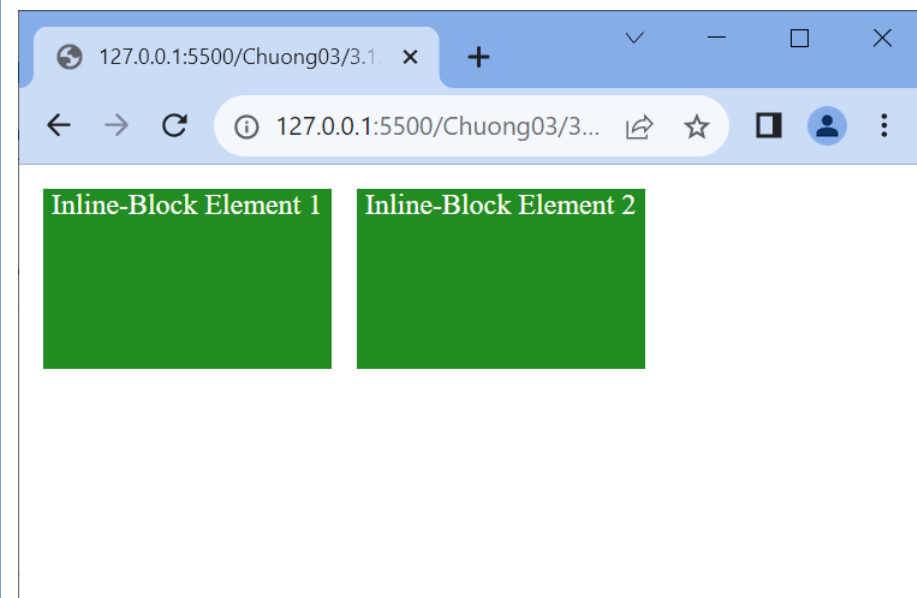
```
<!DOCTYPE html>
<html>
<head>
  <style>
    .inline-element {
      display: inline;
      background-color: yellowgreen;
      color: white;
      padding: 5px 5px;
      margin: 5px;
      border: solid 5px red;
    }
  </style>
</head>
<body>
  <p></p>
  <div class="inline-element">Inline Element 1</div>
  <div class="inline-element">Inline Element 2</div>
</body>
</html>
```



display: block, inline, inline-block, flex

display: inline-block

- Hiển thị như một phần tử inline nhưng vẫn có thể có chiều rộng và chiều cao được điều chỉnh như block.



```
<!DOCTYPE html>
<html>
<head>
  <style>
    .inline-block-element {
      display: inline-block;
      width: 160px;
      height: 100px;
      background-color: forestgreen;
      color: white;
      text-align: center;
      margin: 5px;
    }
  </style>
</head>
<body>
  <div class="inline-block-element">Inline-Block Element 1</div>
  <div class="inline-block-element">Inline-Block Element 2</div>
</body>
</html>
```



display: block, inline, inline-block, flex

display: inline-block => Ứng dụng:

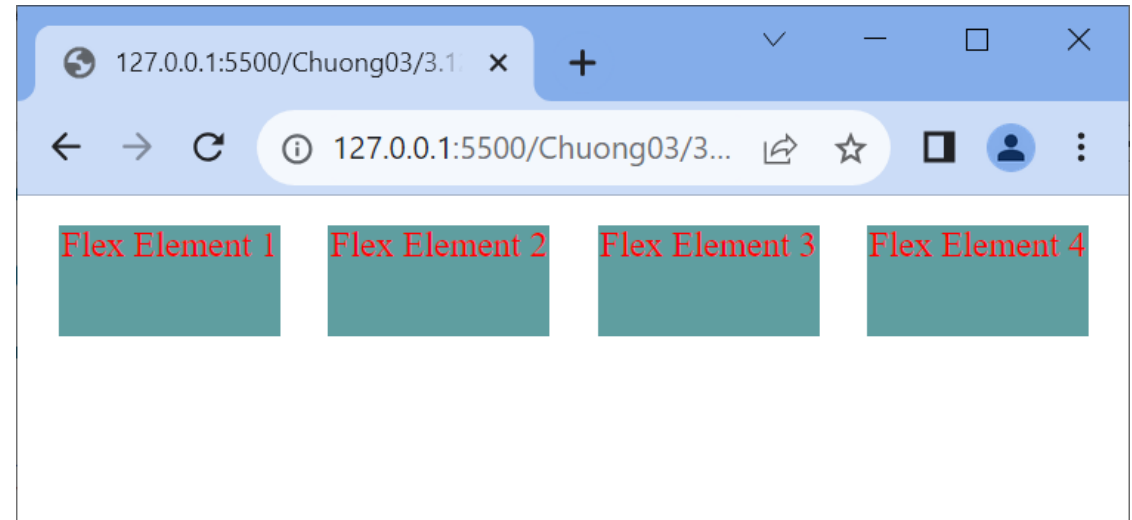
1. Button hoặc liên kết inline-block
2. Hiển thị danh sách menu ngang
3. Hiển thị hộp chứa nội dung inline-block



Box Model – display: block, inline, inline-block, flex

display: flex

- Thuộc tính display: flex trong CSS tạo ra một container chứa các phần tử con.
- Cho phép bạn điều chỉnh và căn chỉnh các phần tử con bên trong một container một cách linh hoạt và dễ dàng.





Box Model – display: block, inline, inline-block, flex

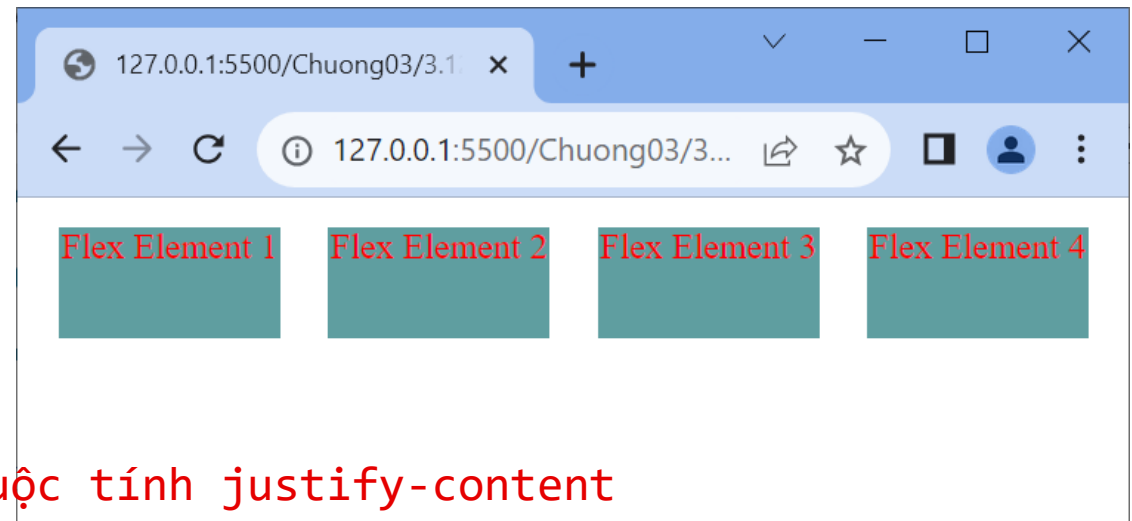
display: flex

```
<style>
  .flex-container {
    display: flex;
    justify-content: space-around;
    /*Căn chỉnh cách đều ele con*/
  }

  .flex-element {
    width: 100px;
    height: 50px;
    background-color: cadetblue;
    color: red;
    text-align: center;
    margin: 5px;
  }
</style>
```

Tìm hiểu thêm các giá trị của thuộc tính justify-content

```
<div class="flex-container">
  <div class="flex-element">Flex Element 1</div>
  <div class="flex-element">Flex Element 2</div>
  <div class="flex-element">Flex Element 3</div>
  <div class="flex-element">Flex Element 4</div>
</div>
```





Float – Lề

Float – Lề

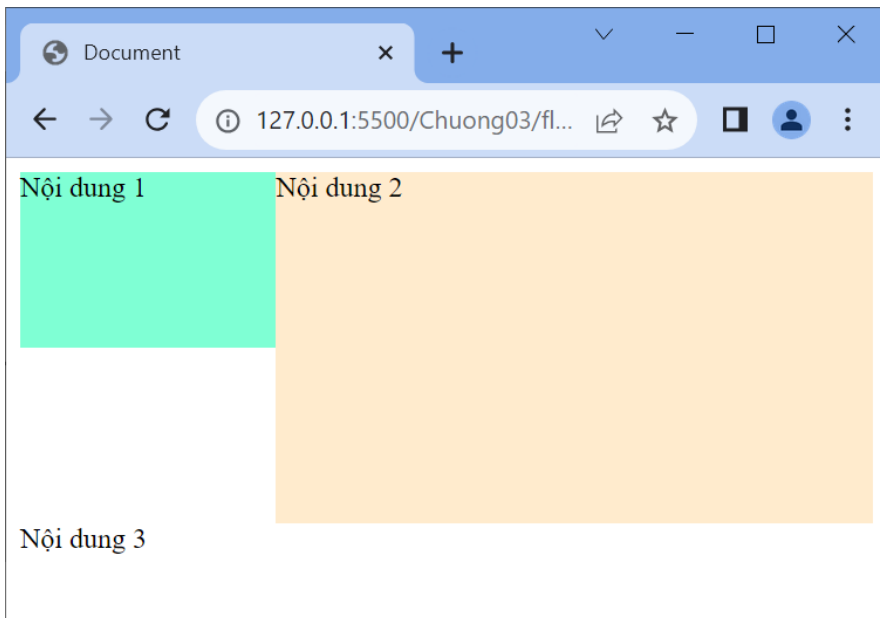
- Thuộc tính **float** dùng để căn lề các khối theo hướng **trái** hoặc **phải**.
- Sử dụng để tạo cấu trúc nhiều cột.
- Có hai giá trị chính cho thuộc tính float:
 - **left**: Phần tử sẽ đặt bên **trái** của phần tử cha và các phần tử khác sẽ căn lề xung quanh phía bên phải.
 - **right**: Phần tử sẽ được đặt bên **phải** của phần tử cha và các phần tử khác sẽ căn lề xung quanh phía bên trái.



Float – Lề

Minh họa

```
<body>
  <div class="left-column">Nội dung 1</div>
  <div class="right-column">Nội dung 2</div>
  <div class="clear">Nội dung 3</div>
</body>
```



```
.left-column {
  float: left;
  width: 30%; /* Độ rộng của cột trái */
  height: 100px;
  background-color: aquamarine;
}

.right-column {
  float: right;
  width: 70%; /* Độ rộng của cột phải */
  height: 200px;
  background-color: blanchedalmond;
}

/* Xóa các float tránh ảnh hưởng nội dung bên dưới */
.clear{
  clear: both;
}
```



Bài tập Buổi 5

- *GV cho trên lớp*



Layout

Giới thiệu

- Tổ chức, sắp xếp phần tử HTML để tạo nên cấu trúc và giao diện tổng thể cho trang web hoặc ứng dụng.
- Layout đóng vai trò quan trọng trong thiết kế web:
 - Thể hiện phương pháp trình bày thông tin trang web.
 - Giao diện người dùng được tạo ra.
- Các loại:
 - Normal Flow
 - Float
 - Positioning
 - Flexbox
 - CSS Grid



Layout: Normal Flow

- Trình bày từ trên xuống dưới và từ trái qua phải



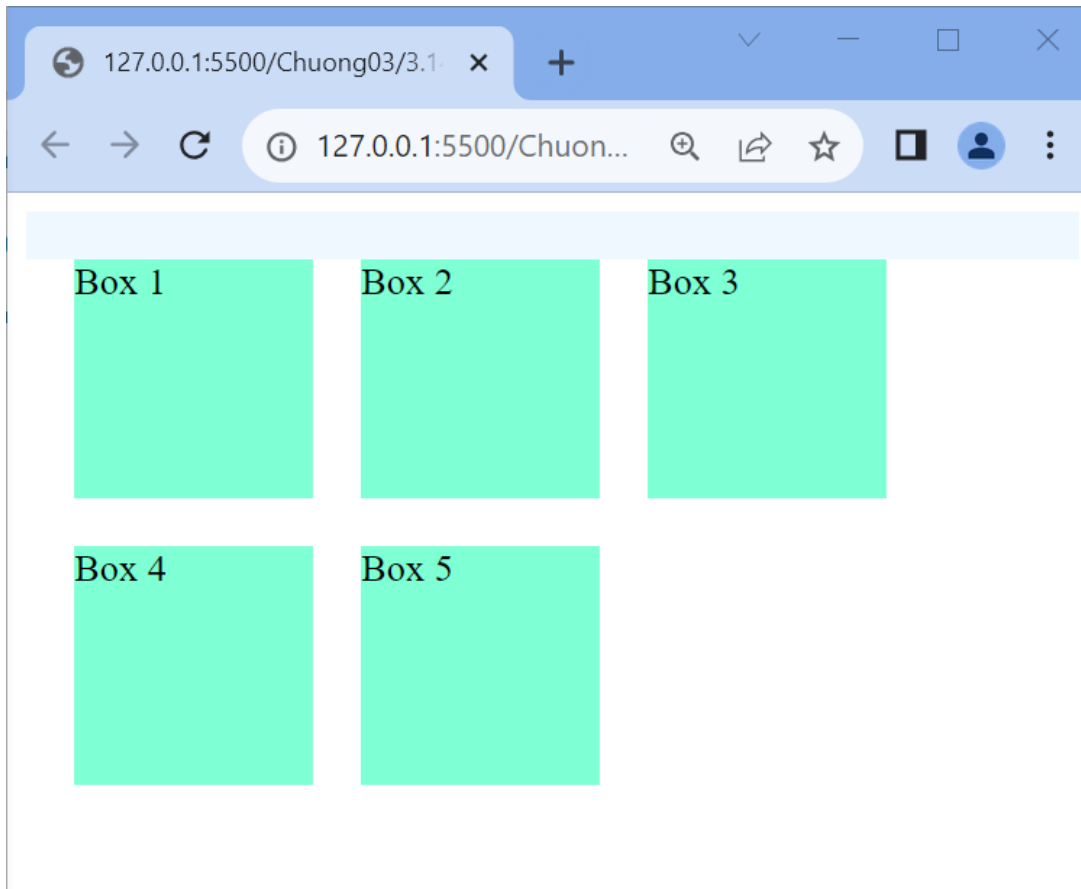
```
.container {  
    border: 1px solid royalblue;  
    padding: 10px;  
}  
  
.box {  
    /* width: 100px; */  
    height: 100px;  
    background-color: chartreuse;  
    margin: 10px;  
}
```

```
<div class="container">  
    <div class="box">Box 1</div>  
    <div class="box">Box 2</div>  
    <div class="box">Box 3</div>  
    <div class="box">Box 4</div>  
    <div class="box">Box 5</div>  
</div>
```



Layout: Normal Flow

- Trình bày từ trên xuống dưới và từ trái qua phải



```
.container {  
    background-color: aliceblue;  
    padding: 10px;  
}  
.box {  
    width: 100px;  
    height: 100px;  
    background-color: aquamarine;  
    margin: 10px;  
    float: left;  
}
```

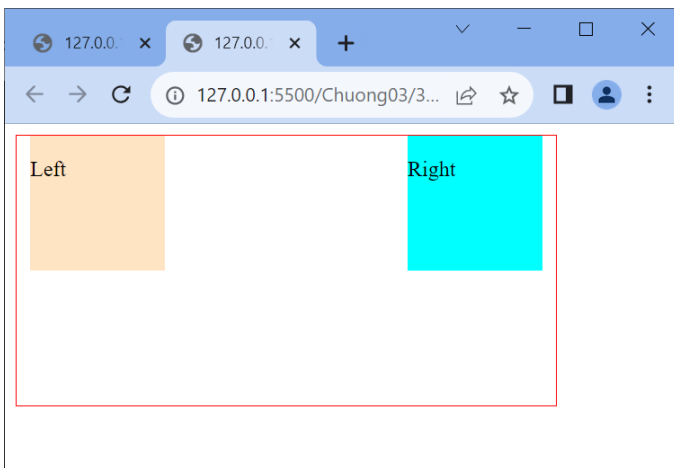
```
<div class="container">  
    <div class="box">Box 1</div>  
    <div class="box">Box 2</div>  
    <div class="box">Box 3</div>  
    <div class="box">Box 4</div>  
    <div class="box">Box 5</div>  
</div>
```




Layout: Float

- Bố cục theo canh lề trái-phải

```
<div class="container">
  <div class="left-column">
    <p>Left</p>
  </div>
  <div class="right-column">
    <p>Right</p>
  </div>
</div>
```



```
.container {
  width: 400px;
  height: 200px;
  border: 1px solid red;
}
.left-column {
  width: 100px;
  height: 100px;
  float: left;
  margin-left: 10px;
  background-color: bisque;
}
.right-column {
  width: 100px;
  height: 100px;
  float: right;
  margin-right: 10px;
  background-color: aqua;
}
```

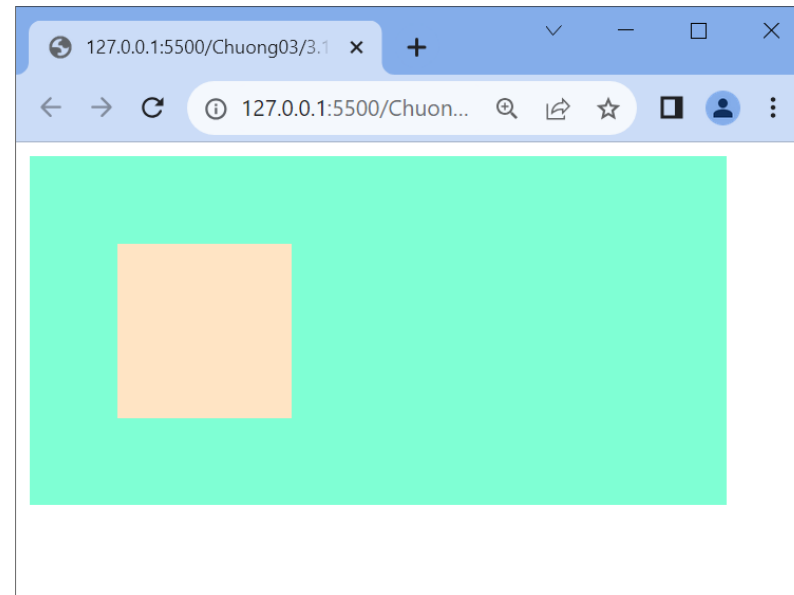


Layout: Positioning

- Đặt vị trí của các phần tử cụ thể hoặc tạo các phần tử cố định.

```
.container {  
  width: 400px;  
  height: 200px;  
  background-color: aquamarine;  
  position: relative;  
}  
.box {  
  width: 100px;  
  height: 100px;  
  background-color: bisque;  
  position: absolute;  
  top: 50px;  
  left: 50px;  
}
```

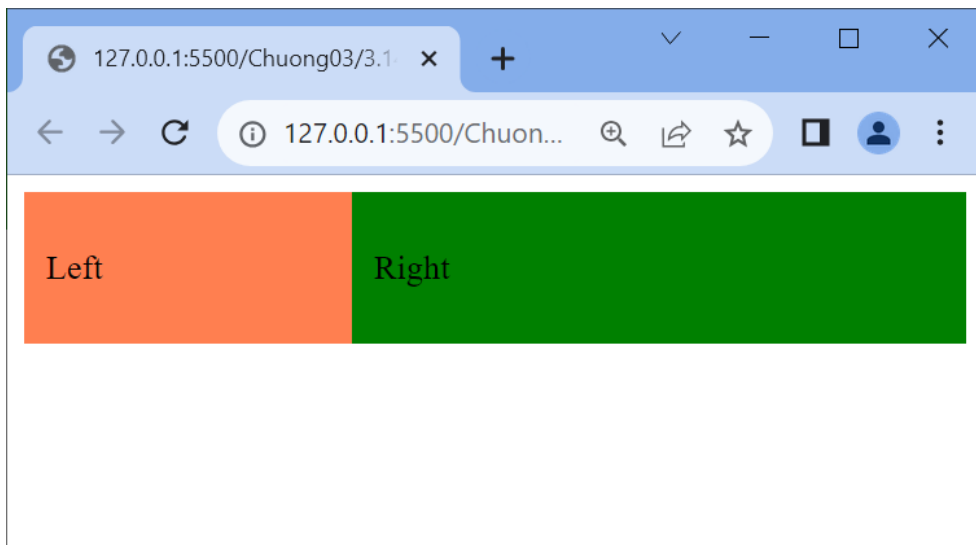
```
<div class="container">  
  <div class="box"></div>  
</div>
```





Layout: Flexbox

- Flexbox: sắp xếp các phần tử trên một trục hoặc nhiều trục



```
.container {  
  display: flex;  
}  
.left-column {  
  flex: 1;  
  background-color: coral;  
  padding: 10px;  
}  
.right-column {  
  flex: 2;  
  background-color: green;  
  padding: 10px;  
}
```

```
<div class="container">  
  <div class="left-column">  
    <p>Left</p>  
  </div>  
  <div class="right-column">  
    <p>Right</p>  
  </div>  
</div>
```

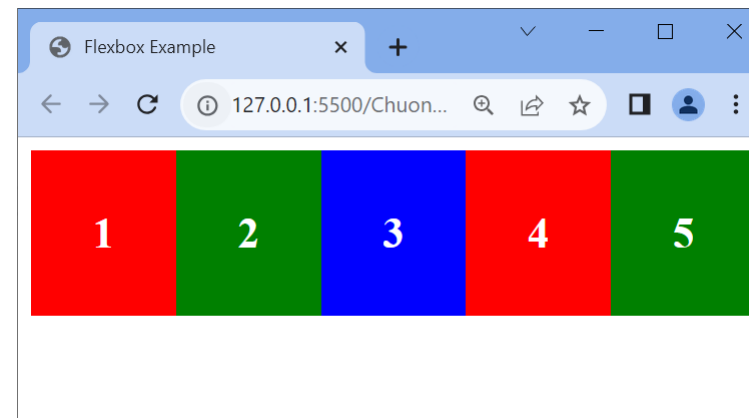


Layout: Flexbox

```
.container {  
  display: flex;  
}  
.box {  
  width: 100px;  
  height: 100px;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  font-size: 26px;  
  font-weight: bold;  
}
```

```
.red {  
  background-color: red;  
  color: white;  
}  
.green {  
  background-color: green;  
  color: white;  
}  
.blue {  
  background-color: blue;  
  color: white;  
}
```

```
<div class="container">  
  <div class="box red">1</div>  
  <div class="box green">2</div>  
  <div class="box blue">3</div>  
  <div class="box red">4</div>  
  <div class="box green">5</div>  
</div>
```





Layout: CSS Grid

- Kiểm soát các phần tử HTML được sắp xếp trong một lưới (grid) hai chiều: hàng và cột.
- Xác định số hàng và cột trong lưới bằng thuộc tính
 - `grid-template-rows`
 - `grid-template-columns`
- Đơn vị: pixel (px), phần trăm (%), hoặc tỷ lệ (fr - fractional unit).

```
.grid-container {  
  display: grid;  
  grid-template-rows: 100px 200px 1fr;  
}
```

```
.grid-container {  
  display: grid;  
  grid-template-rows: 100px 1fr 2fr;  
}
```

- 2 dòng đầu tiên có kích thước cố định là 100px và 200px.
- 1fr (fr là fractional unit) dòng cuối sử dụng phần còn lại của không gian trong Grid Container.

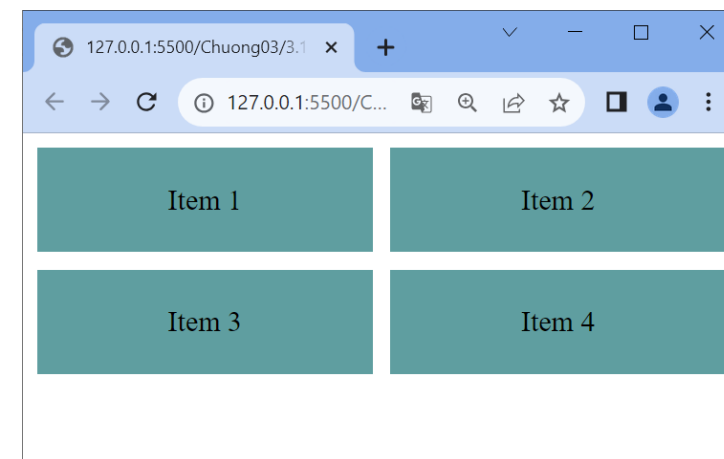
- Dòng đầu có chiều cao 100px
- Dòng 2 chiếm phần tương đối (fr)
- Dòng 3 chiếm gấp đôi phần tương đối so dòng 2



Layout: CSS Grid

- ```
.grid-container {
 display: grid;
 grid-template-columns: 1fr 1fr; /* Chia thành hai cột bằng tỷ lệ 1:1 */
 gap: 10px; /* Khoảng cách giữa các phần tử */
}
.grid-item {
 background-color: cadetblue;
 padding: 20px;
 text-align: center;
}
```

```
<div class="grid-container">
 <div class="grid-item">Item 1</div>
 <div class="grid-item">Item 2</div>
 <div class="grid-item">Item 3</div>
 <div class="grid-item">Item 4</div>
</div>
```





# Bài tập Buổi 6



Thank you





# Thảo luận

