



Bộ môn Công nghệ Phần mềm
Viện CNTT & TT
Trường Đại học Bách Khoa Hà Nội

Bài 03: Cascading Style Sheets



Mục đích

- Quan sát sự thay đổi hình thức trang Web khi thay đổi template
- Template gồm:
 - *template_css*
 - thư mục về JavaScript, Picture... các file css khác, file index.php,...



Mục đích

- Cung cấp cho Sinh viên những kiến thức về CSS để
 - Thực hiện việc chi phối hình thức trang Web ở mức cao hơn
 - Tạo ra sự nhất quán và thống nhất trong bố cục và trình bày trang Web



Nội dung

- 1. Giới thiệu về Cascading Style Sheet (CSS)
- 2. Lý do sử dụng CSS
- 3. Phân loại Style
- 4. Cách chèn một Style
- 5. Cấu trúc một Style
- 6. Các style cơ bản và nâng cao
- 7. Trắc nghiệm



1. Giới thiệu về Cascading Style Sheet (CSS)

- Cascading **Style Sheet** hay còn gọi tắt là **Style**
Mẫu quy định cách thức thể hiện các thẻ HTML
- Ban đầu HTML
 - Tập các quy tắc cho phép NSD xem các trang Web
 - Không hỗ trợ nhà thiết kế có nhiều quyền chi phối thiết kế
=> Các trình duyệt bổ sung thêm các thẻ HTML riêng, không theo chuẩn
- W3C (World Wide Web Consortium) ra đời CSS cho phép hỗ trợ nhà thiết kế



2. Lý do sử dụng CSS

- Style tiết kiệm thời gian
- CSS dễ thay đổi
- Sự nhất quán
- Khả năng chi phối đa dạng
- Tạo ra định dạng chung cho toàn bộ Web
- *Nhược điểm?*



3. Phân loại CSS

- Có 4 loại Style
 - Browser default
 - External Style Sheet
 - Internal Style Sheet
 - Inline Style
- Độ ưu tiên?



Tăng dần



4. Cách chèn một Style

- Với Inline style:

`<tên_thẻ style="tt1:gt1;tt2:gt2;...">`

- Ví dụ

`<b style="color:blue">Văn bản này
in đậm và màu xanh`

`Văn bản này
in đậm và màu xanh `

- *Style có lợi ích không?*



4. Cách chèn một Style

- Style thường được đưa vào trong thẻ `<div>` hoặc thẻ ``
- Thẻ `<div>`: dùng để quy định một hoặc nhiều khối phần tử
- Thẻ ``: quy định kiểu của phần tử text



4. Cách chèn một Style

- **Với Internal style:**

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

```
<!--
```

```
    Nội dung định nghĩa style
```

```
-->
```

```
</style>
```

- Thiết lập các Style ở trong hai thẻ <head> và </head>
- Cặp thẻ <!-- và --> nhằm che giấu Style khỏi các Browser không hỗ trợ chúng



4. Cách chèn một Style

- **Với External style:**

- Định nghĩa style trong file riêng (thường có đuôi .CSS)
- Nhúng file CSS đã định nghĩa vào trang web ở giữa hai thẻ <head> và </head>:

```
<link href="địa_chỉ_file"  
rel="stylesheet" type="text/css">
```



4. Cách chèn một Style

- Vai trò của từng loại
- Inline Style:

Ưu

- Phương pháp dễ dàng nhất để bổ sung thuộc tính vào thẻ.
- Dễ dàng quản lý style theo từng tag
- Có độ ưu tiên cao nhất

Nhược

- Khai báo cho từng tag thủ công, khó cập nhật



4. Cách chèn một Style

- Internal Style

Ưu

- Lý tưởng cho việc áp dụng trong những tài liệu riêng lẻ

Nhược

- Cần phải khai báo lại style cho tài liệu khác trong mỗi lần sử dụng



4. Cách chèn một Style

■ External Style

Ưu

- Đồng nhất cho mọi trang Web theo cùng một định dạng
- Thông tin style được trình duyệt lưu vào cache
- Mỗi sự thay đổi của mẫu style ngoài, các trang Web sẽ cập nhật thay đổi theo

Nhược

- Tốn thời gian tải file .css và biên dịch ở trình duyệt cho lần đầu sử dụng.
- File .css hỏng sẽ ảnh hưởng đến toàn bộ website



5. Cấu trúc một Style

Chú ý:

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

```
/*
```

```
    Đoạn ghi chú
```

```
*/
```



5. Cấu trúc một Style

```
selector{  
    Property1:Value1;  
    Propertyn:Valuen;  
}
```

- Trong đó
 - Selector: một thẻ gốc HTML hay định nghĩa của Style
 - Sau dấu ":" không có khoảng trắng
 - Các Property ngăn cách bởi dấu ;



5.1. Style áp dụng cho thẻ cụ thể

- Trường hợp 1 thẻ: *selector* là *tên_thẻ*

```
p {  
    color:red;  
}
```

- Khai báo đồng thời nhiều thẻ: Viết danh sách tên thẻ phân cách bởi dấu phẩy

```
h1,h2,h3,h4,h5,h6{  
    font-family:arial;  
}
```

5.1. Selector

Loại	Mô tả phạm vi ảnh hưởng	Ví dụ
element	Định dạng áp dụng cho ND tất cả các tag Element trong tài liệu Web	<code>h1 {color: red;}</code> /* ND của thẻ <h1> bị định dạng màu chữ=đỏ */
#id	Định dạng áp dụng cho ND tất cả các tab có thuộc tính id trong tài liệu Web	<code>#test {color: green;}</code> /* ND của bất kỳ tag có thuộc tính id=test đều bị định dạng màu chữ=xanh lá */
.class	Định dạng áp dụng cho ND tất cả các tab có thuộc tính class trong tài liệu Web	<code>.note {color: yellow;}</code> /* ND của bất kỳ tag có thuộc tính class=note đều bị định dạng màu chữ=vàng */
element . class	Định dạng áp dụng cho ND các tag Element có thuộc tính class tương ứng	<code>h1.note {text-decoration: underline;}</code> /* ND của các thẻ <h1> có thuộc tính class=note đều bị định dạng gạch chân */
Grouping	Định dạng áp dụng cho ND một nhóm các tag trong tài liệu.	<code>h1,h2,h3 {background-color: orange;}</code> /* ND của các thẻ <h1> <h2> <h3> đều bị định dạng màu nền = màu cam */
Contextual	Định dạng áp dụng cho ND các thẻ được lồng trong một thẻ cha nào đó	<code>p strong {color: purple;}</code> /* ND của các thẻ nằm trong thẻ <p> đều bị định dạng màu chữ=màu tím */
Pseudo Class Pseudo element	Định dạng được áp dụng dựa vào trạng thái của các Element. (Không xuất hiện trong mã lệnh HTML)	



5.1. Style áp dụng cho thẻ cụ thể

- Dữ liệu bị tác động bởi các thẻ <P> và <H1> có còn những thuộc tính của thẻ gốc?

=> Các thẻ HTML giữ lại (kế thừa) các thuộc tính ban đầu của mình

- Điều gì xảy ra khi trong định nghĩa loại bỏ thuộc tính của thẻ gốc?



5.2. Style áp dụng cho lớp

- Gắn với thẻ cụ thể: *selector* là **tên_thẻ.tên_lớp**

```
p.loai1{  
    color:red;  
}  
  
p.loai2{  
    color:blue;  
}
```

- Không gắn với thẻ cụ thể: bỏ phần **tên_thẻ** đi, giữ lại dấu chấm:

```
.loai3{  
    color:green;  
}
```



5.2. Style áp dụng cho lớp

- Đặt thuộc tính ***class*** của thẻ = ***"tên_lớp"***:

`<tên_thẻ class="tên_lớp">`

- Ví dụ:

`<p class="loai1">Đoạn này màu đỏ</p>`

`<h1 class="loai2">Style không có hiệu lực</h1>`

`<h2 class="loai3">Tiêu đề màu xanh</h2>`



5.3. Style kiểu định danh (id)

- Tương tự như class. Thay dấu chấm (.) thành dấu thăng (#).
- Cho thẻ cụ thể: `tên_thẻ#định_danh{...}`
- Tổng quát: `#định_danh{...}`



5.3. Style kiểu định danh (id)

- Ví dụ:

```
p#id1 {  
    color:red;  
}  
  
#id2 {  
    color:blue;  
}
```



5.3. Style kiểu định danh (id)

- *Mỗi id chỉ xuất hiện một lần trên trang*
- Đặt thuộc tính `id` của thẻ = `định_danh`
`<tên_thẻ id="định_danh">`
- Ví dụ:
`<p id="id1">Đoạn này màu đỏ</p>`
`<h1 id="id2">Tiêu đề màu xanh</h1>`



5.4. Style áp dụng cho lớp con

- Ví dụ

```
b i {background-color:yellow; }
```

```
b,i {color:blue; }
```

- Ý nghĩa

- Với dấu " " khi có cả 2 thẻ (lớp) thuộc tính mới được áp dụng
- Với dấu "," thuộc tính được áp dụng cho cả 2 thẻ (lớp)



5.4. Style áp dụng cho lớp con

- Ví dụ

```
<b>Doan nay mau blue</b>  
<b><i>Doan nay mau vang  
</i></b>  
<i>Doan nay mau blue</i>
```

Một số quy ước về đơn vị trong

CSS

■ Đơn vị chiều dài

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

Một số quy ước về đơn vị trong



■ 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,



Basic CSSs

1. CSS Background
2. CSS Text
3. CSS Font
4. CSS Border
5. CSS Margin
6. CSS Padding
7. CSS List



Advanced CSSs

1. CSS Dimension
2. CSS Classification
3. CSS Positioning
4. CSS Pseudo-class
5. CSS Pseudo-element
6. CSS Media Types



6.1. CSS Background

- Thuộc tính Background cho phép thiết lập màu nền, ảnh nền, lặp ảnh...



6.1. CSS Background

Property	Description	Values	NN	IE	W3C
background	A shorthand property for setting all background properties in one declaration	<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	Sets whether a background image is fixed or scrolls with the rest of the page	scroll fixed	6.0	4.0	CSS1
background-color	Sets the background color of an element	<i>color-rgb</i> <i>color-hex</i> <i>color-name</i> transparent	4.0	4.0	CSS1



6.1. CSS Background

background-image	Sets an image as the background	<i>url</i> none	4.0	4.0	CSS1
background-position	Sets the starting position of a background image	top left top center top right center left center center center right bottom left bottom center bottom right <i>x-% y-%</i> <i>x-pos y-pos</i>	6.0	4.0	CSS1
background-repeat	Sets if/how a background image will be repeated	repeat repeat-x repeat-y no-repeat	4.0	4.0	CSS1

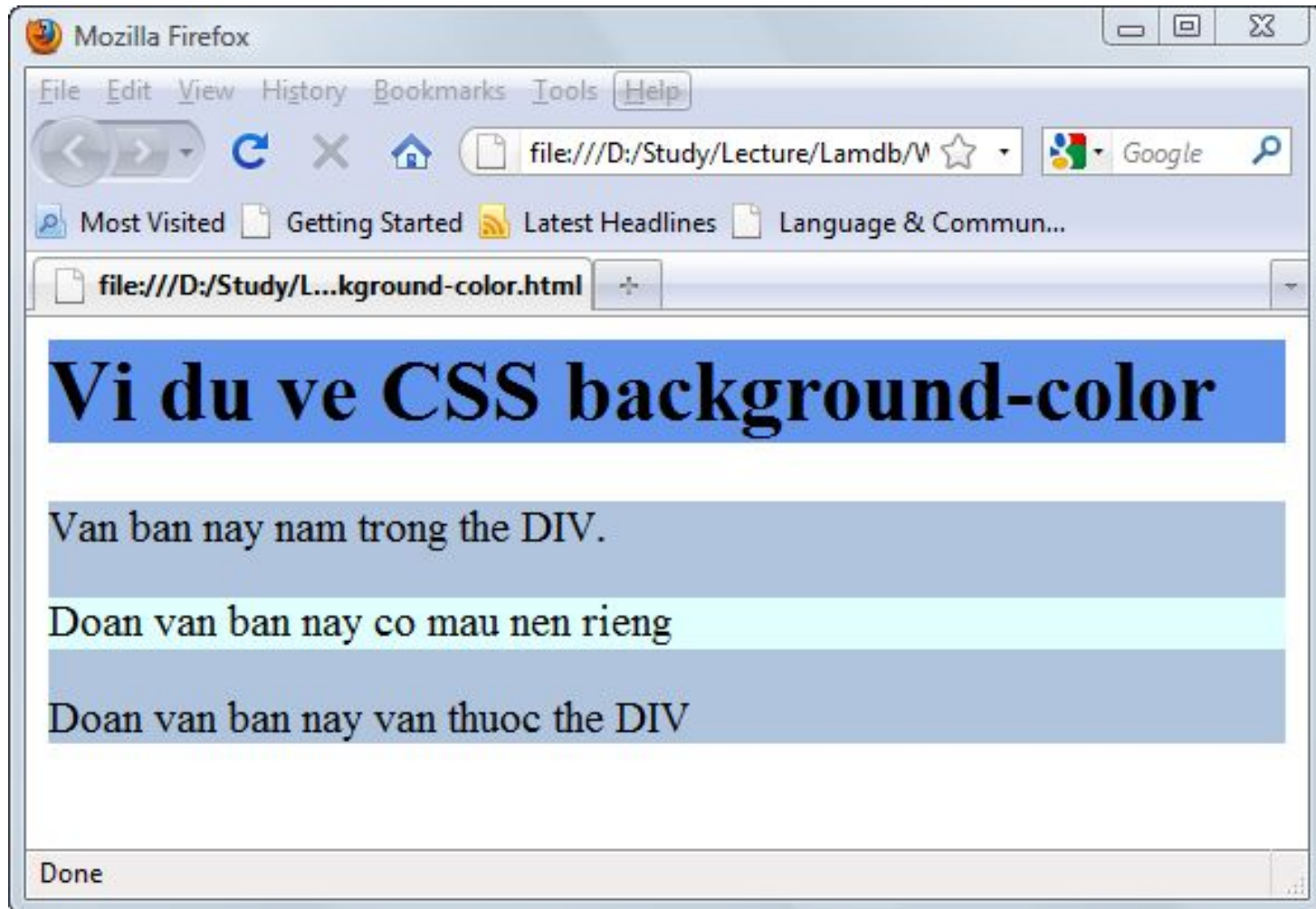


6.1. CSS Background

background-color

- Thường áp dụng cho thẻ: body và các thẻ hiển thị dữ liệu: DIV, P, H1...để đặt màu nền
- Có thể được xác định bằng 3 cách
 - name : tên màu như "red"
 - RGB: giá trị RGB như "rgb(255,0,0)"
 - Hex: giá trị hexa của màu như "#ff0000"
- Ví dụ
 - `body{background-color:"red";}`
 - `h1{background-color:#645eff;}`

6.1. CSS Background





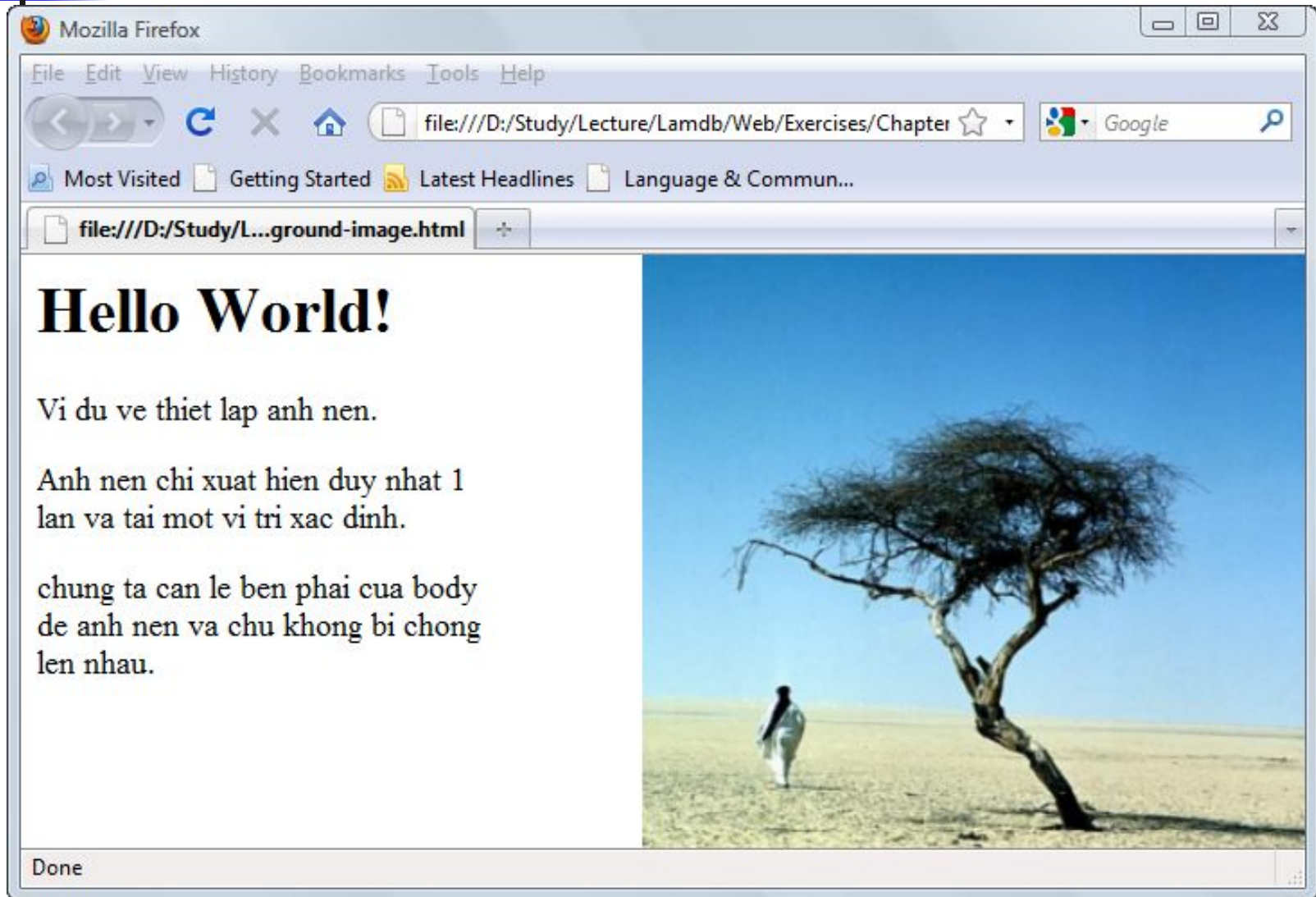
6.1. CSS Background

background-image

- Thực hiện
 - Thường sử dụng để chèn một ảnh nền BODY
 - Căn ảnh, xác định vị trí ảnh và cho phép lặp ảnh hay không
- Ví dụ

```
body{  
background-image:url('tree.png');  
background-repeat:no-repeat;  
background-position:top right;  
background-attachment:fixed;  
margin-right:300px;  
}
```

6.1. CSS Background





Thuộc tính background rút gọn

- Chuyển:
background-color:transparent;
background-image: url(logo.png);
background-repeat: no-repeat;
background-attachment: fixed;
background-position: right bottom;
- Thành:
background:transparent url(logo.png) no-repeat fixed right bottom;
- background: <background-color> |
<background-image> | <background-repeat> |
<background-attachment> | <background-position>



6.2. CSS Text

Property	Description	Possible Values	NN	IE	W3C
color	Sets the color of a text	<i>color</i>	4.0	3.0	CSS1
direction	Sets the text direction	ltr rtl			CSS2
letter-spacing	Increase or decrease the space between characters	normal <i>length</i>	6.0	4.0	CSS1
text-align	Aligns the text in an element	left right center justify	4.0	4.0	CSS1
text-decoration	Adds decoration to text	none underline overline line-through blink	4.0	4.0	CSS1
text-indent	Indents the first line of text in an element	<i>length</i> %	4.0	4.0	CSS1



6.2. CSS Text

text-shadow		none <i>color</i> <i>length</i>			
text-transform	Controls the letters in an element	none capitalize uppercase lowercase	4.0	4.0	CSS1
unicode-bidi		normal embed bidi-override		5.0	CSS2
white-space	Sets how white space inside an element is handled	normal pre nowrap	4.0	5.5	CSS1
word-spacing	Increase or decrease the space between words	normal <i>length</i>	6.0	6.0	CSS1



6.2. CSS Text

direction

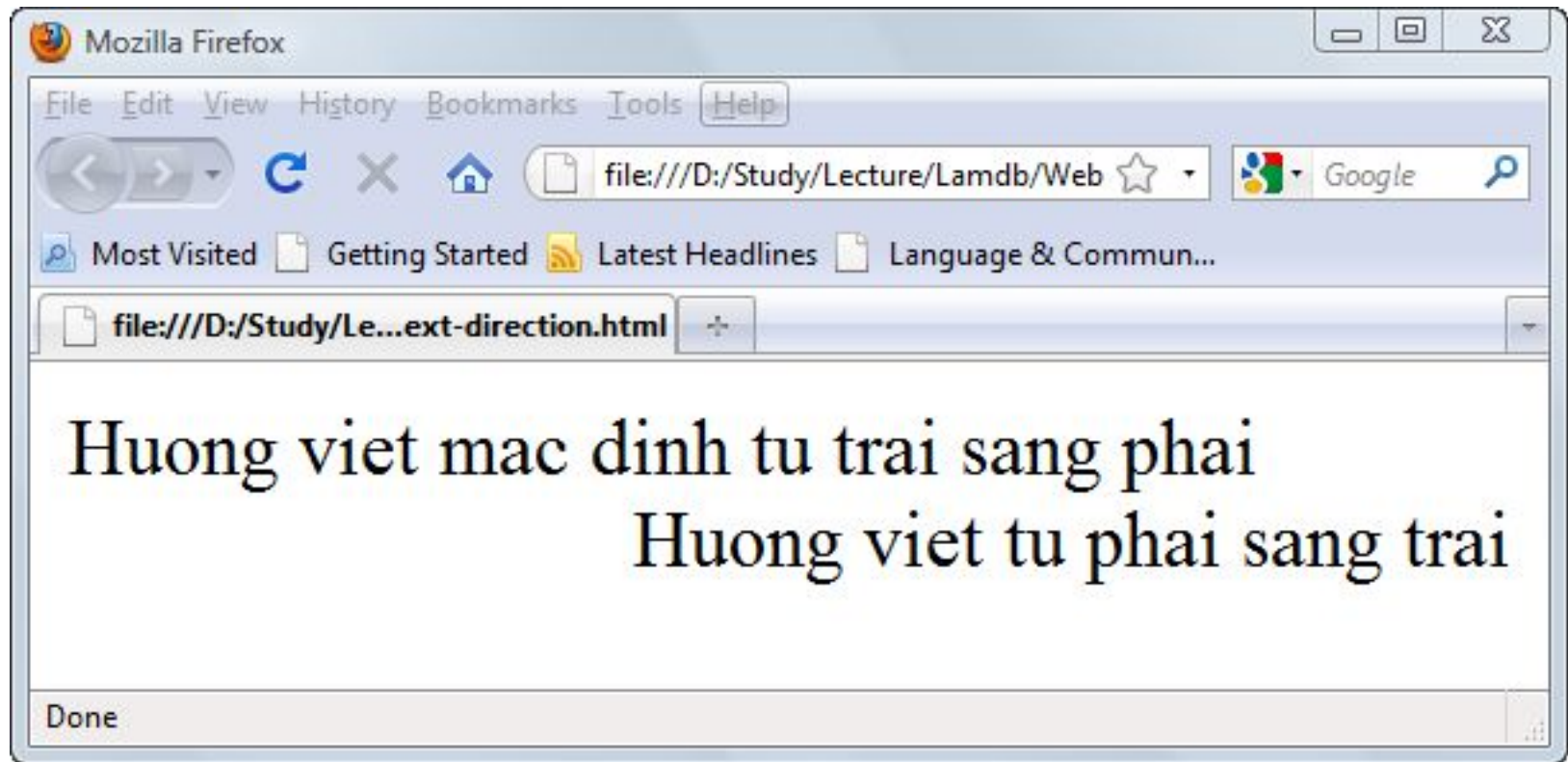
- Các hướng

- ltr: left to right. Chữ hiển thị từ bên trái sang phải
- rtl: right to left. Chữ hiển thị từ bên phải sang trái

- Ví dụ

```
<style type="text/css">  
    div.ex1 {direction:rtl}  
</style>
```

6.2. CSS Text





6.2. CSS Text

letter-spacing

- Tăng/giảm khoảng cách giữa các kí tự so với trạng thái thông thường (normal)
 - Giá trị có thể là số dương hoặc âm
- Ví dụ

```
<style type="text/css">
  h1 {letter-spacing:2px}
  h2 {letter-spacing:-3px}
</style>
```

6.2. CSS Text





6.2. CSS Text

text-align

- Căn nội dung văn bản theo chiều ngang: left, center, right, justify
- Ví dụ

```
<style type="text/css">
    h1 {text-align:center}
    h2 {text-align:left}
    h3 {text-align:right}
</style>
```

6.2. CSS Text





6.3. CSS Font

Property	Description	Values	NN	IE	W3C
font	A shorthand property for setting all of the properties for a font in one declaration	<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	A prioritized list of font family names and/or generic family names for an element	<i>family-name</i> <i>generic-family</i>	4.0	3.0	CSS1



6.3. CSS Font

font-size	Sets the size of a font	xx-small x-small small medium large x-large xx-large smaller larger <i>length</i> <i>%</i>	4.0	3.0	CSS1
font-size-adjust	Specifies an aspect value for an element that will preserve the x-height of the first-choice font	none <i>number</i>			CSS2



6.3. CSS Font

font-stretch	Condenses or expands the current font-family	normal wider narrower ultra-condensed extra-condensed condensed semi-condensed semi-expanded expanded extra-expanded ultra-expanded			CSS2
font-style	Sets the style of the font	normal italic oblique	4.0	4.0	CSS1



6.3. CSS Font

font-variant	Displays text in a small-caps font or a normal font	normal small-caps	6.0	4.0	CSS1
font-weight	Sets the weight of a font	normal bold bolder lighter 100 200 300 400 500 600 700 800 900	4.0	4.0	CSS1



6.3. CSS Font

font-family

- Các kiểu font

Difference Between Serif and Sans-serif Fonts

F

Sans-serif

F

Serif

F

Serif
(red serifs)

- Sans-serif: Arial, Verdana
- Serif: Times New Roman, Georgia



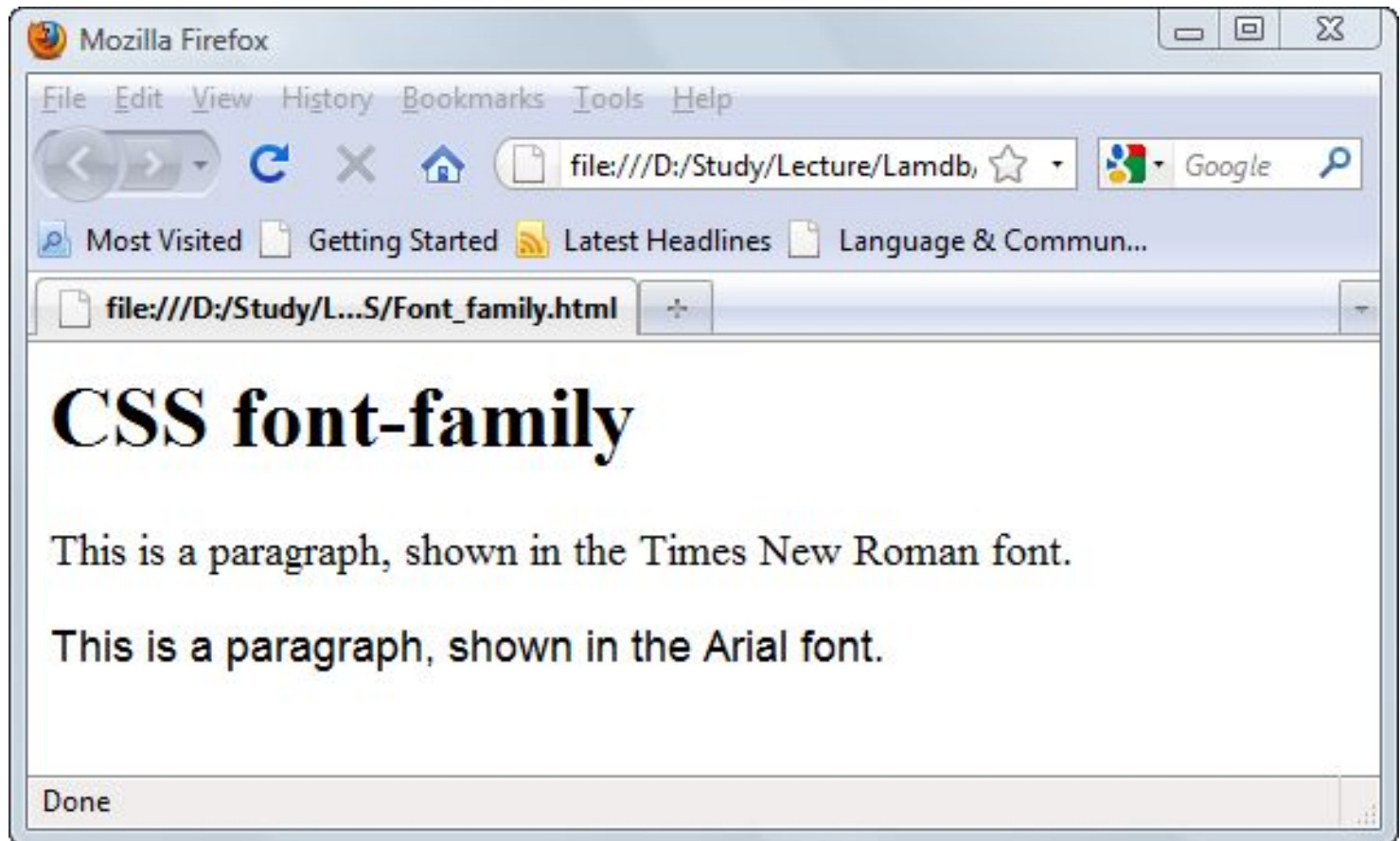
6.3. CSS Font

font-family

- Liệt kê các font sử dụng
- Ngăn cách nhau bởi dấu ,
- Ví dụ

```
<style type="text/css">  
    p{font-family:"Times New  
Roman", Georgia, Serif}  
</style>
```

6.3. CSS Font





6.3. CSS Font

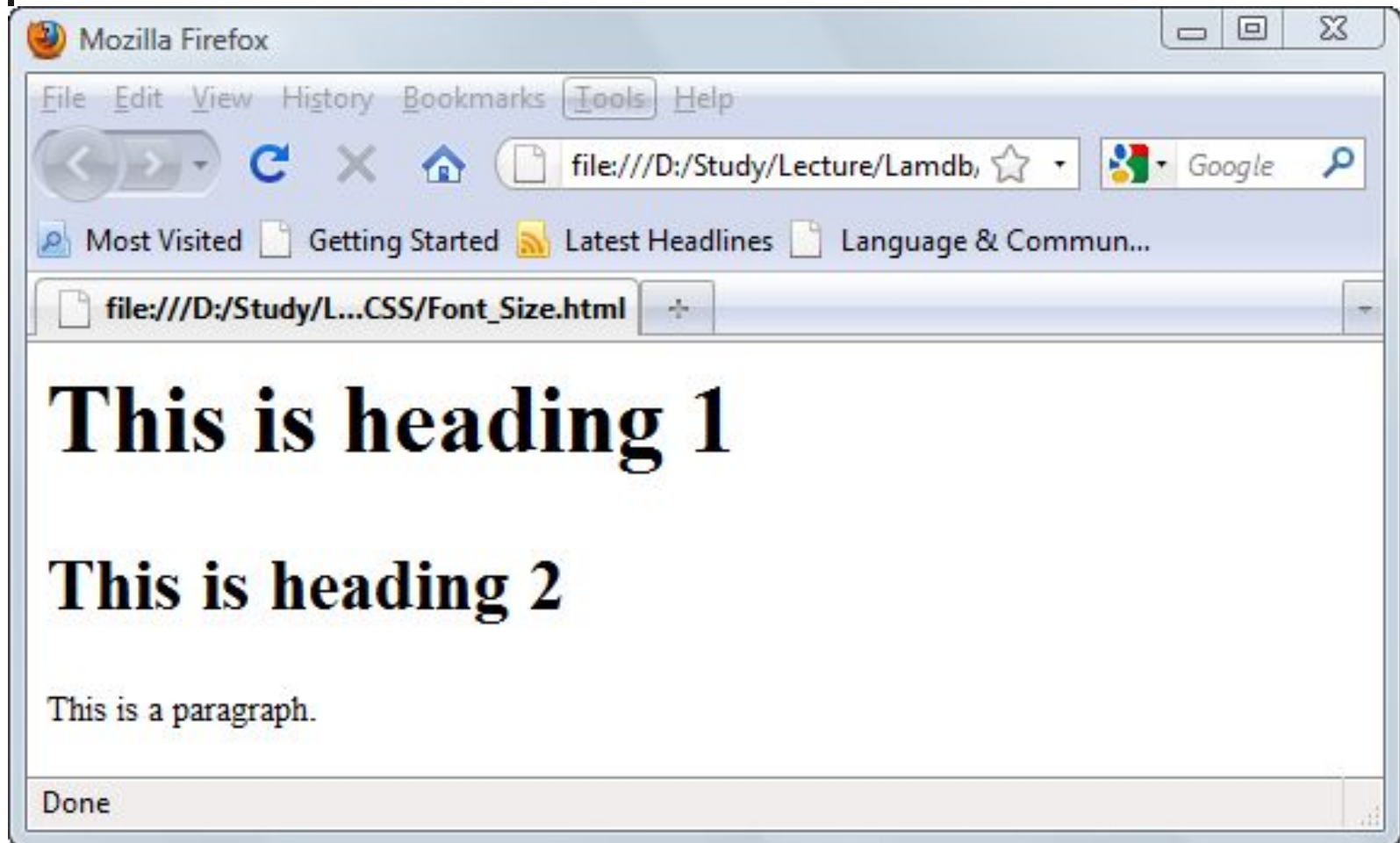
font-size

- Đặt kích thước cho văn bản. Mặc định kích thước thông thường là 16px
- Ví dụ

```
h1 {font-size:40px}  
h2 {font-size:30px}  
p {font-size:14px}
```

- Đơn vị khác: em, %

6.3. CSS Font





Thuộc tính rút gọn (shorthand)

- Chuyển:

```
h1 {  
    font-style: italic;  
    font-variant: small-caps;  
    font-weight: bold;  
    font-size: 35px;  
    font-family: arial, verdana, sans-serif;  
}
```

- Thành:

```
h1 {  
    font: italic bold 35px arial, verdana, sans-serif;  
}
```

- Cấu trúc rút gọn cho các thuộc tính nhóm font:

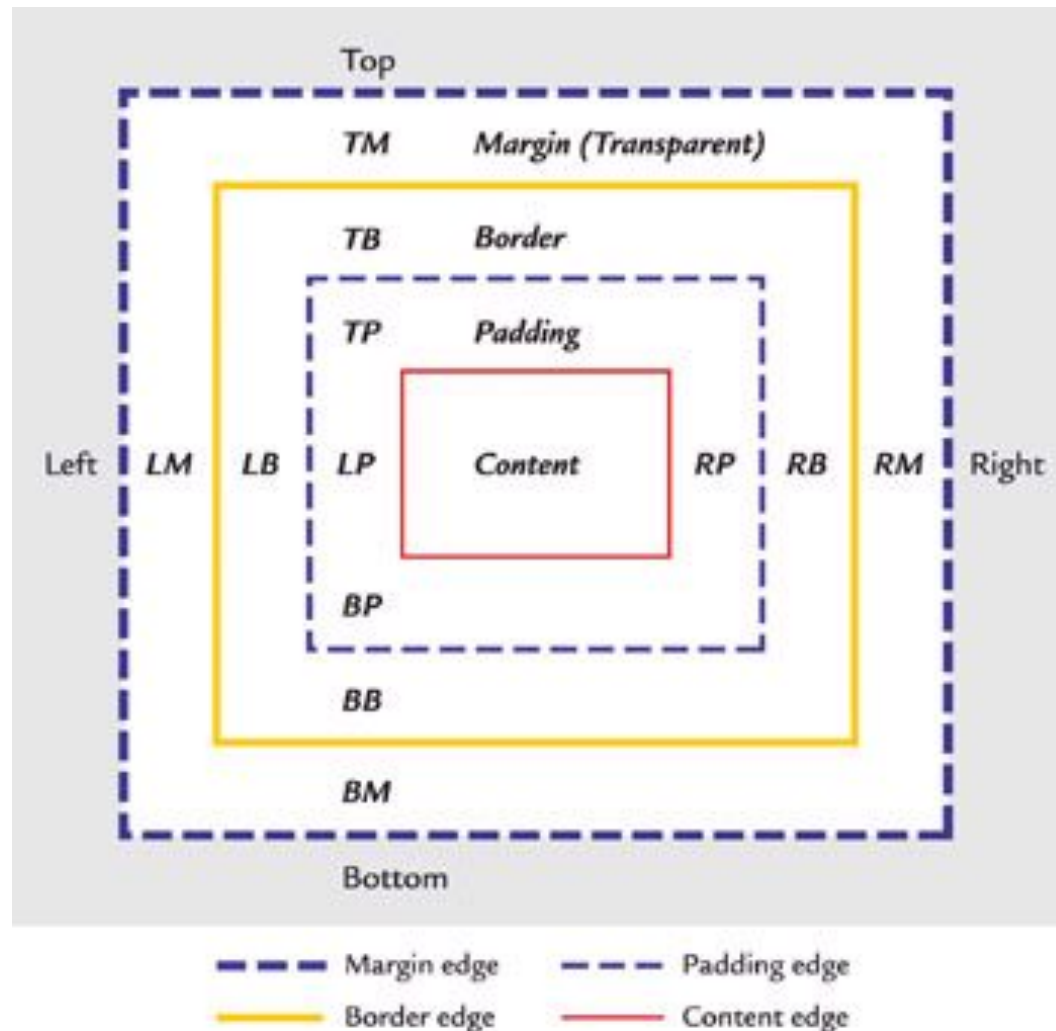
- Font :<font-style> | <font-variant> | <font-weight> | <font-size>
|<font-family>



Mô hình hộp

- Trong CSS, box model (mô hình hộp) mô tả cách mà CSS định dạng khối không gian bao quanh một thành phần.
- Mỗi một phần tử trong trang được xem như là một hình chữ nhật được tạo ra từ thành phần content, padding, border và margin.
 - Padding bao quanh phần nội dung. Thường được sử dụng để tạo nền cho một phần tử.
 - Border tạo ra đường viền bao quanh phần padding.
 - Margin trong suốt và không thể nhìn thấy, chúng được sử dụng để ngăn cách giữa các thành phần.
 - Padding, border, và margin là lựa chọn không bắt buộc và có thể nhận giá trị bằng 0.

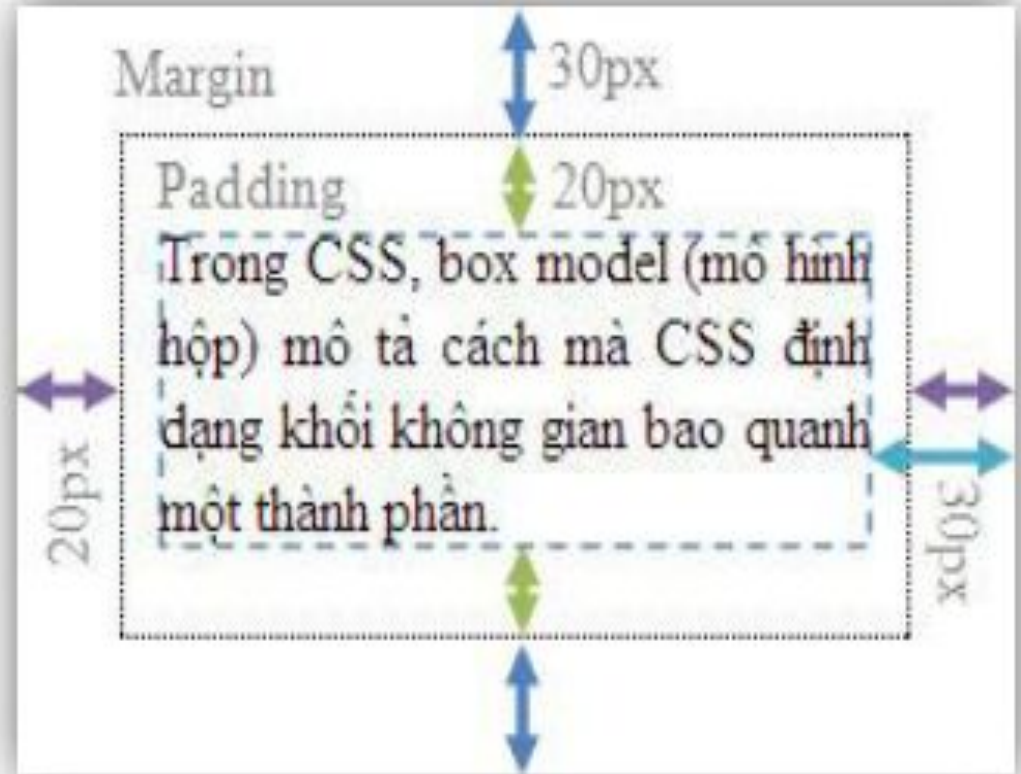
Mô hình hộp (con't)



Mô hình hộp

Ví dụ:

```
p {  
  width:200px;  
  margin:30px 20px;  
  padding:20px 10px;  
  border:1px solid  
    #000;  
  text-align:justify  
}
```



Mô hình hộp

■ Có 2 loại hộp:

- **Block-Level Elements:** có thể chứa các phần tử nội tuyến và các phần tử block-level khác. Block-box được sinh ra bởi thẻ p, div, table
- **Inline Elements** xuất hiện bên trong nội dung, và có thể chứa văn bản hoặc các phần tử inline khác. Ví dụ: a, em, strong, img, input.

Thuộc tính **display**

- **block:** sinh ra một block box
- **inline** : sinh ra một inline box
- **list-item:** sinh ra danh sách bulleted

```
p { display: block }
em { display: inline }
li { display: list-item }
img { display: none }
/* Không hiển thị ảnh*/
```



6.4. CSS Border

Property	Description	Values	NN	IE	W3C
border	A shorthand property for setting all of the properties for the four borders in one declaration	<i>border-width</i> <i>border-style</i> <i>border-color</i>	4.0	4.0	CSS1
border-bottom	A shorthand property for setting all of the properties for the bottom border in one declaration	<i>border-bottom-width</i> <i>border-style</i> <i>border-color</i>	6.0	4.0	CSS1
border-bottom-color	Sets the color of the bottom border	<i>border-color</i>	6.0	4.0	CSS2
border-bottom-style	Sets the style of the bottom border	<i>border-style</i>	6.0	4.0	CSS2
border-bottom-width	Sets the width of the bottom border	thin medium thick <i>length</i>	4.0	4.0	CSS1
border-color	Sets the color of the four borders, can have from one to four colors	<i>color</i>	6.0	4.0	CSS1



6.4. CSS Border

<code>border-left</code>	A shorthand property for setting all of the properties for the left border in one declaration	<i>border-left-width</i> <i>border-style</i> <i>border-color</i>	6.0	4.0	CSS1
<code>border-left-color</code>	Sets the color of the left border	<i>border-color</i>	6.0	4.0	CSS2
<code>border-left-style</code>	Sets the style of the left border	<i>border-style</i>	6.0	4.0	CSS2
<code>border-left-width</code>	Sets the width of the left border	thin medium thick <i>length</i>	4.0	4.0	CSS1

border-right	A shorthand property for setting all of the properties for the right border in one declaration	<i>border-right-width</i> <i>border-style</i> <i>border-color</i>	6.0	4.0	CSS1
border-right-color	Sets the color of the right border	<i>border-color</i>	6.0	4.0	CSS2
border-right-style	Sets the style of the right border	<i>border-style</i>	6.0	4.0	CSS2
border-right-width	Sets the width of the right border	thin medium thick <i>length</i>	4.0	4.0	CSS1
border-style	Sets the style of the four borders, can have from one to four styles	none hidden dotted dashed solid double groove ridge inset outset	6.0	4.0	CSS1

border-style

Viền Kiểu Solid

Viền Kiểu Dotted

Viền Kiểu Dashed

Viền Kiểu Double

Viền Kiểu Groove

Viền Kiểu Ridge

Viền Kiểu Inset

Viền Kiểu Outset



6.4. CSS Border

<code>border-top</code>	A shorthand property for setting all of the properties for the top border in one declaration	<i>border-top-width</i> <i>border-style</i> <i>border-color</i>	6.0	4.0	CSS1
<code>border-top-color</code>	Sets the color of the top border	<i>border-color</i>	6.0	4.0	CSS2
<code>border-top-style</code>	Sets the style of the top border	<i>border-style</i>	6.0	4.0	CSS2
<code>border-top-width</code>	Sets the width of the top border	thin medium thick <i>length</i>	4.0	4.0	CSS1
<code>border-width</code>	A shorthand property for setting the width of the four borders in one declaration, can have from one to four values	thin medium thick <i>length</i>	4.0	4.0	CSS



6.4. CSS Border

Property	Description	Values	NN	IE	W3C
<code>clear</code>	Sets the sides of an element where other floating elements are not allowed	left right both none	4.0	4.0	CSS1
<code>cursor</code>	Specifies the type of cursor to be displayed	<i>url</i> auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help	6.0	4.0	CSS2

display

Sets how/if an element is displayed

none
inline
block
list-item
run-in
compact
marker
table
inline-table
table-row-group
table-header-group
table-footer-group
table-row
table-column-group
table-column
table-cell
table-caption

4.0

4.0

CSS1



6.4. CSS Border

float	Sets where an image or a text will appear in another element	left right none	4.0	4.0	CSS1
position	Places an element in a static, relative, absolute or fixed position	static relative absolute fixed	4.0	4.0	CSS2
visibility	Sets if an element should be visible or invisible	visible hidden collapse	6.0	4.0	CSS2



6.4. CSS Border

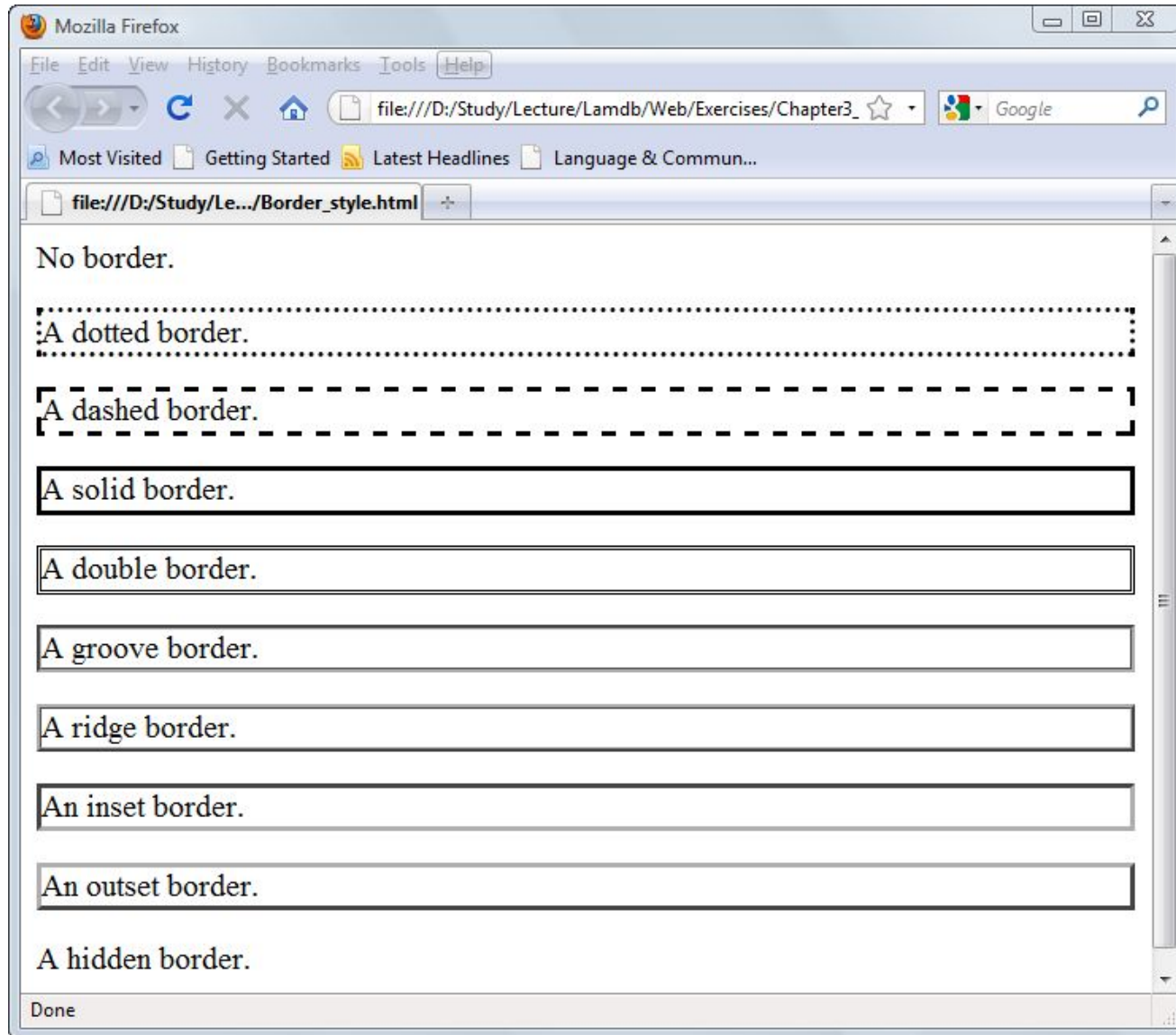
- Cho phép xác định màu và kiểu (style) trong đường viền của các thành phần

border-style

- Có thể xác định từ 1 đến 4 kiểu
- Ví dụ

```
border-style:dotted solid double dashed;  
border-style:dotted solid double;  
border-style:dotted solid;  
border-style:dotted;
```

6.4. CSS Border





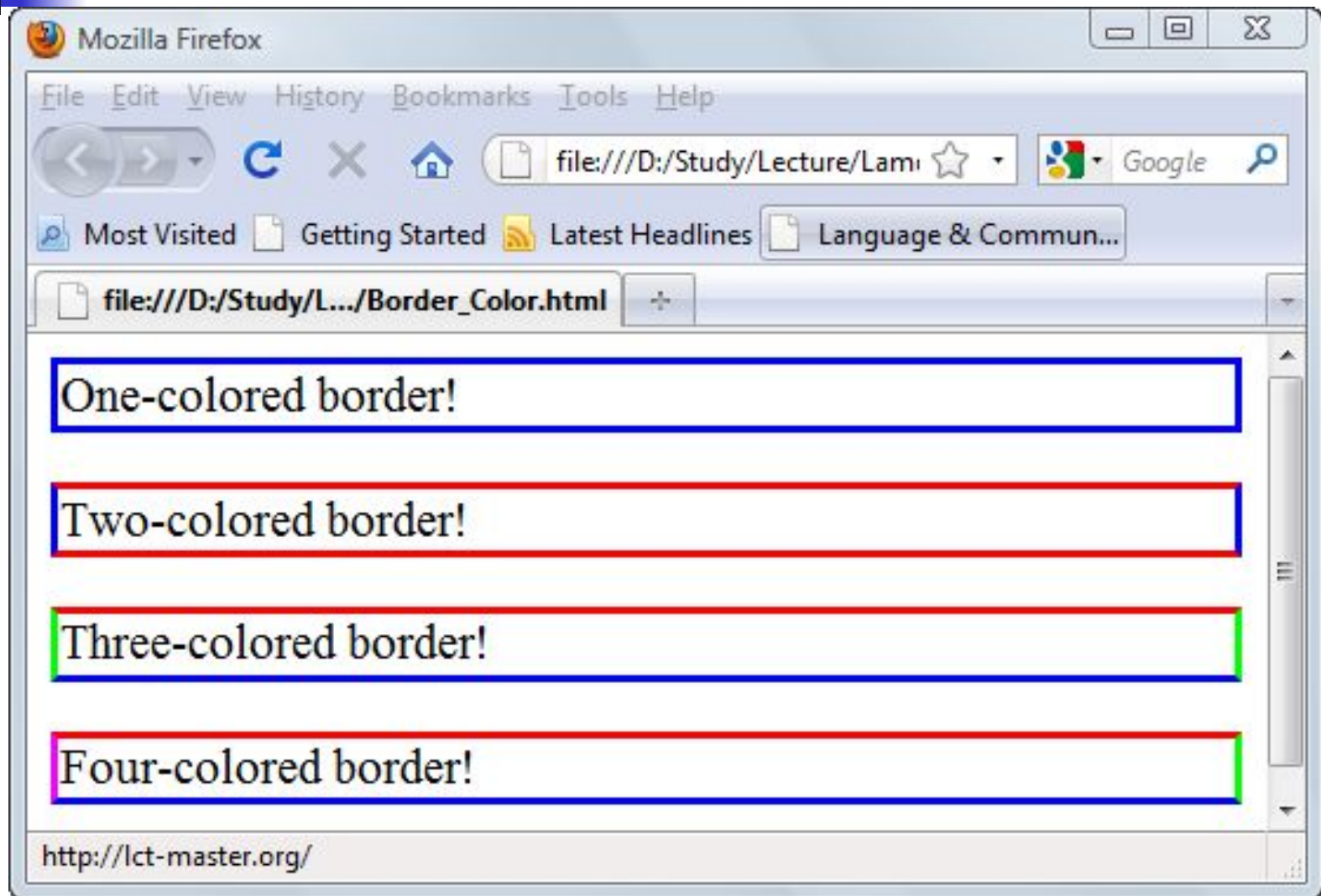
6.4. CSS Border

border-color

- Có thể xác định từ 1 đến 4 màu, giống style
- Phải đi kèm với border-style
- Ví dụ

```
border-color:red green blue pink;  
border-color:red green blue;  
border-color:red green;  
border-color:blue;
```


6.4. CSS Border





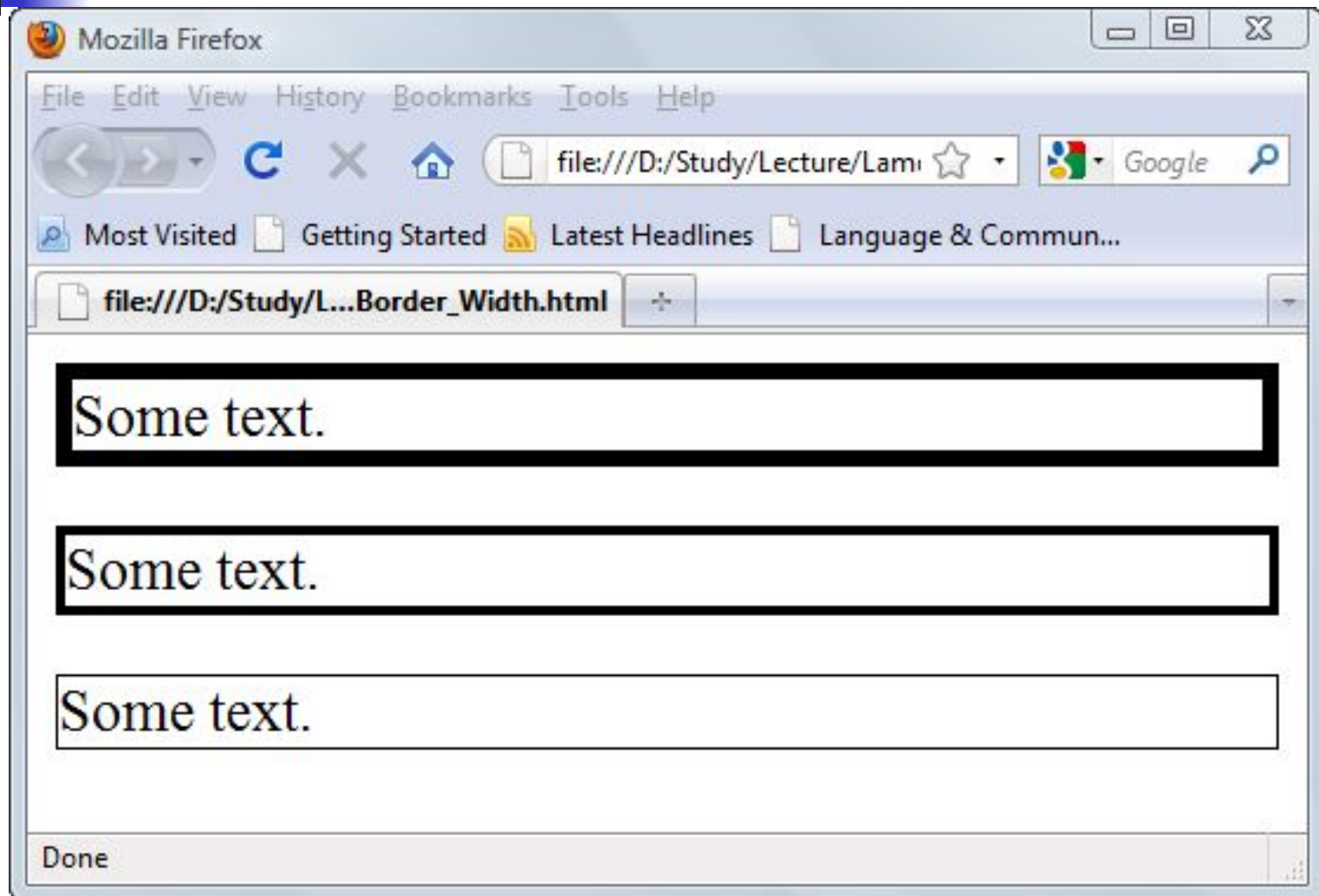
6.4. CSS Border

border-width

- Có thể xác định từ 1 đến 4 giá trị, giống color
- Phải đi kèm với border-style
- Ví dụ

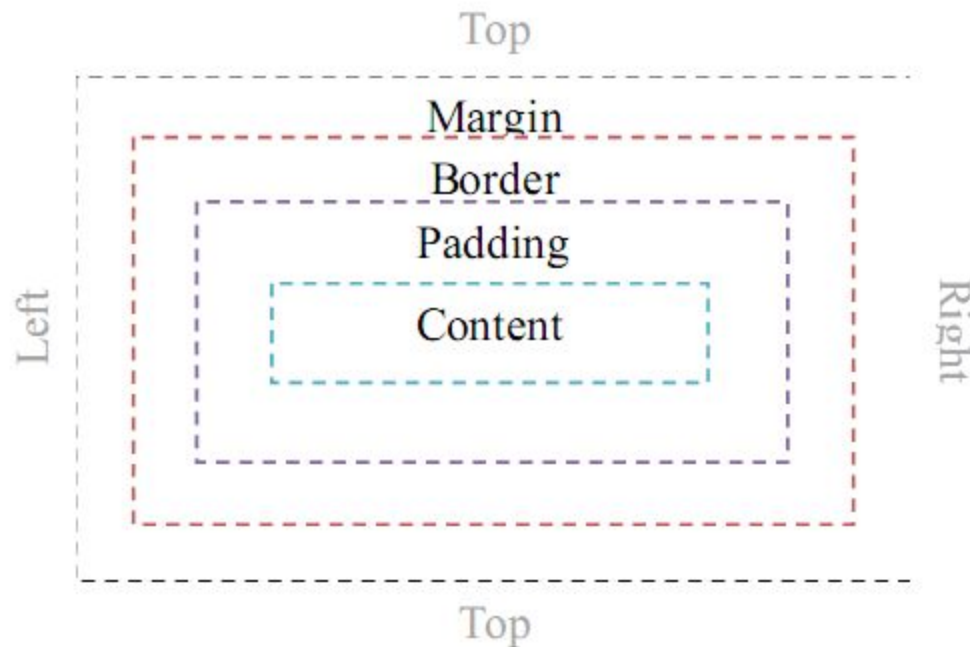
```
border-width:thin medium thick  
10px;  
border-width:thin medium thick;  
border-width:thin medium;  
border-width:thin;
```

6.4. CSS Border



6.5. CSS Margin

- Box model-Mô hình hộp

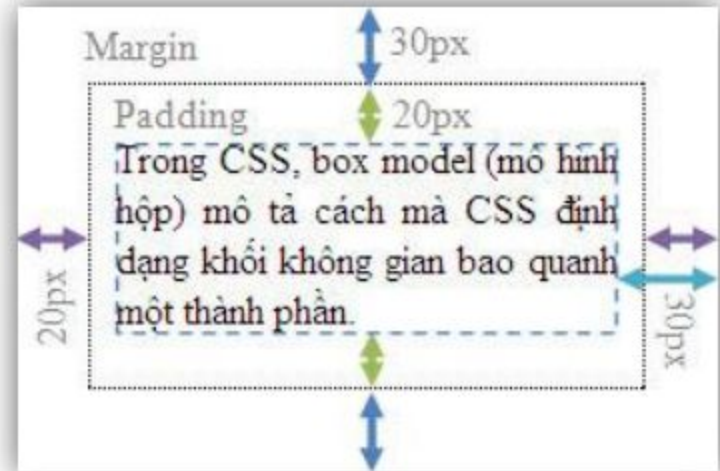


Box model

<p>Trong CSS, box model (mô hình hộp) mô tả cách mà CSS định dạng khối không gian bao quanh một thành phần.
</p>

■ Style:

```
p {  
  width:200px;  
  margin:30px 20px;  
  padding:20px 10px;  
  border:1px solid #000;  
  text-align:justify  
}
```





6.5. CSS Margin

Property	Description	Values	CSS
<u>margin</u>	A shorthand property for setting the margin properties in one declaration	<i>margin-top</i> <i>margin-right</i> <i>margin-bottom</i> <i>margin-left</i>	1
<u>margin-bottom</u>	Sets the bottom margin of an element	auto <i>length</i> %	1
<u>margin-left</u>	Sets the left margin of an element	auto <i>length</i> %	1
<u>margin-right</u>	Sets the right margin of an element	auto <i>length</i> %	1
<u>margin-top</u>	Sets the top margin of an element	auto <i>length</i> %	1



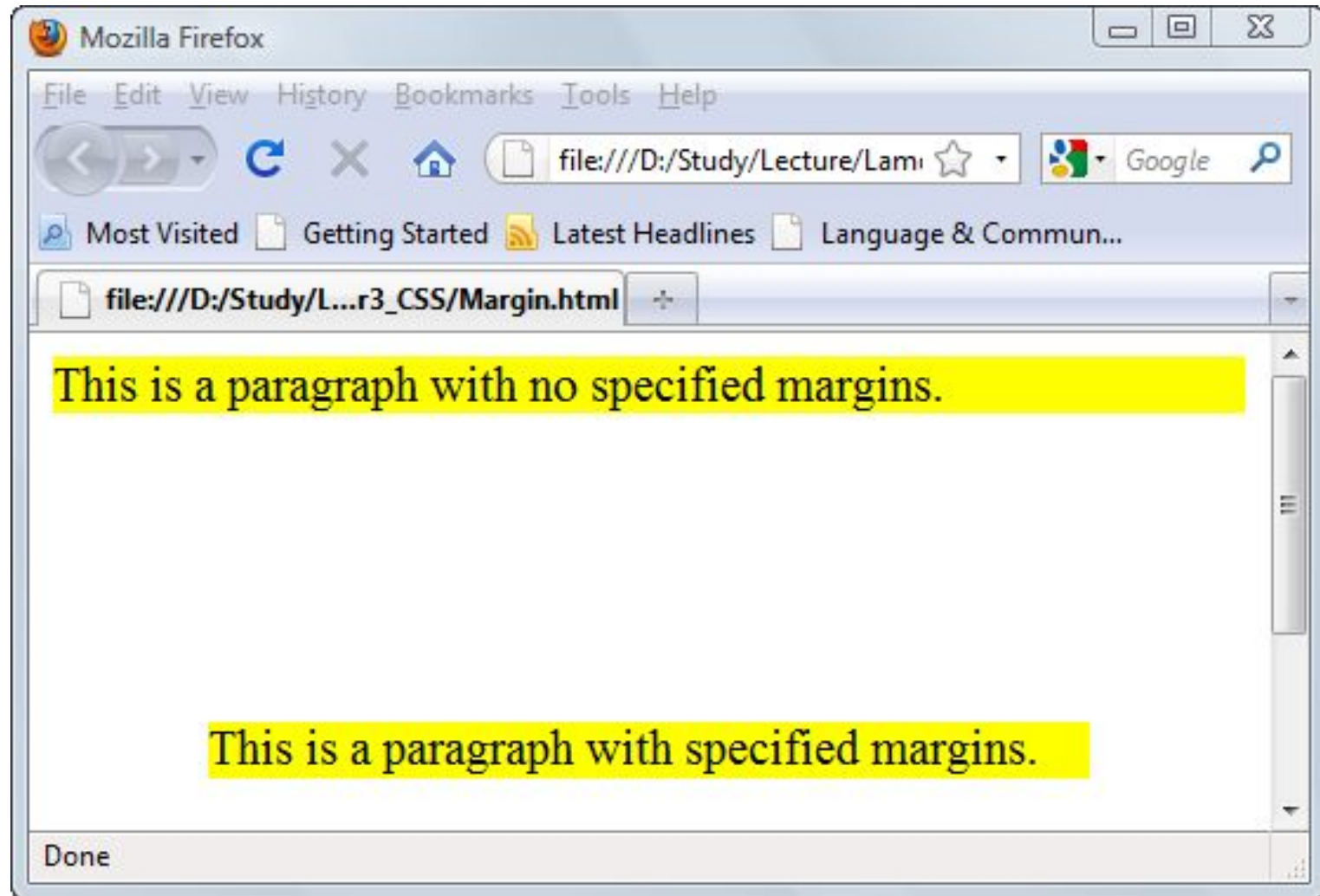
6.5. CSS Margin

margin

- Có thể xác định từ 1 đến 4 giá trị, giống với CSS border
- Ví dụ

```
margin:25px 50px 75px 100px;  
margin:25px 50px 75px;  
margin:25px 50px;  
margin:25px;
```

6.5. CSS Margin



6.6. Float & Clear

- Ứng dụng: Chia layout
- 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



6.6. Float & Clear

- Thuộc tính clear: quyết định Content có tràn lên không
- Thuộc tính clear có tất cả 4 giá trị:
 - left (tràn bên trái)
 - right (tràn bên phải)
 - both (không tràn)
 - none.





6.7. CSS khác

- CSS Padding: thiết lập khoảng cách giữa nội dung và đường viền
- CSS List: thiết lập vị trí, kiểu, ảnh của mỗi thành phần trong danh sách
- CSS Table: thiết lập cách hiển thị bảng như khoảng cách giữa các ô, chú thích



5. Lược đồ vị trí trong CSS

- **Normal Flow** : theo mặc định của trình duyệt đối với những phần tử không thiết lập position:absolute hoặc fixed và không float
- **Floats**: được thiết lập theo chế độ của normal flow sau đó được dịch trái hoặc phải
- **Absolute Positioning** : liên quan tới vị trí tuyệt đối của phần tử liên quan tới khối nội dung
- **Relative Positioning** : Lược đồ vị trí được sử dụng bởi hai thuộc tính position và float



5. Lược đồ vị trí trong CSS

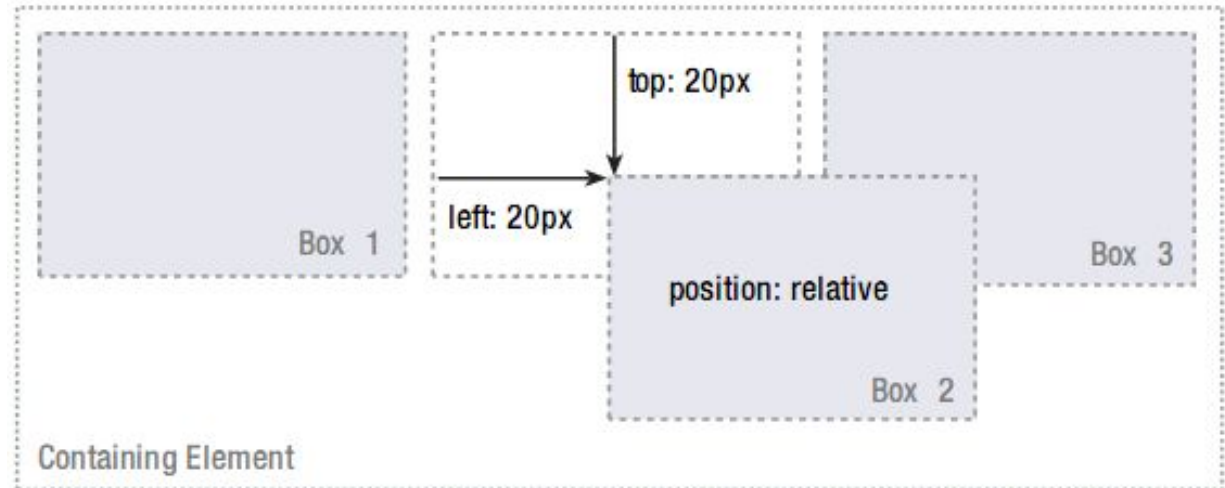
Thuộc tính **position**

- **Static:** Box là một box cơ bản, nằm theo normal flow. Thuộc tính left và top không có tác dụng
- **Relative:** Vị trí box được tính toán theo normal flow. Box sẽ được dịch chuyển so với normal flow.
- **Absolute:** Vị trí của box được xác định bởi các thuộc tính top, bottom, left, right không phụ thuộc vào normal flow
- **Fix:** Được tính giống như absolute nhưng sẽ cố định.

5.1. Relative positioning

- Một box ở vị trí relative được sinh ra khi thuộc tính **position** có giá trị **relative**.
- Sự dịch chuyển so với normal flow được xác định bởi các thuộc tính 'top' Sự dịch chuyển so với normal flow được xác định bởi các thuộc tính 'top', 'bottom' Sự dịch chuyển so với normal flow được xác định bởi các thuộc tính 'top', 'bottom', 'left' Sự dịch chuyển so với normal flow được xác định bởi các thuộc tính 'top', 'bottom', 'left', và 'right'.

```
#myBox {  
position: relative;  
left: 20px;  
top: 20px;  
}
```



5.1. Relative positioning

FLOATS

- Một float là một box được dịch chuyển sang trái hoặc phải so với dòng hiện thời.
- Đặc tính đáng quan tâm nhất của float đó là nội dung có thể chảy theo các phía của nó
- Nội dung chảy xuống bên phải của hộp định trái và bên trái của hộp định phải





5.1. Relative positioning

Thuộc tính **float**

- *Left*: Nội dung chạy bên phải của box, bắt đầu từ đỉnh. Thuộc tính display sẽ bị loại bỏ trừ khi giá trị display được gán bằng none.
- *Right*: Tương tự như left, nhưng nội dung theo bên trái của box, bắt đầu từ đỉnh.
- None



5.1. Relative positioning

Thuộc tính **float**

Ví dụ: thực hiện float ảnh logo sang trái để phần nội dung bên dưới tràn lên nằm cạnh logo.

```
#logo { float:left; }
```

Ví dụ: dùng float để chia 2 cột văn bản.

```
.column1
{
    width:45%;
    float:left;
    text-align:justify;
    padding:0 20px;
}
.column2
{ border-right:1px
    solid #000
}
```




5.1. Relative positioning

Thuộc tính **clear**:

- Thuộc tính clear là một thuộc tính thường được gán vào các phần tử liên quan tới phần tử đã được **float** để quyết định hướng xử sự của phần tử này.
- Ở ví dụ trên, khi chúng ta float tấm ảnh qua trái thì mặc nhiên văn bản sẽ được tràn lên để lấp vào chỗ trống. Nhưng khi chúng ta đặt vào văn bản thuộc tính **clear** thì chúng ta có quyền quyết định xem phần văn bản đó có được tràn lên hay không.



5.1. Relative positioning

Thuộc tính clear:

- Thuộc tính clear có tất cả 4 thuộc tính:
 - left (tràn bên trái),
 - right (tràn bên phải),
 - both (không tràn) và
 - none.



5.2. Absolute positioning

- Một phần tử ở vị trí tuyệt đối được bố trí liên quan đến cha có vị trí gần nhất.
- Nội dung của phần tử định vị tuyệt đối không bị ảnh hưởng bởi các bất kì box nào.
- Sử dụng giá trị absolute và fixed của thuộc tính position
- Giả sử chúng ta muốn định vị một ảnh ở vị trí 70px cách đỉnh và 90px từ bên trái tài liệu, chúng ta sẽ viết CSS như sau:

```
img { position:absolute; top:70px; left:90px }
```



5.2. Absolute positioning

- Ví dụ: đặt bốn ảnh ở bốn góc tài liệu bằng định vị tuyệt đối.

```
#logo1 { position:absolute; top:50px; left:70px }  
#logo2 { position:absolute; top:0; right:0 }  
#logo3 { position:absolute; bottom:0; left:0 }  
#logo4 { position:absolute; bottom:70px; right:50px }
```



5.3. Layer

- Cách đặt các thành phần web ở các lớp khác nhau với thuộc tính z-index.
- Nói đơn giản hơn thì đó là cách đặt một thành phần này lên trên một thành phần khác.
- Với mục đích này, gán cho mỗi phần tử một con số. Theo đó, phần tử có số cao hơn sẽ nằm trên, phần tử có số thấp hơn sẽ nằm dưới.

5.3. Layer

- Ví dụ:

```
#apples {
```

```
  position: absolute;
```

```
  top: 20px;
```

```
  left: 20px;
```

```
  width: 200px;
```

```
  z-index: 1;
```

```
}
```

```
#oranges {
```

```
  position: absolute;
```

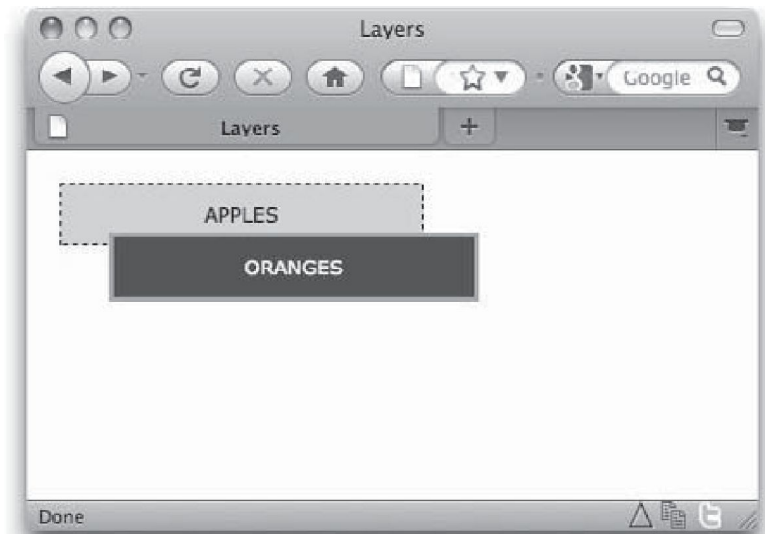
```
  top: 50px;
```

```
  left: 50px;
```

```
  width: 200px;
```

```
  z-index: 2;
```

```
}
```





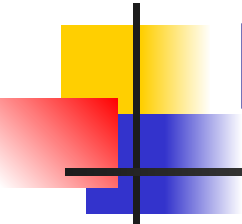
Ví dụ xây dựng layout web đơn giản với thẻ div

- Bố cục của trang gồm 4 phần
 - header, sidebar, main, footer
- Mỗi phần được xác định bằng thẻ div và thuộc tính id

```
<div id="header">  
  <p>Header content</p>  
</div>  
<div id="sidebar">  
  <p>Sidebar content</p>  
</div>  
<div id="main">  
  <h1>Main content</h1>  
</div>  
<div id="footer">  
  <p>Footer content</p>  
</div>
```

Trang Web ban
đầu

header
main
sidebar
footer



Layout (1): header

- margin và padding thiết lập bằng 0 (**reset CSS**)
- Xác định CSS chung cả trang và cho phần header

```
* {  
  margin: 0;  
  padding: 0;  
}  
body {  
  background-color: white;  
  color: black;  
}  
div#header {  
  background-color: red;  
  color: white;  
}
```

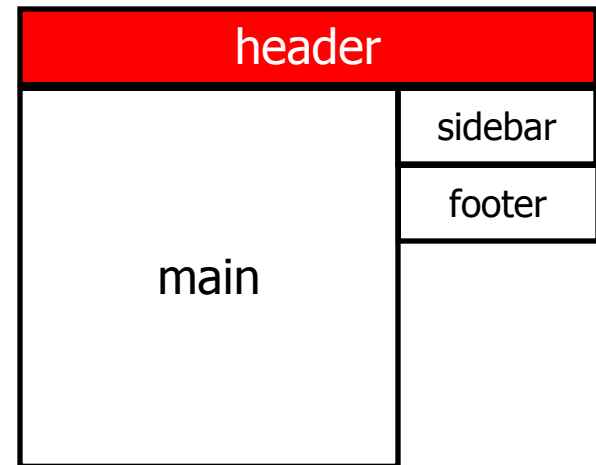
header
main
sidebar
footer



Layout (2): main

- Xác định CSS cho phần main

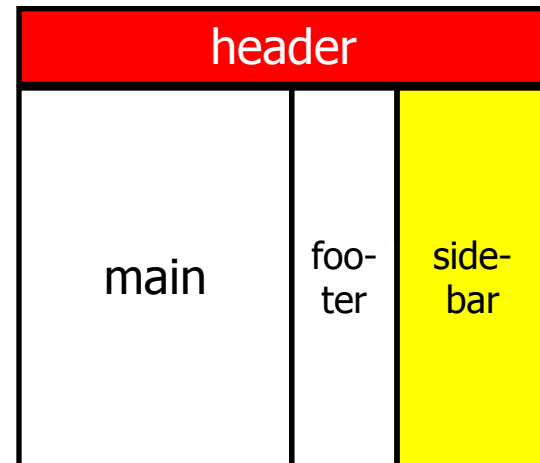
```
div#main {  
  float: left;  
  height: 400px;  
}
```



Layout (3): sidebar

- Xác định CSS cho phần sidebar
 - Thiết lập chiều cao và đẩy sang bên phải
 - Giới hạn chiều rộng sidebar là 25% phần chứa nó

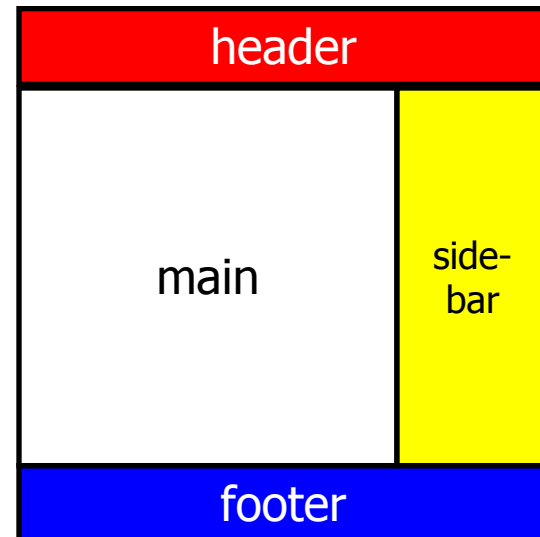
```
div#sidebar {  
  float: right;  
  height: 400px;  
  width: 25%;  
  background-color: yellow;  
  color: black;  
}
```



Layout (4): footer

- Xác định CSS cho phần footer
- Sử dụng "clear: both;"

```
div#footer {  
  clear: both;  
  background-color: blue;  
  color: white;  
}
```





Bài tập 0

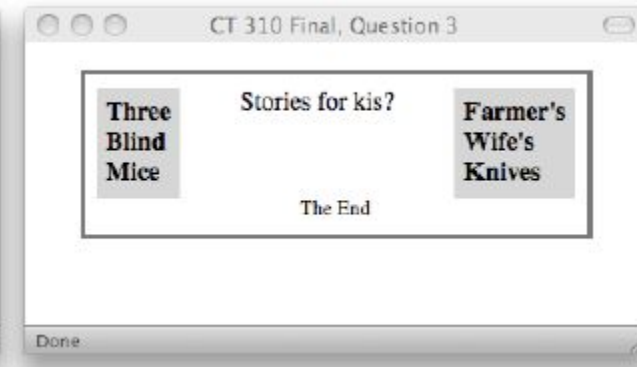
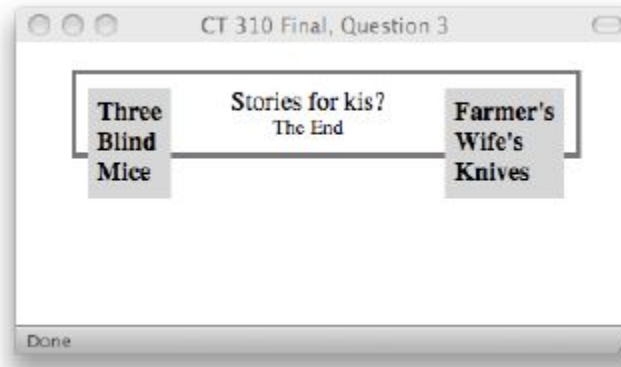
- Bạn hãy viết đoạn mã CSS để trình bày style như mô tả dưới đây:
 - Màu nền cho cả trang: #99CC66
 - Font chữ: Arial
 - Tiêu đề của bảng: chữ in nghiêng, màu chữ là màu xanh, màu nền là #CCCCFF

Bài tập 1

- Viết đoạn mã tài liệu HTML để tạo một trang đăng nhập. Trang gồm có 2 ô textbox nhập username, password, một nút “Sign in”, một nút “Cancel” và một ô check box: “Ghi nhớ mật khẩu”.
- Người ta muốn bổ sung thêm các style để trình bày trang Web tốt hơn:
 - Màu nền cho cả trang: #99CC66
 - Font chữ: Arial, được in đậm và có màu xanh
 - Các ô textbox để nhập thông tin có màu nền là màu vàng.
 - Một khung bao xung quanh form đăng nhập: nét đứt, màu đỏ
- Bạn hãy viết đoạn mã CSS để trình bày style như mô tả trên./.

Bài tập

Sửa lại CSS cho phù hợp



```
<head>
<title>CT 310 Final, Question
3</title>

<link href="./final09q3.css"
      rel="stylesheet"
      type="text/css" />

</head>
<body>
<div id="contents">
  <div id="logo1"> Three <br/>
                        Blind <br/>
                        Mice </div>

  <div id="logo2"> Farmer's <br/>
                        Wife's <br/>
                        Knives </div>

  <div id="toptext">
    Stories for kis? </div>
  <div id="bottomtext"> The End
</div>
</div>
</body>
</html>
```

```
#contents {
  border-style:solid;
  border-color:#666;
  padding:8px;
  margin-left:auto;
  margin-right:auto;
  margin-top:16px;
  margin-bottom:16px;
  width:80%; }

#logo1 {
  float:left;
  font-weight:bold;
  border-color:black;
  background-color:#CCC;
  padding:6px; }

#logo2 {
  float:right;
  font-weight:bold;
  border-color:black;
  background-color:#CCC;
  padding:6px; }

#toptext {
  text-align:center;
  font-size:medium; }

#bottomtext {
  text-align:center;
  font-size:small; }
```



Bài tập 3

■ Hoàn thiện CSS

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
<title>CT 310 Final, Question 2 - Style 1</title>
<link href="./final09q2style1.css" rel="stylesheet"
      type="text/css" />
</head>
<body>
  <div id="contents">
    <div id="John"> John </div>
    <div id="Paul"> Paul </div>
    <div id="George"> George </div>
    <div id="Ringo"> Ringo </div>
  </div>
</body>
</html>
```

```
#John {
  ...
}
#Paul {
  ...
}
#George {
  ...
}
#Ringo {
  ...
}
```



Bà

CT 310 Final, Question 2 – Style 7

John	Paul	George
Ringo		

CT 310 Final, Question 2 – Style 2

John	Paul	George
Ringo		

CT 310 Final, Question 2 – Style 5

John	Paul	George
	Ringo	

CT 310 Final, Question 2 – Style 8

John	Paul
George	Ringo

CT 310 Final, Question 2 – Style 1

John
Paul
George
Ringo

CT 310 Final, Question 2 – Style 3

John	George	Paul
	Ringo	

CT 310 Final, Question 2 – Style 4

John	George	Paul
Ringo		

CT 310 Final, Question 2 – Style 6

John	Paul	George
	Ringo	

Done

Done



7. Trắc nghiệm

- Nguồn: w3schools.com
- File
[..\Quiz.doc](#)

Câu hỏi

