

CHƯƠNG 3

CASCADING STYLE SHEET (CSS)

Ths. Phạm Văn Tiệp
Khoa Công nghệ thông tin
Đại học Đại Nam

CSS

Xây dựng



Trang trí nội thất

Con người



Trang điểm

Web



CSS

1. Giới thiệu

CSS là gì?

CSS (Cascading Style Sheets) là một ngôn ngữ quy định cách trình bày cho các tài liệu viết bằng HTML, XHTML, XML, UML,...

1. Giới thiệu

❑ Tại sao CSS?

- ✓ CSS cung cấp nhiều thuộc tính trình bày dành cho các đối tượng với sự sáng tạo trong kết hợp các thuộc tính giúp mang lại hiệu quả cao.
- ✓ CSS đã được hỗ trợ bởi tất cả các trình duyệt và hiển thị “như nhau” trên mọi hệ điều hành.
- ✓ CSS đưa ra phương thức áp dụng từ một file CSS ở ngoài. Có hiệu quả đồng bộ khi tạo một website có hàng trăm trang hay khi muốn thay đổi một thuộc tính trình bày nào đó.
- ✓ CSS được cập nhật liên tục mang lại các trình bày phức tạp và tinh vi hơn.

1. Giới thiệu

Học CSS cần những gì?

- Có là một kiến thức về HTML.
- Một trình soạn thảo văn bản để bạn có thể viết mã CSS như: Notepad trong Windows, Pico trong Linux, Simple Text trong Mac. Hay từ các chương trình DreamWeaver, FrontPage, Golive,...
- Một trình duyệt web.

2. Cú pháp CSS

Gồm 3 thành phần:

Bộ chọn (Selector)

Thuộc tính (Property)

Giá trị (Value)

Cú pháp:

```
Selector {  
    Thuộc tính 1: giá trị;  
    ...  
    Thuộc tính n: giá trị;  
}
```

Dấu chấm phẩy
dùng để phân
cách thuộc tính

2. Cú pháp CSS

- Selector thường là tên thẻ HTML
- Mỗi thuộc tính cần có giá trị
- Một thuộc tính và giá trị phân cách dấu ":"
- Hai cặp thuộc tính – giá trị phân cách nhau bởi dấu ";"
- Toàn bộ các cặp thuộc tính – giá trị của thẻ HTML được đặt trong cặp dấu ngoặc nhọn.

Chú ý:

- Css phân biệt chữ hoa, chữ thường
- Để ghi chú trong style sử dụng:

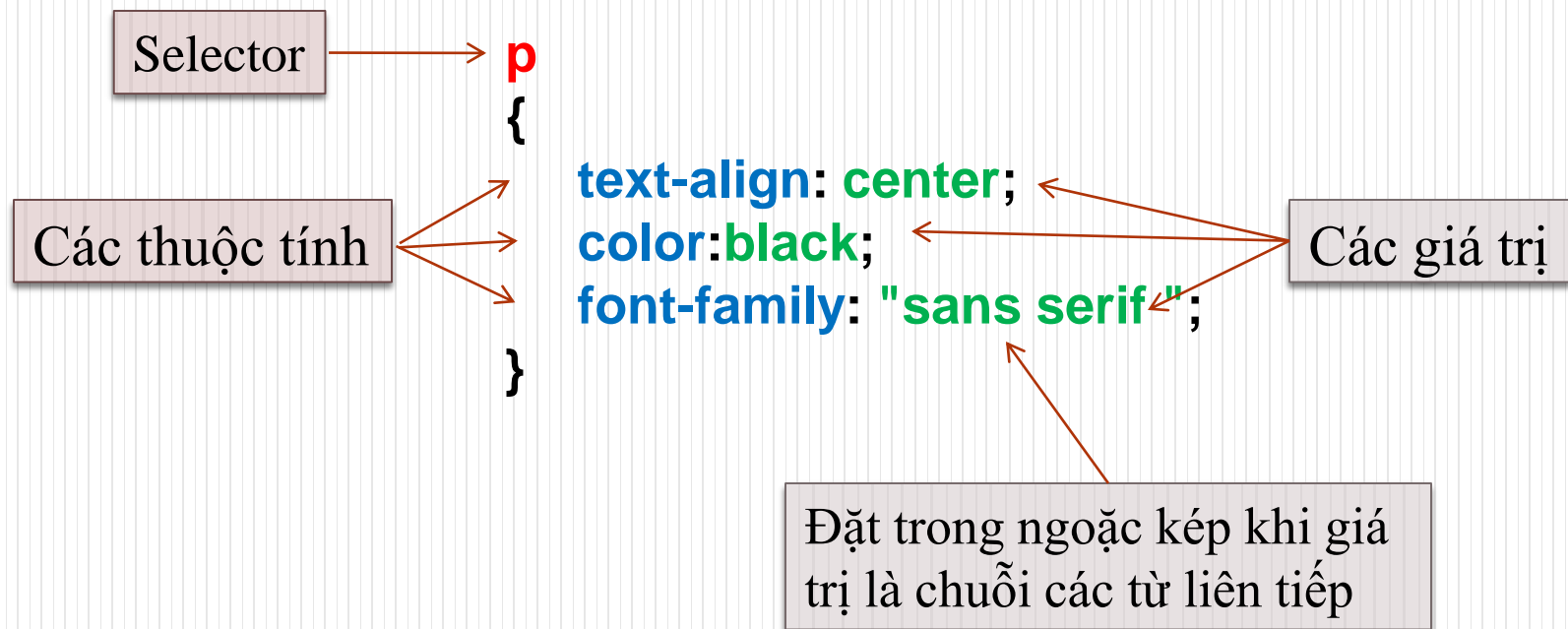
/*

Đoạn ghi chú

*/

2. Cú pháp CSS

❑ Ví dụ:



3. Áp dụng CSS vào trang HTML

□ Ba cách áp dụng CSS trong tài liệu:

- **Inline style**

Sử dụng thuộc tính style của thẻ để định dạng

- **Internal style sheet**

Định nghĩa style bên trong <head> của tài liệu

- **External style sheet**

Liên kết đến một tập tin *.css chứa toàn bộ style sử dụng trong tài liệu

3. Áp dụng CSS vào trang HTML

❑ Cách 1: Inline style

Là cách dùng style ngay trong câu lệnh, bổ sung thêm thuộc tính style vào sau một phần tử HTML.

Cú pháp:

<thẻ style="thuộc tính: giá trị; thuộc tính: giá trị;...">

Nội dung

</thẻ >

Ví dụ:

<b style="color: navy;">Màu xanh nước biển.

3. Áp dụng CSS vào trang HTML

❑ Nhận xét:

- Một inline style áp dụng cho bất cứ thẻ nào và chỉ có tác dụng trên chính thẻ đó!
- Dùng inline style làm cho tài liệu rõ ràng hơn, nhưng cũng có thể dẫn đến **việc viết mã quá nhiều** và thiếu sự nhất quán trên toàn site!
- Làm cho mã nguồn bị **dư thừa**, khó bảo trì!

3. Áp dụng CSS vào trang HTML

❑ Cách 2: Internal style sheet

- Áp dụng thống nhất cho toàn trang web
- Định nghĩa riêng một khối, phân biệt bởi thẻ `<style>` và được đặt trong phần `head` của tài liệu.
- Khối này là một tập các style rule (quy tắc về kiểu dáng), trong đó mỗi quy tắc định nghĩa style cho một phần tử hay nhóm phần tử HTML.

3. Áp dụng CSS vào trang HTML

□ Ví dụ:

```
<head>
```

```
  <style type="text/css">
```

```
    h3 { border: 1px solid blue; color: red; }
```

```
    p { border: 1px solid red; }
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <h3>Đoạn văn bản có chữ màu đỏ, viền xanh</h3>
```

```
  <p>Đoạn văn bản có viền màu đỏ</p>
```

```
</body>
```

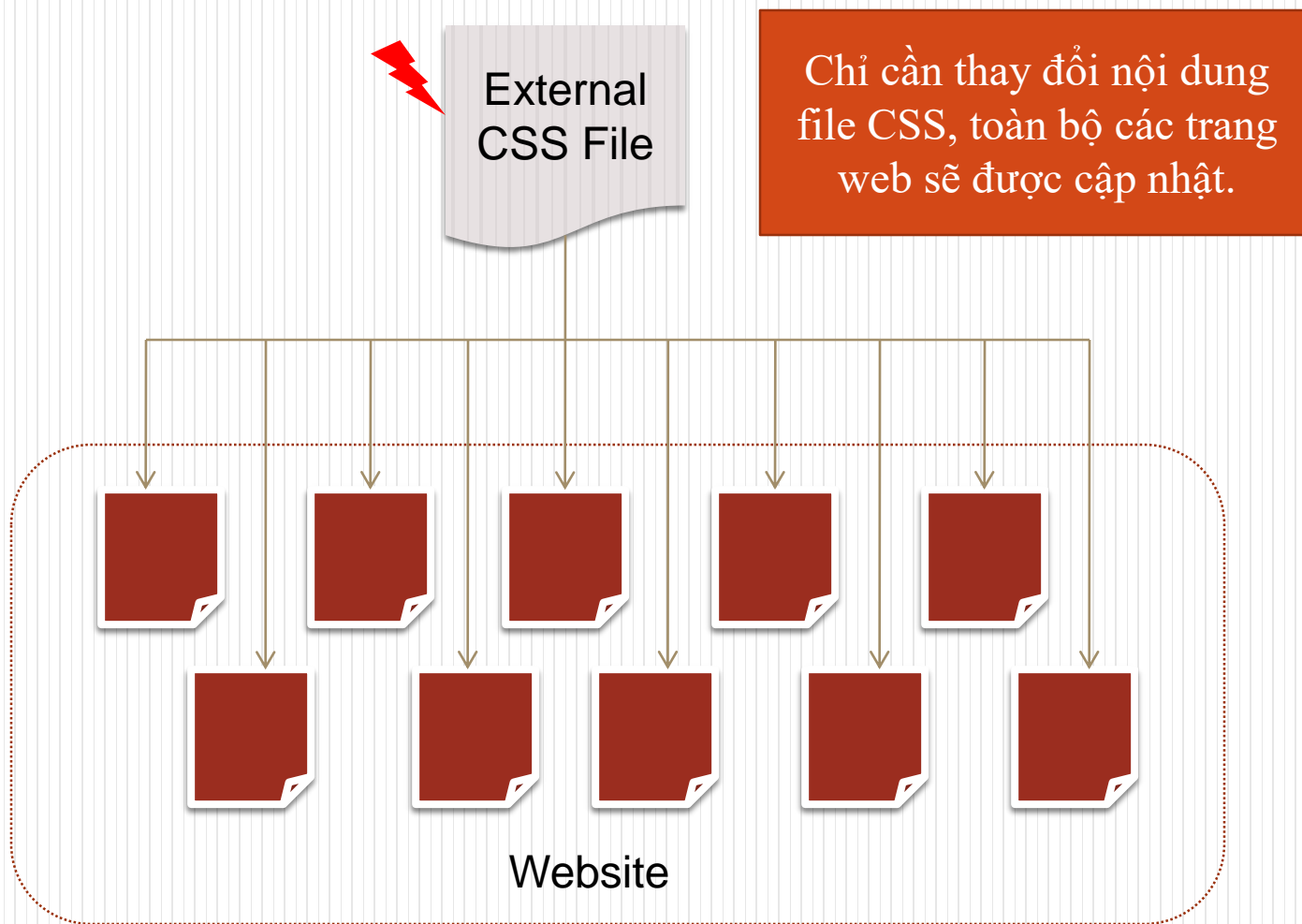
3. Áp dụng CSS vào trang HTML

❑ Cách 3: External style sheet

- Áp dụng cho toàn site
- Tạo một tập tin có phần mở rộng là ***.css** bên ngoài HTML, sử dụng thẻ **<link>** trong phần **<head>** để liên kết tập tin này trong HTML.
- Các style rule chứa riêng biệt trong file ***.css** và hoạt động tương tự như **internal style sheet**.

<link rel="stylesheet" type="text/css" href="mystyle.css"/>

3. Áp dụng CSS vào trang HTML



4. Selector

❑ Các selector trong HTML

1. Universal selector
2. Type selector
3. Identity selector
4. Class selector
5. Descendant selector
6. Child selector
7. Adjacent sibling selector
8. Attribute selector
9. Pseudo class selector
10. Group selector

4.1. Universal selector

❑ Universal là selector yếu nhất trong CSS, Universal selector thường được sử dụng để reset CSS.

Cú pháp:

```
*{  
    Thuộc tính 1: giá trị;  
    ...  
    Thuộc tính n: giá trị;  
}
```

Ví dụ:

Sử dụng Universal selector để reset margin và padding của tất cả các thành phần về 0.

```
* {margin: 0; padding: 0}
```

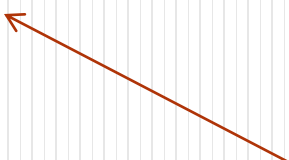
4.2. Type Selector

- Bộ chọn đơn giản nhất, dùng các thẻ HTML

Ví dụ:

Định dạng tất cả các siêu liên kết trong toàn bộ trang web không có đường gạch chân.

```
a
{
    /*Bỏ gạch chân cho các hyperlink*/
    text-decoration: none;
}
```



Comment trong CSS

4.3. ID Selector

- Định danh ID cho phép chia thẻ thành nhiều loại khác nhau
- Định danh chỉ có thể áp dụng duy nhất cho 1 thẻ và tên của định danh phải là duy nhất trên trang web

Cú pháp:

```
[Tên thẻ]#<id của thẻ> {  
    Thuộc tính 1: giá trị ;  
    ...  
    Thuộc tính n: giá trị;  
}
```

4.3. ID Selector

❑ Ví dụ:

```
<html>
  <head>
    <style type="text/css">
      p#para2{
        text-align: center;
        color: red;
      }
    </style>
  </head>
  <body>
    <p id = "para1">Đoạn văn bản 1</p>
    <p id = "para2">Đoạn văn bản 2</p>
  </body>
</html>
```

4.3.ID Selector

❑ **Ví dụ:** đoạn mã sau đây có thể hiệu lực cho thẻ đầu tiên có id là "xyz".

```
<html>
  <head>
    <style type="text/css">
      p#xyz{
        text-align: center;
        color: red;
      }
    </style>
  </head>
  <body>
    <p id = "xyz">Đoạn văn bản có hiệu lực</p>
    <b id = "xyz">Đoạn văn bản không có hiệu lực</b>
  </body>
</html>
```

4.4. Class Selector

- ❑ Việc tạo các lớp, cho phép định nghĩa **nhiều kiểu thể hiện khác nhau** cho cùng một thẻ HTML và áp dụng các lớp này vào **các vị trí cần thiết** trên trang web.
- ❑ **Có 2 cách định nghĩa bộ chọn lớp**
 - Định nghĩa bộ chọn lớp cho thẻ cụ thể
 - Định nghĩa bộ chọn lớp không xác định thẻ cụ thể
 - ✓ Áp dụng một lớp cho nhiều thẻ khác nhau.

4.4. Class Selector

□ Bộ chọn lớp cho thẻ cụ thể

Cú pháp:

```
[Tên thẻ].<Tên class>{  
    Thuộc tính 1: giá trị;  
    ...  
    Thuộc tính n: giá trị;  
}
```

4.4.Class Selector

□ Ví dụ:

```
<html>
  <head>
    <style type="text/css">
      p.giua{
        text-align: center;
      }
    </style>
  </head>
  <body>
    <p class = "giua">Đoạn văn bản có hiệu lực</p>
    <p>Đoạn văn bản không có hiệu lực</p>
  </body>
</html>
```


4.4. Class Selector

❑ Bộ chọn lớp không xác định thể

Cú pháp:

```
.<Tên class>{  
    Thuộc tính 1: giá trị;  
    ...  
    Thuộc tính n: giá trị;  
}
```

4.4.Class Selector

□ Ví dụ:

```
<html>
  <head>
    <style type="text/css">
      .giua{
        text-align: center;
      }
    </style>
  </head>
  <body>
    <p class = "giua">Đoạn văn bản có hiệu lực</p>
    <h3 class = "giua">Đoạn văn bản cũng có hiệu lực</h3>
  </body>
</html>
```

4.4.Class Selector

❑ Sử dụng nhiều lớp

Một thẻ có thể gán nhiều lớp bằng cách chỉ ra các lớp, phân cách nhau bằng khoảng trắng.

Ví dụ:

```
.warning {color: red}
```

```
.highlight {background-color: yellow}
```

```
<h3 class = "warning highlight"> Danger </h3>
```

```
<p class = "highlight"> An important point </p>
```

4.5.Descendant Selector

□ Dùng để chọn một thành phần/thẻ là con hoặc cháu (hoặc cấp sâu hơn) của thành phần/thẻ khác.

Cú pháp:

```
Thẻ Thẻ Thẻ ...{  
    Thuộc tính 1: giá trị;  
    ...  
    Thuộc tính n: giá trị;  
}
```

4.5.Descendant Selector

□ Ví dụ:

```
<style type="text/css">
```

```
  p em {color: red}
```

```
  div em{color: blue}
```

```
</style>
```

```
<body>
```

```
  <p><em>Thành phần em thứ nhất là con của p</em></p>
```

```
    <div>
```

```
      <ul>
```

```
        <li><em>Thành phần em thứ 2 là con của li, là cháu của ul và div</em></li>
```

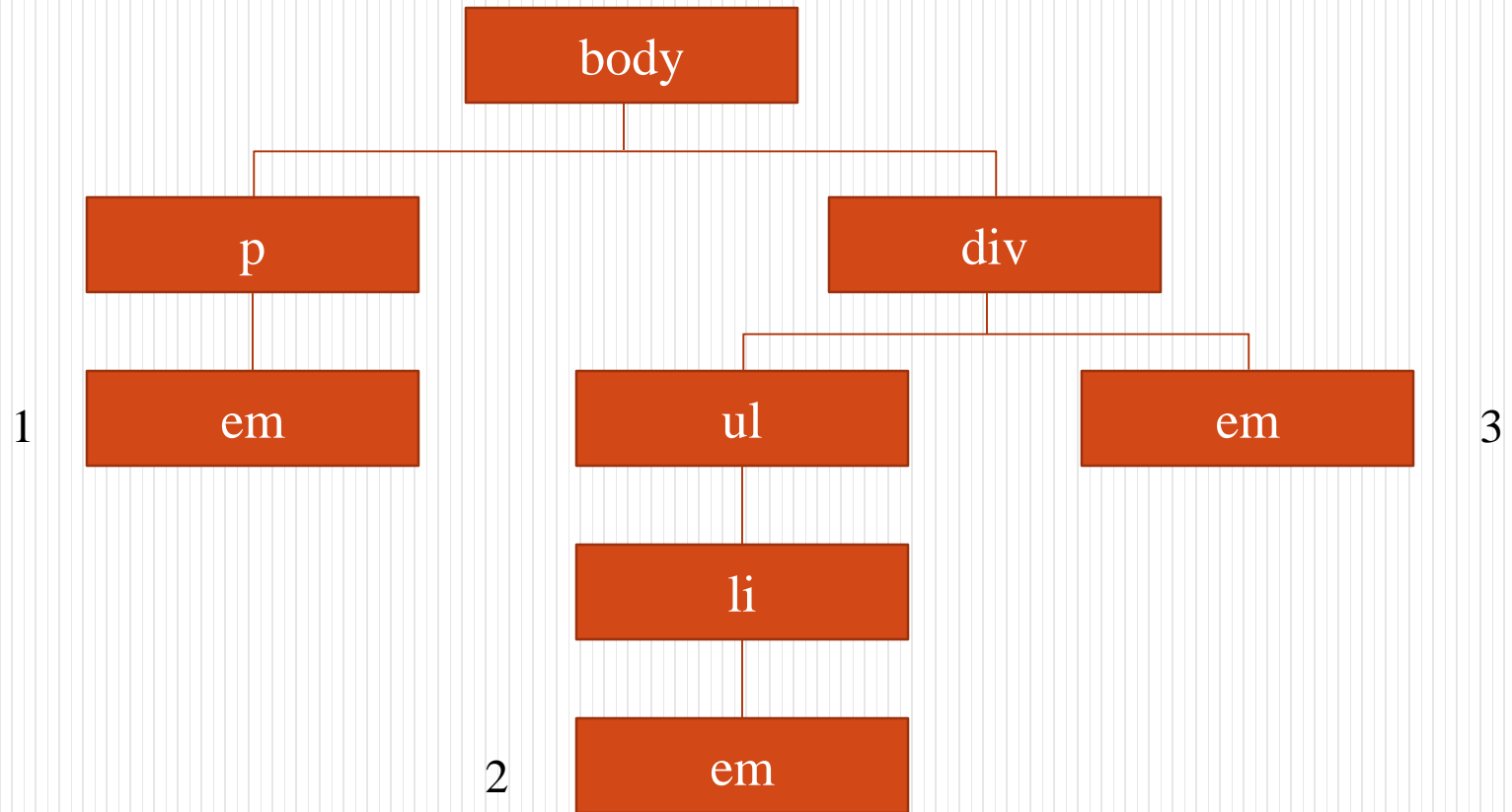
```
      </ul>
```

```
    <em>Thành phần em thứ 3 là con của div</em>
```

```
  </div>
```

```
</body>
```

4.5.Descendant Selector



Thành phần em thứ 2 là hậu duệ và thành phần em thứ 3 là con của div nên được hiển thị với chữ màu xanh.

4.6. Child Selector

❑ Dùng để chọn thành phần/ thẻ con của một thành phần/thẻ khác.

Cú pháp:

```
Thẻ > Thẻ > Thẻ ... {  
    Thuộc tính 1: giá trị;  
    ...  
    Thuộc tính n: giá trị;  
}
```

4.6. Child Selector

❑ Ví dụ:

```
<style type="text/css">  
  p em {color: red}  
  div > em{color: blue}  
</style>
```

```
<body>  
  <p><em>Thành phần em thứ nhất là con của p</em></p>  
    <div>  
      <ul>  
        <li><em>Thành phần em thứ 2 là con của li, là cháu của ul và div</em></li>  
      </ul>  
      <em>Thành phần em thứ 3 là con của div</em>  
    </div>  
</body>
```


4.7. Adjacent sibling selectors

□ Cho phép chọn một thành phần B ở gần thành phần A nhất trong tài liệu HTML, mà giữa A và B không có một thành phần D nào khác.

Cú pháp:

```
A + B {  
    Thuộc tính 1: giá trị ;  
    ...  
    Thuộc tính n: giá trị;  
}
```

4.7. Adjacent sibling selectors

□ Ví dụ:

<body>

<p>Đoạn văn bản này là thành phần A</p>

Thành phần span này là B

<div>Thành phần div là C</div>

</body>

Chọn thành phần <div> C dựa vào thành phần B và hiển thị với màu đỏ:

```
span + div {  
    color: red;  
}
```

4.8. Attribute Selector

□ Cho phép chọn các phần tử/thẻ dựa vào **thuộc tính** của các phần tử thẻ.

Cú pháp	Mô tả
E[attr]	Chọn phần tử E có thuộc tính là attr mà không quan tâm tới giá trị của thuộc tính này là gì
E[attr="value"]	Chọn phần tử E có thuộc tính là attr được thiết lập giá trị xác định là value
E[attr~="value"]	Chọn phần tử E với thuộc tính attr mà trong danh sách các giá trị của thuộc tính attr có chứa giá trị là value
E[attr^="value"]	Chọn phần tử E với thuộc tính attr có giá trị bắt đầu là value
E[attr\$="value"]	Chọn phần tử E với thuộc tính attr có giá trị kết thúc là value
E[attr*="value"]	Chọn phần tử E với thuộc tính attr có chứa giá trị là value
E[attr = "value"]	Chọn phần tử E với thuộc tính attr mà giá trị thuộc tính attr có dạng "value-?"

4.8.Attribute Selector

- ❑ **Ví dụ 1:** Chọn thẻ p có chứa thuộc tính **title**.

p[title] {color: red;} /*Không quan tâm đến giá trị của title */

<p title = "xyz"> Đoạn văn bản 1</p>

<p title = "abc"> Đoạn văn bản 2</p>

- ❑ **Ví dụ 2:** Chọn thẻ có định nghĩa thuộc tính **title** là **"xyz"**

[title = "xyz"] {color: red;} /*Không xác định thẻ cụ thể nào*/

<p title = "xyz"> Đoạn văn bản 1</p>

<b title = "xyz"> Đoạn văn bản 2

4.8.Attribute Selector

❑ **Ví dụ 3:** Chọn thẻ **h3** nếu trong danh sách các giá trị thuộc tính **class** có chứa giá trị **class2**.

```
h3[ class = "class2"] {font-weight: bold;}
```

<h3 class = "class2"> Thẻ này được chọn</h3>

<h3 class = "class1 class2"> Thẻ này cũng được chọn</h3>

❑ **Ví dụ 4:** Chọn thẻ **p** có thuộc tính **foo** có dạng "bar-?"

```
p[ foo| = "bar"] {background-color: yellow;}
```

<p foo = "bar-foo"> Thẻ này được chọn</p>

<p foo = "foo-bar"> Thẻ này không được chọn</p>

4.9.Pseudo Class Selector

- ❑ Cho phép định dạng các phần tử không nằm trong document tree.
 - Định dạng trạng thái liên kết, định dạng cho ký tự đầu tiên trong văn bản
- ❑ Khái niệm **lớp giả**, hay **phần tử giả** cho phép định dạng các phần tử dựa trên thông tin chứa bên ngoài document tree.

4.9.1 Pseudo Class

- ❑ Dùng để thêm vào các hiệu ứng đặc biệt cho các bộ chọn.
- ❑ Cho phép bộ chọn chọn các phần tử mà không quan tâm đến tên, thuộc tính hay nội dung.

Cú pháp pseudo class:

```
selector:pseudo-class{  
    Thuộc tính 1: giá trị;  
    Thuộc tính 2: giá trị;  
    ...  
}
```

Kết hợp CSS class với pseudo class:

```
selector.CSS-class:pseudo-class{  
    Thuộc tính 1: giá trị;  
    Thuộc tính 2: giá trị;  
    ...  
}
```


4.9.1.Pseudo Class

❑ :first-child:

Lớp ảo cho phép chọn phần tử đầu tiên của một phần tử khác.

Ví dụ: Chọn phần tử **p** đầu tiên nằm trong phần tử **div**, quy định chữ màu đỏ và in đậm cho phần tử **p** này.

```
<html>
  <head>
    <style type = "text/css">
      div > p:first-child{color: red; font-weight: bold;}
    </style>
  </head>
  <body>
    <div>
      <p>Đoạn văn bản thứ nhất</p>
      <p>Đoạn văn bản thứ 2</p>
      <p>Đoạn văn bản thứ 3</p>
    </div>
  </body>
</html>
```

4.9.1.Pseudo Class

❑ Anchor pseudo class:

Lớp ảo định nghĩa style cho các trạng thái link

Có 4 trạng thái link:

link: link chưa được mở (unvisited link)

visited: link đã được mở (visited link)

hover: link đang rê chuột bên trên (mouse over link)

active: link đã được chọn (selected link)

Pseudo Class Selector – Anchor

```
<html>
<head>
  <style type="text/css">
    a:link {color: #FF0000}
    a:visited {color: #00FF00}
    a:hover {color: #FF00FF}
    a:active {color: #0000FF}
  </style>
</head>
<body>
<p><b><a href="#" target="_blank">This is a link</a></b></p>
</body>
</html>
```

4.9.1.Pseudo Class

❑ Kết hợp css class và pseudo class

```
<html>
<head>
  <style type="text/css">
    a.one:link {color: #ff0000}
    a.one:visited {color: #0000ff}
    a.one:hover {background: #66ff66}

    a.two:link {color: #ff0000; text-decoration: none}
    a.two:visited {color: #0000ff; text-decoration: none}
    a.two:hover {text-decoration: underline}
  </style>
</head>
<body>
  <p><b><a class="one" href="#" target="_blank">Using style one</a></b></p>
  <p><b><a class="two" href="#" target="_blank">Using style two</a></b></p>
</body>
</html>
```

4.9.1. Pseudo Class

❑ :focus pseudo class:

Lớp ảo định nghĩa style của phần tử khi phần tử đang **focus**

```
<style type="text/css">
  a:link {color: #FF0000}
  a:focus {color: #00FF00}
</style>
```

Link sẽ chuyển sang
màu xanh lá cây khi
focus

4.9.2. Pseudo Element

❑ Bổ sung một số hiệu ứng đặc biệt cho bộ chọn. Cho phép chọn và định dạng cho các phần **văn bản đặc biệt** trong tài liệu.

Cú pháp của pseudo-element:

```
selector:pseudo-element{  
    Thuộc tính 1: giá trị;  
    Thuộc tính 2: giá trị;  
    ...  
}
```

Kết hợp CSS class với pseudo element:

```
Selector.CSS-class:pseudo-element {  
    Thuộc tính 1: giá trị;  
    Thuộc tính 2: giá trị;  
    ...  
}
```

4.9.2. Pseudo Element

❑ :first-letter pseudo element:

Phần tử mô tả cho ký tự đầu tiên của đoạn văn bản.

```
<html>
<head>
  <style type="text/css">
    p:first-letter { color: #ff0000; font-size:xx-large }
  </style>
</head>
<body>
  <p>
    You can use the :first-letter pseudo-element to add a
    special effect to the first letter of a text!
  </p>
</body>
</html>
```

You can use the first-letter pseudo-element to add a special effect to the first letter of a text!

4.9.2. Pseudo Element

❑ Các thuộc tính của first-letter:

- font properties
- color properties
- background properties
- margin properties
- padding properties
- border properties
- text-decoration
- vertical-align
- text-transform
- line-height
- float
- clear

4.9.2. Pseudo Element

❑ :first-line pseudo element:

Quy định định style cho dòng đầu tiên

```
<html>
<head>
  <style type="text/css">
    p:first-line { color: #ff0000; font-variant: small-caps }
  </style>
</head>
<body>
  <p>
    You can use the :first-line pseudo-element to add a special
    effect to the first line of a text!
  </p>
</body>
</html>
```

You can use the first-line pseudo-element to
add a special effect to the first line of a text!

4.9.2. Pseudo Element

❑ Các thuộc tính của first-line

- font properties
- color properties
- background properties
- word-spacing
- letter-spacing
- text-decoration
- vertical-align
- text-transform
- line-height
- clear

4.9.2. Pseudo Element

❑ :before pseudo element:

Chèn nội dung phía trước nội dung của một thành phần

```
<html>
<head>
  <style type="text/css">
    h1:before { content: url(smiley.gif) }
  </style>
</head>
<body>
  <h1>This is a header</h1>
  <p> The :before pseudo-element can be used to insert
    some content before the content of an element.</p>
</body>
</html>
```



This is a header

The :before pseudo-element can be used to insert some content before the content of an element.

4.9.2. Pseudo Element

❑ :after pseudo element:

Chèn nội dung phía sau nội dung của một thành phần

```
<head>
  <style type="text/css">
    h1:after { content: url(smiley.gif) }
  </style>
</head>
<body>
  <h1>This is a header</h1>
  <p> The :after pseudo-element can be
    used to insert some
    content after the content of an element.
  </p>
</body>
```

This is a header 

The :after pseudo-element can be used to insert some content after the content of an element.

4.10. Group Selector

- Group selector là cách nhóm các selector có cùng chung một quy luật giúp file CSS được gọn gàng hơn, mỗi selector sẽ được ngăn cách nhau bởi dấu , (Phẩy)

Ví dụ:

```
h1 {color: blue}
```

```
h2 {color: blue}
```

```
h3 {color: blue}
```

Ta có thể nhóm chung 3 selector trên lại như sau:

```
h1, h2, h3
```

```
{
```

```
  color: blue;
```

```
}
```

← Các bộ chọn phân cách nhau bằng ","

← Thiết lập thuộc tính color là green cho các bộ chọn

5. Đơn vị đo lường CSS

Đơn vị	Mô tả
%	Phần trăm
In	Inch (1 inch = 2.54 cm)
cm	Centimeter
mm	Millimeter
pc	Pica (1 pc = 12 pt)
px	Pixels (điểm ảnh trên màn hình máy tính)
pt	Point (1 pt = 1/72 inch)
em	1 em tương đương kích thước font hiện hành, nếu font hiện hành có kích cỡ 14px thì 1 em = 14 px. Đây là một đơn vị rất hữu ích

Chúng ta thường sử dụng các đơn vị sau để code:

?: thường được dùng để định dạng phần trăm cho chiều rộng hay khoảng cách, hoặc font size.

em: thường được dùng cho những nội dung có khả năng co giãn khi zoom text.

px: thường được dùng để định dạng khoảng cách, chiều rộng, chiều cao,...

5.Đơn vị đo lường CSS

Đơn vị màu sắc

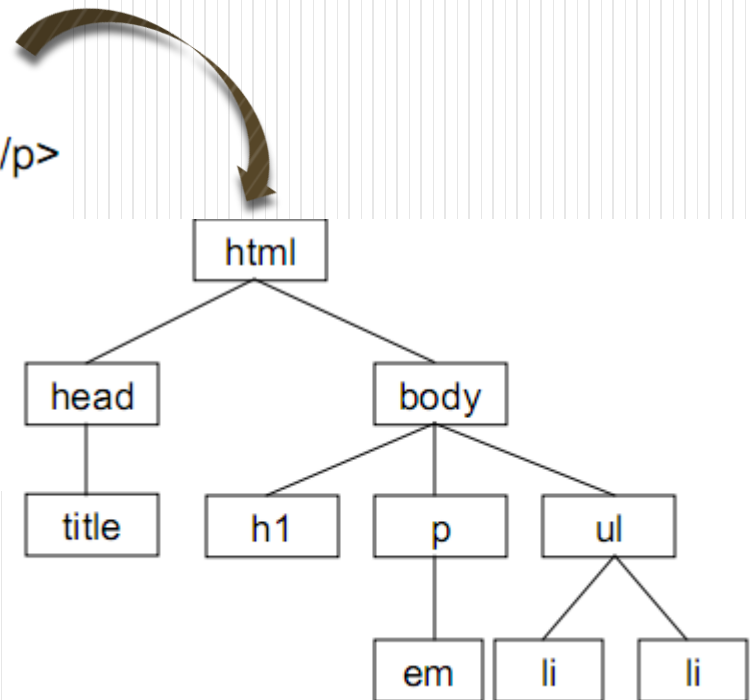
Đơn vị	Mô tả
Color-name	Tên màu. Ví dụ: black, white, red, green, blue, cyan, magenta,...
RGB (r,g,b)	Màu RGB với 3 giá trị R, G, B có trị từ 0 – 255 kết hợp với nhau tạo ra vô số màu.
RGB(%r,%g,%b)	Màu RGB với 3 giá trị R, G, B có trị từ 0 – 100% kết hợp.
Hexadecimal RGB	Mã màu RGB dạng hệ thập lục. Ví dụ: #FFFF: trắng, #000: đen,

6. Kế thừa thuộc tính

- Tài liệu HTML được biểu diễn theo cấu trúc cây (document tree)
- Các phần tử con sẽ kế thừa một số thuộc tính từ phần tử cha. Hay có thể nói phần tử cha truyền một số thuộc tính xuống phần tử con.
- Một số thuộc tính kế thừa như:
 - ✓ Màu sắc
 - ✓ Font
 - ✓ Canh lề

6. Kế thừa thuộc tính

```
<html>  
  <head>  
    <title>Tài liệu HTML</title>  
  </head>  
  <body>  
    <h1>Tiêu đề 1</h1>  
    <p>Đoạn văn bản <em>in nghiêng</em></p>  
    <ul>  
      <li>Mục 1</li>  
      <li>Mục 2</li>  
    </ul>  
  </body>  
</html>
```



6. Kế thừa tính

❑ Các style có thể kế thừa từ phần tử cha

Không kế thừa **border** & **padding**

`<div style="color: blue; border: 1px solid red; padding: 10px;">`

Đoạn văn bản này có chữ màu xanh

`<p>`Đoạn văn bản này cũng có chữ màu xanh, bởi vì thuộc tính color được kế thừa, nhưng thuộc tính border và padding không được kế thừa từ phần tử cha `div</p>`

`</div>`

Đoạn văn bản này có chữ màu xanh

Đoạn văn bản này cũng có chữ màu xanh, bởi vì thuộc tính color được kế thừa, nhưng thuộc tính border và padding không được kế thừa từ phần tử cha div

6. Kế thừa thuộc tính

❑ Ghi đè thuộc tính của phần tử cha

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

```
p{background-color: yellow;}
```

```
<body>
```

```
  <div>
```

```
    <h3>Tiêu đề</h3>
```

```
    <p>Đoạn văn bản này có nền vàng</p>
```

```
    <b>Đoạn văn bản in đậm</b>
```

```
  </div>
```

```
</body>
```

Tiêu đề

Đoạn văn bản này có nền vàng

Đoạn văn bản in đậm

7. Các nhóm thuộc tính trong CSS

7.1. Thuộc tính định dạng font chữ, văn bản (font, text)

7.2. Thuộc tính định dạng nền (background)

7.3. Mô hình hộp (box model)

7.4. Các thuộc tính định margin, padding, border

7.5. Thuộc tính định dạng cách hiển thị (display)

7.6. Thuộc tính xác định vị trí (position)

7.1. Thông tin định dạng font chữ

Thuộc tính	Mô tả	Giá trị	NN	IE	W3C
font	Một thuộc tính nhỏ để thiết lập tất cả thuộc tính cho một phong trong một khai báo.	<i>font-style</i> <i>font-variant</i> <i>font-weight</i> <i>font-size/line-height</i> <i>font-family</i> caption icon menu message-box small-caption status-bar	4.0	4.0	CSS1
font-family	Một danh sách ưu tiên của các họ phong cho một thành phần.	<i>family-name</i> <i>generic-family</i>	4.0	3.0	CSS1

7.1. Thông tin định dạng font chữ

Thuộc tính	Mô tả	Giá trị	NN	IE	W3C
font-size	Thiết lập kích cỡ cho một phong chữ.	xx-small x-small small medium large x-large xx-large smaller larger <i>length</i> %	4.0	3.0	CSS1
font-style	Thiết lập kiểu cách của phong.	normal italic oblique	4.0	4.0	CSS1

7.1. Thông tin định dạng font chữ

Thuộc tính	Mô tả	Giá trị	NN	IE	W3C
font-variant	Hiển thị văn bản trong một phong chữ HOA NHỎ hoặc một phong chữ thường	normal small-caps	60		CSS1
font-weight	Thiết lập trọng lượng của phong.	bold bolder lighter 100 200 300 400 500	4.0	4.0	CSS1

7.2. Thông tin định dạng văn bản

Thuộc tính	Mô tả	Giá trị	NN	IE	W3C
color	Thiết lập màu cho văn bản	<i>color</i>	4.0	3.0	CSS1
direction	Thiết lập hướng cho văn bản	ltr rtl			CSS2
letter-spacing	Tăng hoặc giảm khoảng trống giữa các ký tự	normal <i>length</i>	6.0	4.0	CSS1
text-align	Căn chỉnh văn bản trong một thành phần	left right center justify	4.0	4.0	CSS1

7.2. Thông tin định dạng văn bản

Thuộc tính	Mô tả	Giá trị	NN	IE	W3C
text-decoration	Thêm trang trí cho văn bản	none underline overline line-through blink	4.0	4.0	CSS1
text-indent	Thụt dòng đầu văn bản trong một thành phần	<i>length</i> %	4.0	4.0	CSS1
text-shadow	Thêm bóng cho văn bản	none <i>color</i> <i>length</i>			

7.2. Thông tin định dạng văn bản

Thuộc tính	Mô tả	Giá trị	NN	IE	W3C
text-transform	Kiểm soát ký tự trong một thành phần	none capitalize uppercase lowercase	4.0	4.0	CSS1
white-space	Thiết lập bao nhiêu khoảng trắng trong một thành phần	normal pre nowrap	4.0	5.5	CSS1
word-spacing	Tăng hoặc giảm khoảng trống giữa các từ	normal <i>length</i>	6.0	6.0	CSS1

7.3. Thông tin định dạng nền

Thuộc tính	Mô tả	Giá trị	NN	IE	W3C
background	Một thuộc tính ngắn gọn để thiết lập tất cả các thuộc tính nền trong một khai báo.	<i>background-color</i> <i>background-image</i> <i>background-repeat</i> <i>background-attachment</i> <i>background-position</i>	6.0	4.0	CSS1
background-attachment	Thiết lập liệu một ảnh có đứng cố định một chỗ hay cuộn theo phần văn bản còn lại của trang.	scroll fixed	6.0	4.0	CSS1
background-color	Thiết lập màu nền của một phần tử	<i>color-rgb</i> <i>color-hex</i> <i>color-name</i> transparent	4.0	4.0	CSS1

7.3. Thông tin định dạng nền

Thuộc tính	Mô tả	Giá trị	NN	IE	W3C
background-image	Thiết lập một ảnh như nền của trang.	<i>url</i> <i>none</i>	4.0	4.0	CSS1
background-position	Thiết lập điểm xuất phát của một ảnh nền	<i>top left</i> <i>top center</i> <i>top right</i> <i>center left</i> <i>center center</i> <i>center right</i> <i>bottom left</i> <i>bottom center</i> <i>bottom right</i> <i>x-% y-%</i> <i>x-pos y-pos</i>	6.0	4.0	CSS1
background-repeat	Thiết lập cách một ảnh sẽ được lặp lại như thế nào.	<i>repeat</i> <i>repeat-x</i> <i>repeat-y</i> <i>no-repeat</i>	4.0	4.0	CSS1

7.4. Mô hình hộp (box model)

- Tài liệu HTML gồm các phần tử chứa trong phần tử khác theo document tree.
- Mỗi phần tử được xem như box, chứa các thành phần **nội dung**, **border**, **margin**, **padding**...
- Tài liệu HTML chính là tập các **box lồng nhau**, mỗi box có thể chứa các box con khác và có box lớn ngoài cùng.

7.4. Mô hình hộp (Box model)



7.5. Các thông tin định dạng lề

Thuộc tính	Mô tả	Giá trị	NN	IE	W3C
margin	Một thuộc tính nhanh để thiết lập thuộc tính cho lề trong một khai báo.	<i>margin-top</i> <i>margin-right</i> <i>margin-bottom</i> <i>margin-left</i>	4.0	4.0	CSS1
margin-bottom	Thiết lập lề đáy của một phần tử.	auto <i>length</i> %	4.0	4.0	CSS1
margin-left	Thiết lập lề trái của một phần tử.	auto <i>length</i> %	4.0	3.0	CSS1
margin-right	Thiết lập lề phải của một phần tử.	auto <i>length</i> %	4.0	3.0	CSS1
margin-top	Thiết lập lề đỉnh của một phần tử.	auto <i>length</i> %	4.0	3.0	CSS1

7.6. Thông tin định dạng vùng đệm

Thuộc tính	Mô tả	Giá trị	NN	IE	W3C
padding	Một thuộc tính ngắn gọn để thiết lập tất cả các khoảng đệm chỉ với một lần khai báo.	padding-top padding-right padding-bottom padding-left	4.0	4.0	CSS1
padding-bottom	Thiết lập khoảng đệm tại đáy của một phần tử	length %	4.0	4.0	CSS1
padding-left	Thiết lập khoảng đệm phía trái của một phần tử.	length %	4.0	4.0	CSS1
padding-right	Thiết lập khoảng đệm phía phải của một phần tử.	length %	4.0	4.0	CSS1
padding-top	Thiết lập khoảng đệm trên đỉnh của một phần tử.	length %	4.0	4.0	CSS1

Thông tin định dạng lề + vùng đệm

Khi khai báo rút gọn giá trị margin, padding thì các giá trị tính theo chiều kim đồng hồ tính từ top



Top, Right, Bottom, Left

7.7. Thông tin định dạng biên

Thuộc tính	Mô tả	Giá trị	NN	IE	W3C
border	Một thuộc tính rút gọn để thiết lập thuộc tính cho toàn bộ cả 4 đường bao trong một khai báo.	border-width border-style border-color	4.0	4.0	CSS1
border-bottom	Một thuộc tính rút gọn để thiết lập thuộc tính cho đường bao đáy trong một khai báo.	border-bottom-width border-style border-color	6.0	4.0	CSS1
border-bottom-color	Thiết lập màu sắc của đường bao đáy	border-color	6.0	4.0	CSS2
border-bottom-style	Thiết lập kiểu cách của đường bao đáy.	border-style	6.0	4.0	CSS2

7.7. Thông tin định dạng biên

Thuộc tính	Mô tả	Giá trị	NN	IE	W3C
border-bottom-width	Thiết lập độ rộng của đường bao đáy	thin medium thick length	4.0	4.0	CSS1
border-color	Thiết lập màu sắc của bốn đường bao, có thể đặt màu từ một đến bốn.	color	6.0	4.0	CSS1
border-left	Một thuộc tính tốc hành để thiết lập thuộc tính cho đường bao trái trong một khai báo.	border-left-width border-style border-color	6.0	4.0	CSS1
border-left-color	Thiết lập màu sắc của đường bao trái.	border-color	6.0	4.0	CSS2

7.7. Thông tin định dạng biên

Thuộc tính	Mô tả	Giá trị	NN	IE	W3C
border-left-style	Thiết lập kiểu cách của đường bao trái.	border-style	6.0	4.0	CSS2
border-left-width	Thiết lập độ rộng của đường bao trái.	thin medium thick length	4.0	4.0	CSS1
border-right	Một thuộc tính tổng hành để thiết lập thuộc tính cho đường bao phải trong một khai báo.	border-right-width border-style border-color	6.0	4.0	CSS1
border-right-color	Thiết lập màu sắc của đường bao phải.	border-color	6.0	4.0	CSS2

7.7. Thông tin định dạng biên

Thuộc tính	Mô tả	Giá trị	NN	IE	W3C
border-right-style	Thiết lập kiểu cách của đường bao phải.	border-style	6.0	4.0	CSS2
border-right-width	Thiết lập độ rộng của đường bao phải	thin medium thick length	4.0	4.0	CSS1
border-style	Thiết lập kiểu cách của cả bốn đường bao, có thể đạt được kiểu từ một đến bốn.	none hidden dotted dashed solid double groove ridge inset outset	6.0	4.0	CSS1

7.7. Thông tin định dạng biên

Thuộc tính	Mô tả	Giá trị	NN	IE	W3C
border-top	Một thuộc tính tốc hành để thiết lập thuộc tính cho đường bao đỉnh trong một khai báo.	border-top-width border-style border-color	6.0	4.0	CSS1
border-top-color	Thiết lập màu sắc của đường bao đỉnh.	border-color	6.0	4.0	CSS2
border-top-style	Thiết lập kiểu cách của đường bao đỉnh.	border-style	6.0	4.0	CSS2
border-top-width	Thiết lập độ rộng của đường bao đỉnh	thin medium thick length	4.0	4.0	CSS1
border-width	Một thuộc tính tốc hành để thiết lập độ rộng của bốn đường bao trong một khai báo, có thể có từ một đến bốn giá trị.	thin medium thick length	4.0	4.0	CSS1

7.8. Định dạng cách hiển thị

❑ Thuộc tính Display:

Xác định loại hiển thị của thành phần.

Thuộc tính	Giá trị	Ví dụ	Mô tả
Display	block	display: block;	Thành phần hiển thị như một khối, khi sử dụng giá trị block thành phần sẽ đứng một hàng độc lập so với thành phần trước và sau nó.
	inline	display: inline;	Thành phần sẽ hiển thị như một nội tuyến (inline, không ngắt dòng), đây là dạng mặc định.
	none	display: none;	Thành phần không hiển thị.

7.8. Định dạng cách hiển thị

Thuộc tính	Giá trị	Ví dụ	Mô tả
Display	inline-table	display: inline-table;	Thành phần sẽ hiển thị như một khối nội tuyến, đối xử tương tự <table>, không ngắt dòng trước và sau thành phần.
	list-item	display: list-item;	Thành phần sẽ hiển thị như một khối và một nội tuyến cho các điểm đánh dấu danh sách.
	inline-block	display: inline-block;	Thành phần sẽ hiển thị như một khối, nhưng là một khối nội tuyến.
	run-in	display: run-in;	Thành phần sẽ hiển thị như một khối hoặc một nội tuyến, tùy thuộc vào bối cảnh.

7.8. Định dạng cách hiển thị

Thuộc tính	Giá trị	Ví dụ	Mô tả
Display	table	display: table;	Thành phần sẽ đối xử như một table, ngắt dòng trước và sau thành phần.
	table-caption	display: table-caption;	Thành phần sẽ đối xử như một <caption> của table.
	table-cell	display: table-cell;	Thành phần sẽ đối xử như một ô trong table.
	table-column	display: table-column;	Thành phần sẽ đối xử như một cột trong table.
	table-column-group	display: table-column-group;	Thành phần sẽ đối xử như một nhóm cột (<colgroup>) trong table.

7.8. Định dạng cách hiển thị

Thuộc tính	Giá trị	Ví dụ	Mô tả
Display	table-footer-group	display: table-footer-group;	Thành phần sẽ đối xử như một nhóm footer (<tfoot>) trong table.
	table-header-group	display: table-header-group;	Thành phần sẽ đối xử như một nhóm header (<thead>) trong table.
	table-row	display: table-row;	Thành phần sẽ đối xử như một hàng trong table.
	table-row-group	display: table-row-group;	Thành phần sẽ đối xử như một nhóm hàng trong table.
	inherit	display: inherit;	Xác định thừa hưởng thuộc tính từ thành phần cha (thành phần bao ngoài).

7.8. Định dạng cách hiển thị

❑ Sử dụng **block**:

```
<html>
  <head>
    <style type="text/css">
      span {
        background: red;
        display: block;
      }
    </style>
  </head>
  <body>
    <p>Đây là ví dụ cho <span>display</span> sử dụng giá trị block.</p>
  </body>
</html>
```

Đây là ví dụ cho

display

sử dụng giá trị block.

7.8. Định dạng cách hiển thị

❑ Sử dụng **inline**:

```
<html>
  <head>
    <style type="text/css">
      span {
        background: red;
        display: inline;
      }
    </style>
  </head>
  <body>
    <p>Đây là ví dụ cho <span>display</span> sử dụng giá trị block.</p>
  </body>
</html>
```

Đây là ví dụ cho **display** sử dụng giá trị block.

7.9. Thông tin xác định vị trí

Thuộc tính	Mô tả	Giá trị	
top, bottom, left, right	Khoảng cách cạnh trên, dưới, trái, phải của thành phần với các cạnh của thành phần cha	auto % length	
position	Cách bố trí của thành phần: static, relative, absolute, fixed	static absolute	relative fixed
clip	Cách xén nội dung của thành phần	shape	auto
overflow	Quy định cách hiển thị khi nội dung của thành phần vượt quá vùng quy định	visible scroll	hidden auto
vertical-align	Lề theo phương đứng của thành phần	baseline super text-top bottom length	sub top middle text-bottom %
z-index	Thứ tự trên/dưới của thành phần	auto	number

7.9. Thông tin xác định vị trí

Giá trị thuộc tính position

Giá trị	Mô tả
static	<ul style="list-style-type: none">- Được thiết lập mặc định (cho mọi thành phần)- Thành phần với <code>position:static</code> sẽ có vị trí được xác định theo luồng chuẩn (normal flow) của trang chứa nó- Thành phần với <code>position:static</code> sẽ bỏ qua các giá trị của thuộc tính top, bottom, left và right
relative	Thành phần với <code>position:relative</code> sẽ di chuyển thành phần đến vị trí mới dựa vào vị trí hiện tại của thành phần này và các giá trị của top, bottom, left, right
absolute	Thành phần với <code>position:absolute</code> sẽ đặt thành phần tại vị trí xác định theo các giá trị của top, bottom, left, right
fixed	Thành phần với <code>position:fixed</code> sẽ cố định (neo) thành phần theo các giá trị của top, bottom, left, right

7.9. Thông tin xác định vị trí

Thiết lập vị trí thành phần theo vị trí tương đối

```
h2.pos_left {position:relative;left:-20px}  
h2.pos_right {position:relative;left:20px}
```

```
<h2>  
    This is a heading in normal position  
</h2>  
<h2 class="pos_left">  
    This heading is moved left to its  
    normal position  
</h2>  
<h2 class="pos_right">  
    This heading is moved right to its  
    normal position  
</h2>
```

This is a heading in normal position

This heading is moved left to its normal position

This heading is moved right to its normal position

7.9. Thông tin xác định vị trí

Thiết lập vị trí thành phần theo vị trí tuyệt đối

- **Ví dụ 2:** Thiết lập vị trí của thành phần theo vị trí tuyệt đối, đặt thành phần tại toạ độ (100px, 150px)

```
h2.pos_abs { position:absolute; left:100px; top:150px}
```

- **Ví dụ 3:** Cố định/neo thành phần theo biên trên và phải của thành phần cha (chạy thử ví dụ này trên FireFox)

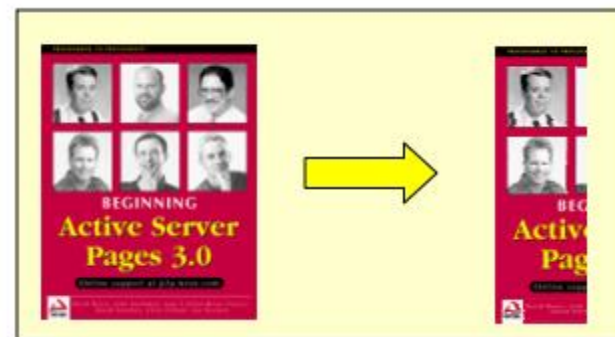
```
p { position:fixed; top:30px; right:5cm; }
```


Thông tin xác định vị trí

Giá trị thuộc tính clip

Giá trị	Mô tả
shape	Tập hợp hình dạng của thành phần
auto	Mặc định. Trình duyệt sẽ quy định hình dạng của thành phần

```
img
{
  clip: rect(0px 50px 200px 0px)
}
```



Thông tin xác định vị trí

Giá trị thuộc tính **overflow**

Giá trị	Mô tả
visible	Mặc định. Nội dung của thành phần không được xén và có thể vượt ra khỏi thành phần
hidden	Nội dung của thành phần được xén, nhưng trình duyệt sẽ không hiển thị thanh cuộn để xem phần dữ liệu bị xén
scroll	Nội dung của thành phần được xén, nhưng trình duyệt sẽ hiển thị thanh cuộn để xem phần bị xén
auto	Nếu nội dung của thành phần được xén, trình duyệt sẽ hiển thị thanh cuộn để xem phần bị xén

Thông tin xác định vị trí

`overflow: auto;`

Thêm một thanh cuộn nếu nội dung vượt ra khỏi vùng bao

`overflow: auto;`

Saeplus ullamcorper interdico, persto appellatio iustum aliquam saluto. Occuro enim inhibeo quidem wisi epulae. Magna, illum dui, praemitto, ymo laoreet, demoveo volutpat vero roto.

`overflow: scroll;`

Hiện đầy đủ một thanh cuộn.

`overflow: scroll;`

Saeplus ullamcorper interdico, persto appellatio iustum aliquam saluto. Occuro enim inhibeo quidem wisi epulae. Magna, illum dui, praemitto, ymo laoreet, demoveo volutpat vero roto.

`overflow-x`

`overflow: hidden;`

Nội dung vượt ngoài vùng bao sẽ bị che.

`overflow: hidden;`

Saeplus ullamcorper interdico, persto appellatio iustum aliquam saluto. Occuro enim inhibeo quidem wisi epulae. Magna, illum dui, praemitto, ymo laoreet, demoveo volutpat vero roto.

`overflow-y`

Thông tin xác định vị trí

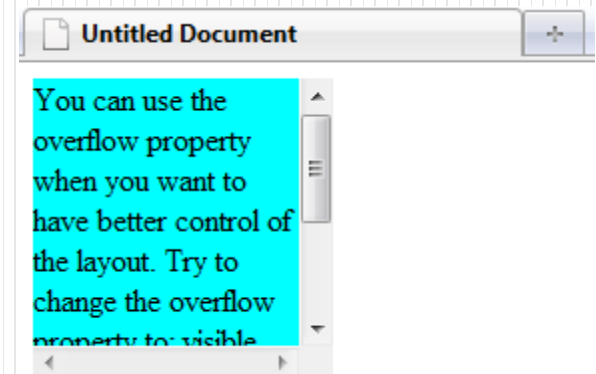
Sử dụng overflow để bật thanh cuộn khi nội dung vượt quá kích thước thành phần

```
div { background-color:#00FFFF;  
width:150px; height:150px;  
overflow: scroll  
}
```

<div>

You can use the overflow property when you want to have better control of the layout. Try to change the overflow property to: visible, hidden, auto, or inherit and see what happens. The default value is visible.

</div>



Thông tin xác định vị trí

Giá trị thuộc tính vertical-align

Giá trị	Mô tả
baseline	Mặc định. Thành phần được đặt trên đường cơ sở (baseline) của thành phần cha
sub	Giống thành phần như một chỉ số dưới (subscript)
super	Giống thành phần như một chỉ số trên (superscript)
top	Đỉnh của thành phần được giống với đỉnh của thành phần cao nhất trong dòng
text-top	Đỉnh của thành phần được giống với đỉnh của font của thành phần cha

Thông tin xác định vị trí

Giá trị thuộc tính vertical-align

Giá trị	Mô tả
middle	Thành phần được đặt chính giữa thành phần cha
bottom	Đáy của thành phần được giống với thành phần thấp nhất trong dòng
text-bottom	Đáy của thành phần được giống với đáy của font của thành phần cha
length	
%	Giống thành phần theo giá trị % của thuộc tính “line-height”. Cho phép đặt giá trị âm

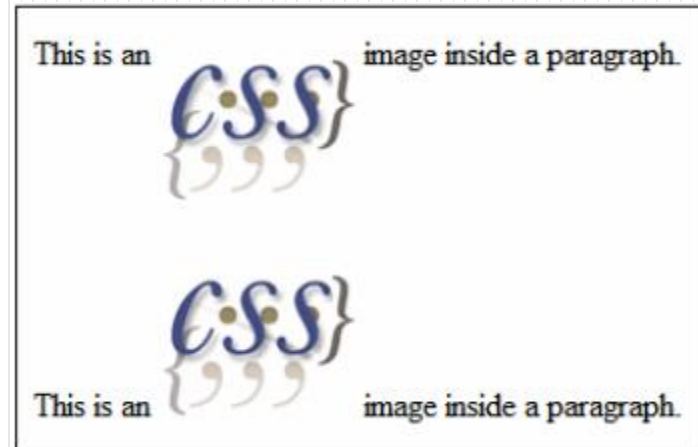
Thông tin xác định vị trí

Sử dụng vertical-align

```
img.top {vertical-align:text-top}  
img.bottom {vertical-align:text-bottom}
```

```
<p> This is an   
image inside a paragraph.</p>
```

```
<p> This is an   
image inside a paragraph.</p>
```



Thông tin xác định vị trí

Giá trị thuộc tính z-order

Giá trị	Mô tả
auto	Mặc định. Thứ tự “stack order” bằng với cha của nó
number	Thứ tự “stack order” của thành phần

```
img.x { position:absolute; left:0px; top:0px;  
        z-index:-1 }
```

```
<h1>This is a Heading</h1>
```

```

```

```
<p>Default z-index is 0. Z-index -1 has lower  
priority.</p>
```



This is a Heading

Default z-index is 0. Z-index -1 has lower priority.

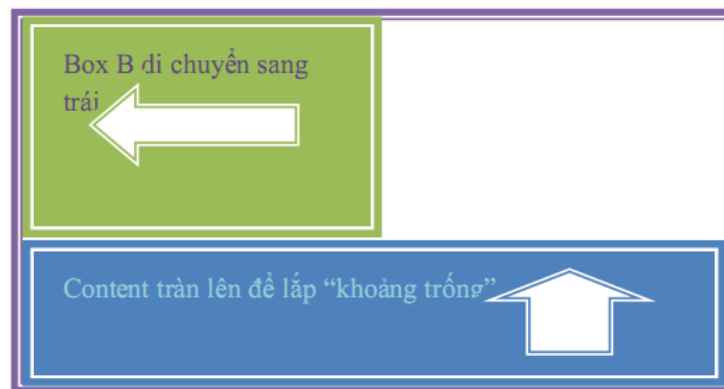
8. Float & Clear

❑ Thuộc tính float

Dùng để cố định một thành phần web về bên trái hay bên phải không gian bao quanh nó. Là một thuộc tính rất cần thiết khi dàn trang, hiển thị văn bản thành cột, hay thực hiện việc định vị trí ảnh và text như kiểu text wrapping của Word.

Thuộc tính float có 3 giá trị:

- + Left: Cố định phần tử về bên trái.
- + Right: Cố định phần tử về bên phải.
- + None: Bình thường



8. Float & Clear

❑ Thuộc tính clear:

Được gán vào các phần tử liên quan tới phần tử đã được float để quyết định hướng xử sự của phần tử này. Thuộc tính clear có 4 thuộc tính: left (tràn bên trái), right (tràn bên phải), both (không tràn) và none