

CHƯƠNG 2

NGÔN NGỮ HTML

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

Nội dung

1. Tổng quan về ngôn ngữ HTML
2. Cấu trúc tổng quát trang HTML
3. Các thẻ HTML thông dụng
4. Các thẻ tạo biểu mẫu
5. Một số thẻ HTML đặc biệt

Lịch sử phát triển của công nghệ Web

- 1991 HTML (Tim Berners-Lee tạo 20 thẻ HTML đầu tiên)
- 1994 HTML 2
- 1996 CSS1 + JAVASCRIPTS
- 1997 HTML 4
- 1998 CSS 2
- 2000 XHTML 1
- 2002 Tableless Web Design
- 2005 AJAX
- 2009 HTML 5

1. Tổng quan về HTML

□ HTML (HyperText Markup Language) là gì?

- Ngôn ngữ đánh dấu siêu văn bản.
- HTML không phải là ngôn ngữ lập trình
- Là ngôn ngữ để viết các trang web.
- Do Tim Berner Lee phát minh và được W3C (World Wide Web Consortium) đưa thành chuẩn năm 1994.

1. Tổng quan về HTML

- HTML sử dụng các thẻ (tags) để **định dạng** dữ liệu
- HTML không phân biệt chữ hoa, chữ thường
- Các trình duyệt thường không báo lỗi cú pháp HTML. Nếu viết sai cú pháp chỉ dẫn đến kết quả hiển thị không đúng với dự định.
- Có nhiều thẻ, mỗi thẻ có 1 tên và mang ý nghĩa khác nhau.

1. Tổng quan về HTML

□ Các element được phân loại:

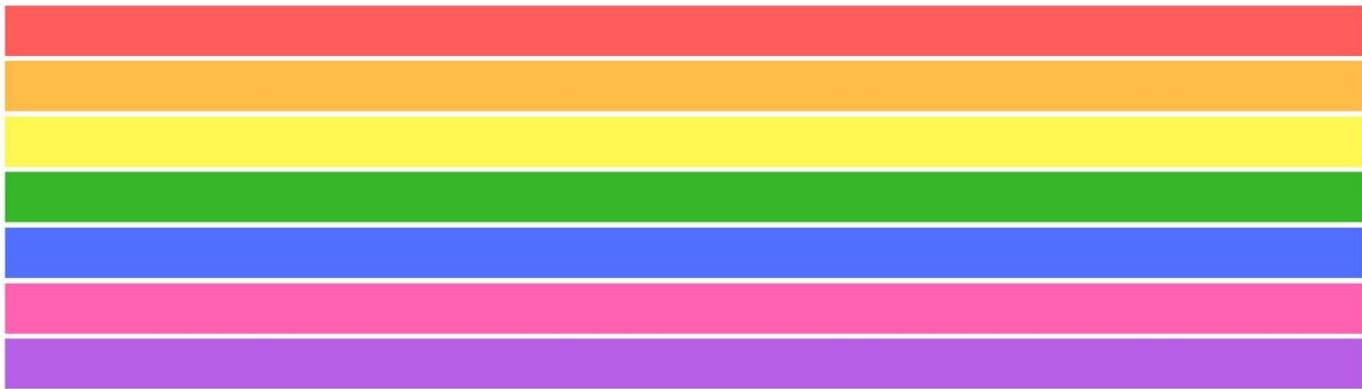
- *element khối (block-level element)*
- *element dòng (inline element)*

1. Tổng quan về HTML

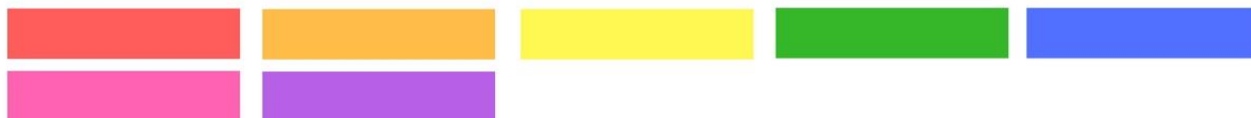
❑ **Block-level element** : nội dung của element khi hiển thị trên trình duyệt sẽ tự xuống dòng để phân cách với các element khác (ví dụ: `<h1>...</h1>` hoặc `<p>...</p>`)

❑ **Inline element** : nội dung của element sẽ tiếp tục ở trên dòng hiển thị của browser (ví dụ: `...`)

BLOCK-LEVEL ELEMENTS:



INLINE ELEMENTS:



1. Tổng quan về HTML

❑ Cách viết thẻ:

<tênthẻ> Dữ liệu </tênthẻ>

▪ Ví dụ:

Học về HTML

▪ Chú ý:

- Tên thẻ bắt nguồn từ một từ tiếng Anh:

B ~ Bold, I ~ Italic

- Luôn có thẻ mở nhưng có thể không có thẻ đóng tương ứng:
 ,
...

1. Tổng quan về HTML

- Một thẻ có thể có các thuộc tính nhằm bổ sung tác dụng cho thẻ.

Viết thẻ có thuộc tính:

<tênthẻ thuộctính1 = "giátrị1" thuộctính2 = "giátrị2"...>

<Dữ liệu>

</tênthẻ>

Ví dụ:

Học về HTML

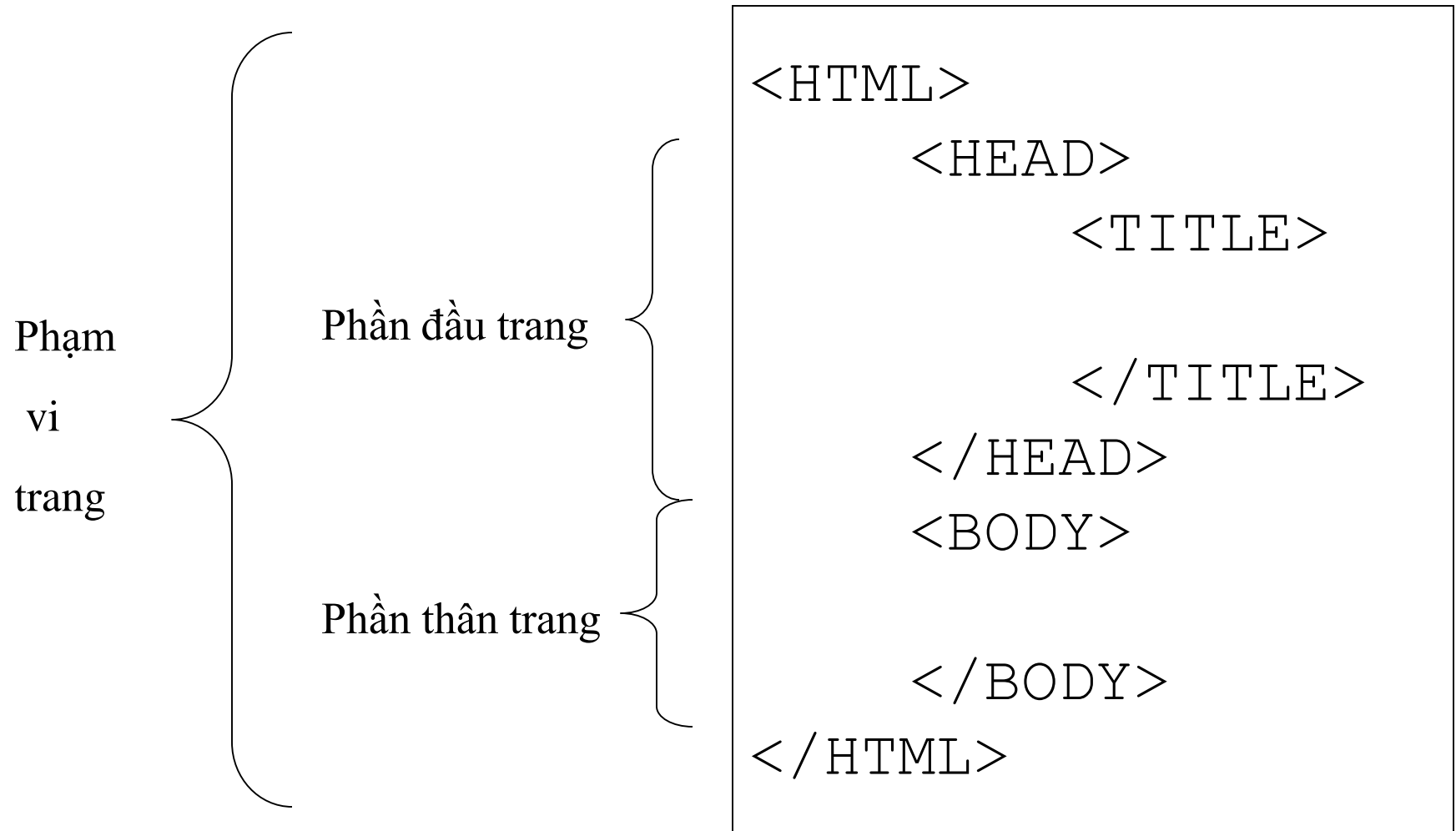
Chú ý: Có thể thay đổi thứ tự, số lượng các thuộc tính mà không gây ra lỗi cú pháp.

1. Tổng quan về HTML

- ❑ Trang web đầu tiên: Chao.htm

```
<HTML>
  <HEAD>
    <TITLE>Xin chào</TITLE>
  </HEAD>
  <BODY>
    Chào mừng đến với HTML!
  </BODY>
</HTML>
```

2. Cấu trúc tổng quát trang HTML



2. Cấu trúc tổng quát trang HTML

- **Phần HTML**: thông báo cho trình duyệt biết nội dung giữa hai thẻ là một tài liệu HTML
- **Phần đầu trang**: chứa các mô tả về trang: tiêu đề trang...
- **Phần thân văn bản**: nội dung trang như văn bản, hình ảnh, liên kết muốn hiển thị trên trang Web.

3. Các thẻ HTML thông dụng

3.1. Các thẻ xử lý văn bản

3.2. Các thẻ tạo bảng

3.3. Thẻ liên kết

3.4. Các thẻ đa phương tiện

3.5. Các thẻ tạo khung

3.1. Các thẻ xử lý văn bản

- **Các thẻ định dạng khối văn bản:**
 - Tiêu đề (Heading): `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`
 - Đoạn văn bản (Paragraph): `<p>`
 - Danh sách (List Items): ``
 - Đường kẻ ngang (Horizontal Rules): `<hr />`
- **Các thẻ định dạng chuỗi văn bản**
 - Định dạng chữ: ``, `<i>`, `` và ``
 - Tạo siêu liên kết: `<a>`
 - Xuống dòng: `
`

3.1. Các thẻ xử lý văn bản

❑ Thẻ <DIV>...</DIV>

- Cho phép định dạng một đoạn văn bản bằng các thuộc tính của chúng.
- Thuộc tính: align căn lề (center, justify...)

Cú pháp:

<DIV>Nội dung đoạn văn bản</DIV>

❑ Đoạn văn – Paragraph <p>

- Tương tự như thẻ <DIV>
- Sau thẻ </P> dữ liệu tự động xuống hàng

Cú pháp:

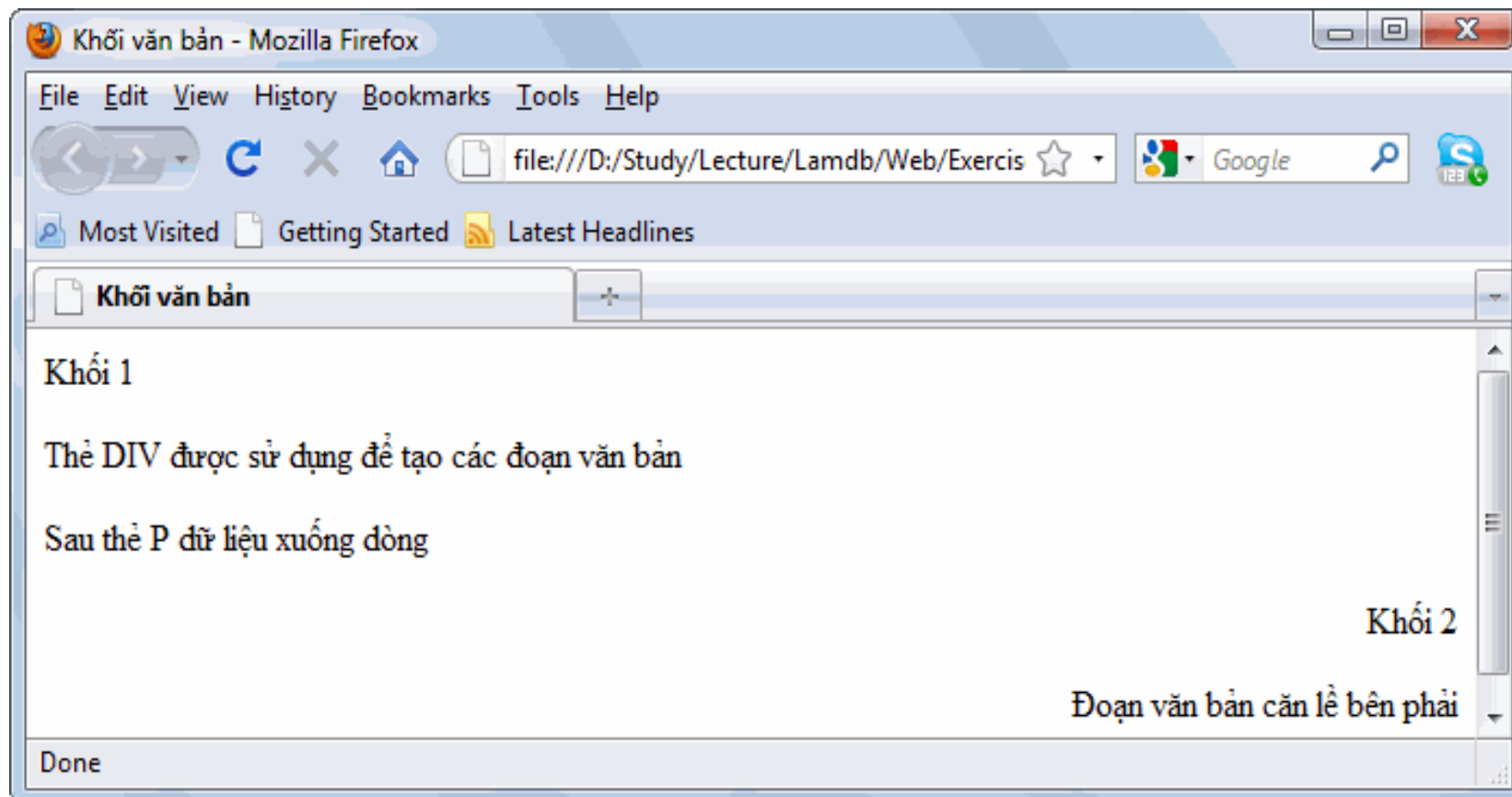
<P>Nội dung đoạn văn bản</P>

3.1. Các thẻ xử lý văn bản

❑ Ví dụ:

```
<HTML>
  <HEAD>
    <TITLE>Khối văn bản</TITLE>
  </HEAD>
  <BODY>
    <DIV>Khối 1
      <P>Thẻ DIV được sử dụng để tạo các đoạn văn bản</P>
      <P>Sau thẻ P dữ liệu xuống dòng</P>
    </DIV>
    <DIV align="right">Khối 2
      <P>Đoạn văn bản căn lề bên phải</P>
    </DIV>
  </BODY>
</HTML>
```


3.1. Các thẻ xử lý văn bản



3.1. Các thẻ xử lý văn bản

❑ Đường kẻ ngang – Horizontal rules <hr/>

- Tạo đường kẻ ngang, không có thẻ đóng
- Thuộc tính:
 - ✓ **align**: canh hàng đường kẻ ngang so với trang web
 - ✓ **width**: chiều dài đường kẻ ngang
 - ✓ **size**: độ dày của đường kẻ ngang
 - ✓ **noshade**: không có bóng

❑ Xuống dòng – Break line

- Xuống dòng, không có thẻ đóng

3.1. Các thẻ xử lý văn bản

□ Thẻ

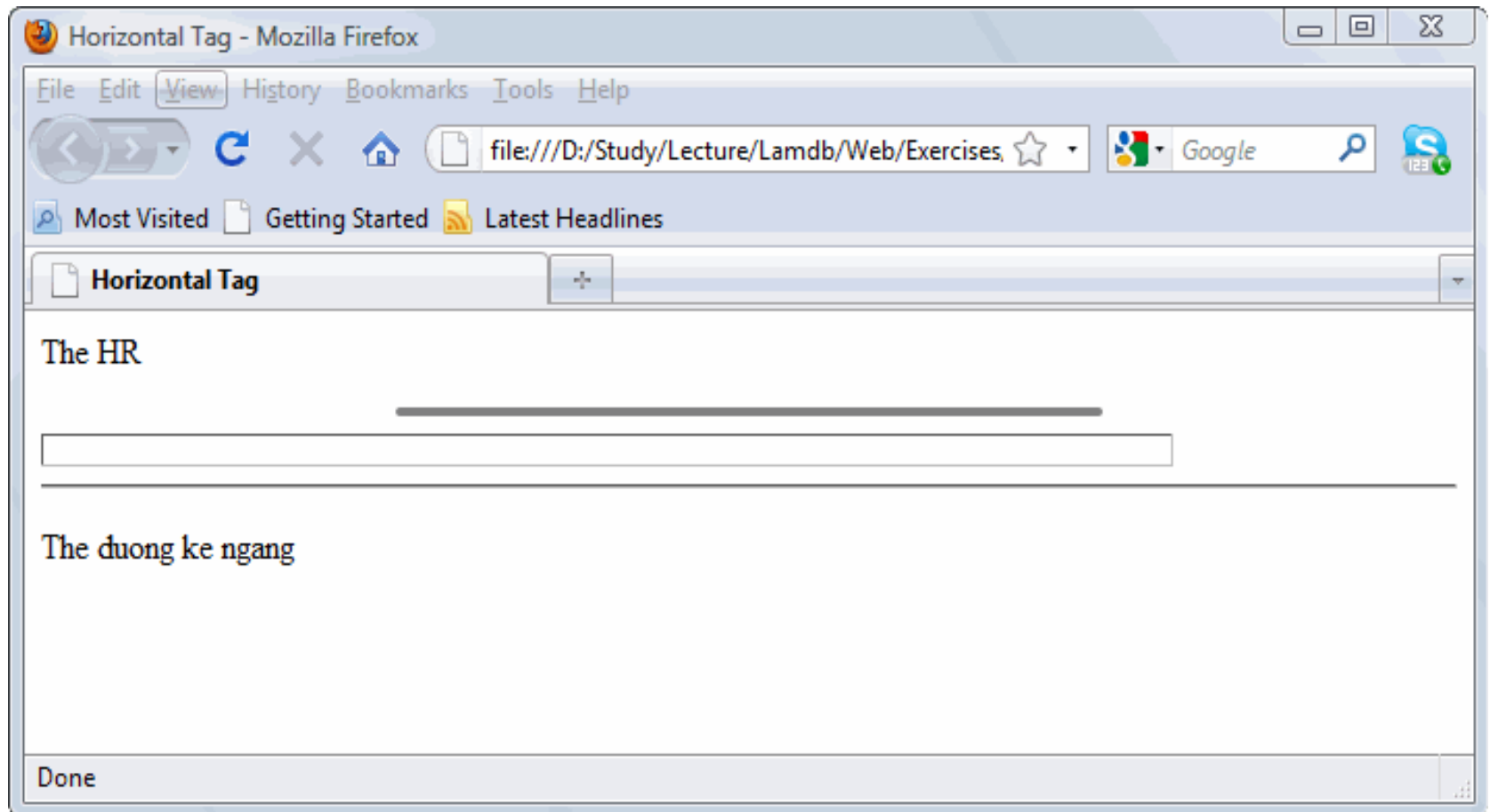
- Được sử dụng để nhóm các **inline** trong văn bản HTML.
- Cú pháp:
` Nội dung văn bản `
- Thuộc tính:
 - ✓ **id**: Xác định tên ID cho thành phần
 - ✓ **title**: Xác định thêm thông tin cho thành phần

3.1. Các thẻ xử lý văn bản

❑ Ví dụ:

```
<HTML>
  <HEAD>
    <TITLE>Horizontal Tag</TITLE>
  </HEAD>
  <BODY>
    <P>The HR</P>
    <HR noshade size="5" align="center" width="50%" />
    <HR size="15" align = "left" width="80%" />
    <HR>
    <P>The duong ke ngang</P>
  </BODY>
</HTML>
```

3.1. Các thẻ xử lý văn bản



3.1. Các thẻ xử lý văn bản

❑ Thẻ ...

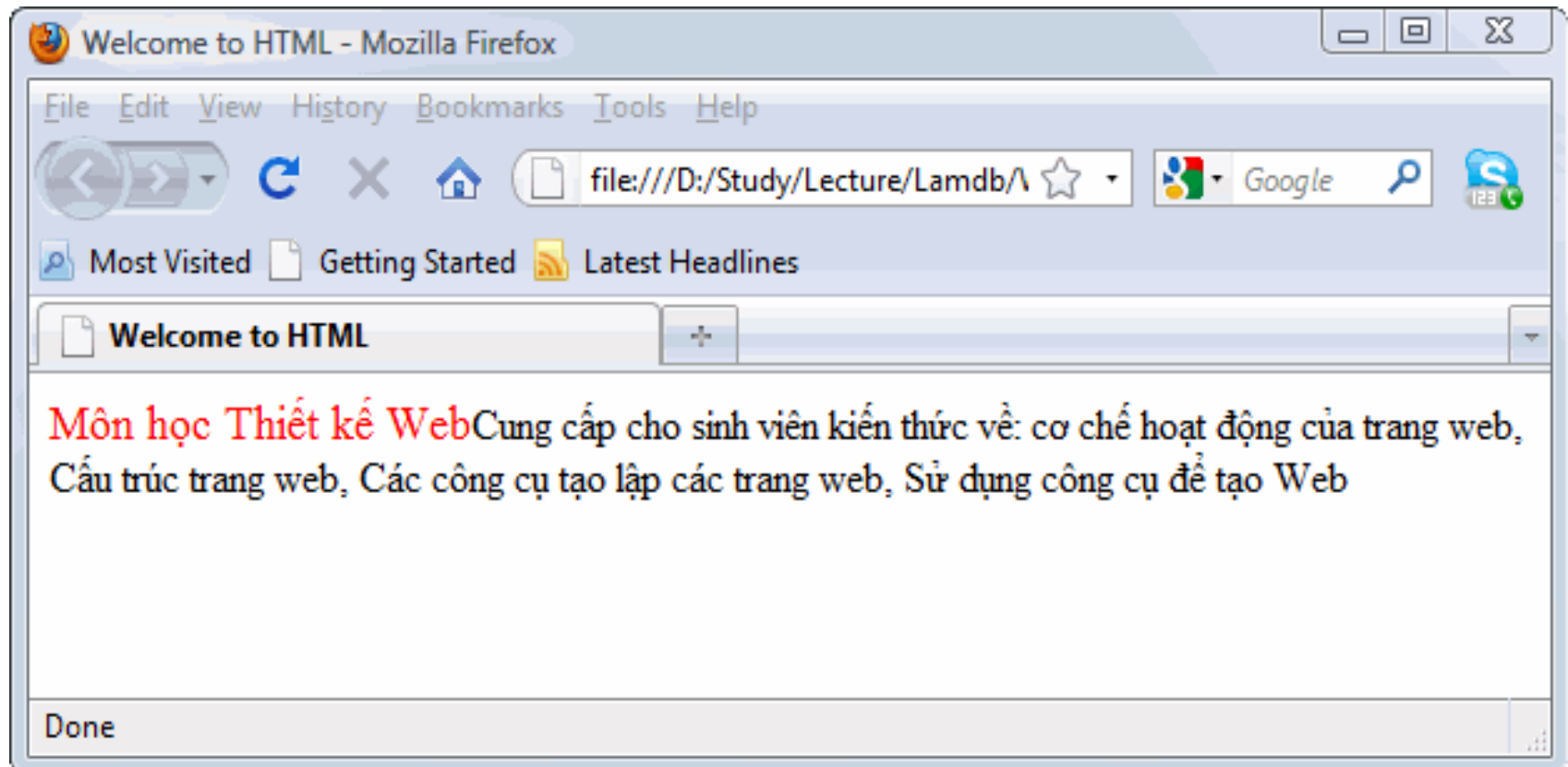
- Định dạng một chuỗi với kiểu chữ, cỡ chữ, màu chữ
- Thuộc tính:
 - ✓ **face**: kiểu chữ như Arial, Tahoma...
 - ✓ **color**: màu chữ được xác định từ 3 màu cơ bản R, G, B. Giá trị biểu diễn: #RRGGBB từ 00-FF
 - ✓ **size**: kích thước như 1, 2, 3...7

3.1.2. Các thẻ định dạng kí tự

□ Ví dụ:

```
<HTML>
  <HEAD>
    <TITLE>Welcome to HTML</TITLE>
  </HEAD>
  <BODY>
    <P align = "justify " ><FONT face = "Times New Roman" size="4"
    color="red">Môn học Thiết kế Web </FONT>Cung cấp cho sinh viên kiến
    thức về: cơ chế hoạt động của trang web, Cấu trúc trang web, Các công cụ
    tạo lập các trang web, Sử dụng công cụ để tạo Web
  </P>
</BODY>
</HTML>
```

3.1.2. Các thẻ định dạng kí tự



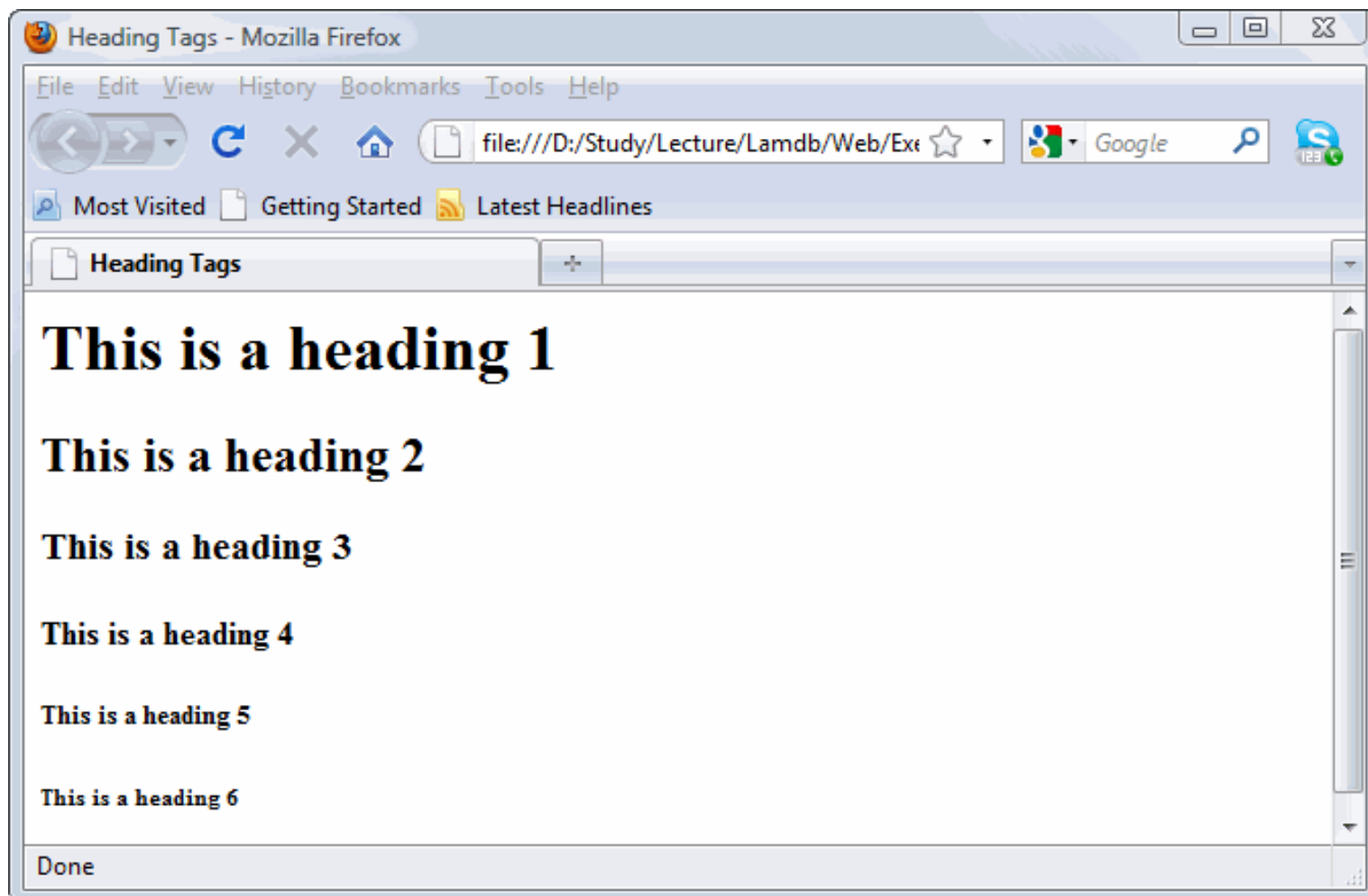
3.1. Các thẻ xử lý văn bản

❑ Tiêu đề - Heading <h>

- Được sử dụng với các thẻ từ <H1> đến <H6>
- Thuộc tính: align

```
<HTML>
  <HEAD>
    <TITLE>Heading Tags</TITLE>
  </HEAD>
  <BODY>
    <H1>This is a heading 1</H1>
    <H2>This is a heading 2</H2>
    <H3>This is a heading 3</H3>
    <H4>This is a heading 4</H4>
    <H5>This is a heading 5</H5>
    <H6>This is a heading 6</H6>
  </BODY>
</HTML>
```

3.1. Các thẻ xử lý văn bản



3.1. Các thẻ xử lý văn bản

□ Định dạng kiểu chữ

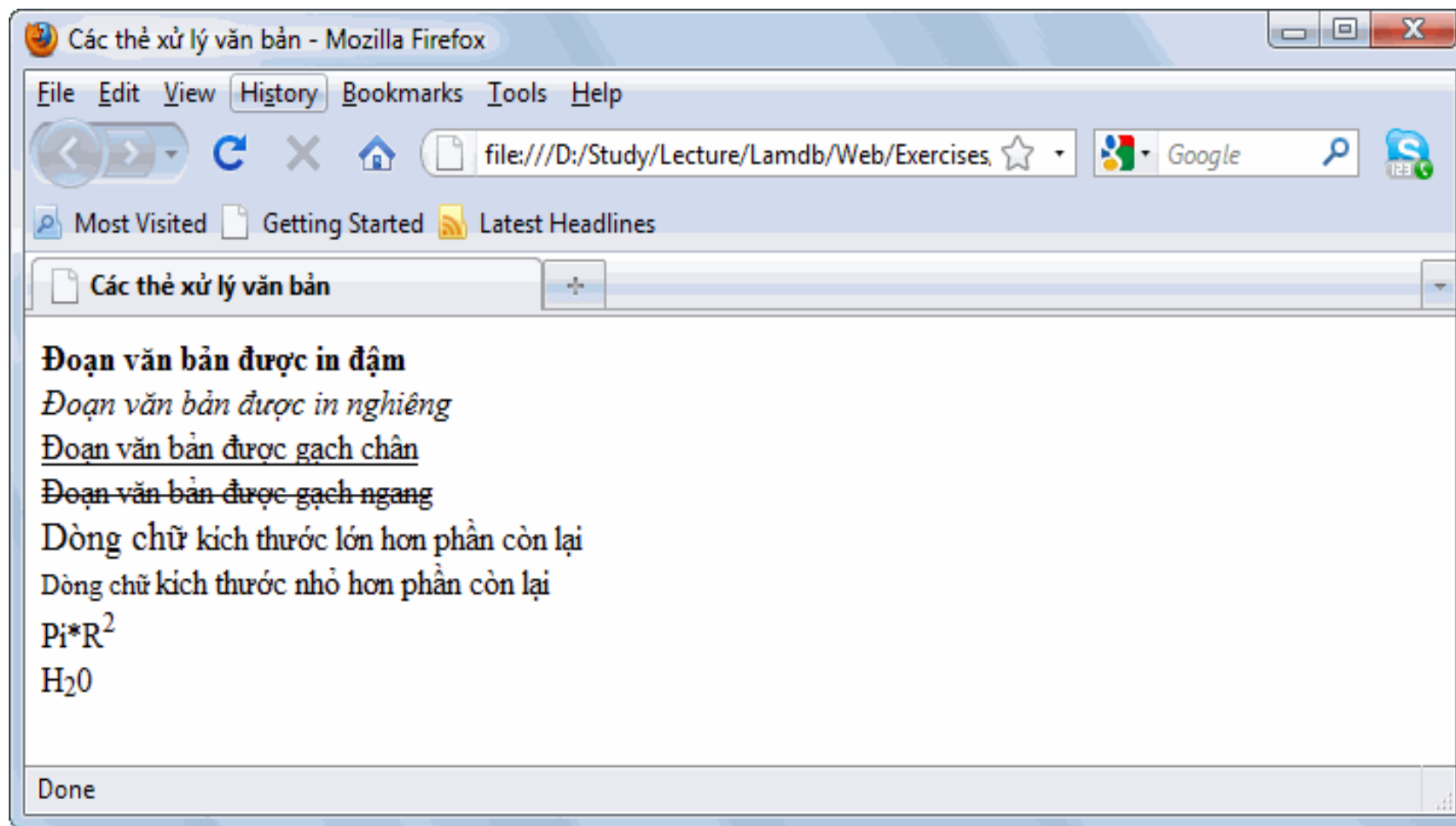
- ****, **<U>**, **<I>**
- **<S>**: dòng chữ bị gạch ngang
- **<BIG>**: hiển thị với kích thước font lớn hơn phần còn lại của văn bản
- **<SMALL>**: hiển thị với kích thước font nhỏ hơn phần còn lại của văn bản.
- **<SUP>**, **<SUB>**: chỉ số trên/dưới
- **...** : Chữ được nhấn mạnh
- **...** : **Chữ đậm**

3.1. Các thẻ xử lý văn bản

□ Ví dụ:

```
<HTML>
  <HEAD>
    <TITLE>Các thẻ xử lý văn bản</TITLE>
  </HEAD>
  <BODY>
    <B>Đoạn văn bản được in đậm</B><BR>
    <I>Đoạn văn bản được in nghiêng</I><BR>
    <U>Đoạn văn bản được gạch chân</U><BR>
    <S>Đoạn văn bản được gạch ngang</S><BR>
    <BIG>Dòng chữ </BIG>kích thước lớn hơn phần còn lại<BR>
    <SMALL>Dòng chữ </SMALL>kích thước nhỏ hơn phần còn
    lại<BR>
    Pi*R<SUP>2</SUP><BR>
    H<SUB>2</SUB>0
  </BODY>
</HTML>
```

3.1. Các thẻ xử lý văn bản



3.1. Các thẻ xử lý văn bản

Ký tự	Mô tả	Tên	Mã
“ ”	Khoảng trống (non-breaking space)	 	
©	Bản quyền	©	©
®	Thương hiệu đã được đăng ký	®	®
™	Thương hiệu	™	™
€	Đồng Euro	€	€
£	Bảng Anh	&puond;	£
<	Ký tự nhỏ hơn (less than)	<	<
>	Ký tự lớn hơn (greater than)	>	>
&	Ký hiệu & (ampersand)	&	&

Một số kí tự đặc biệt

3.2. Các thẻ tạo danh sách

3.2.1. Danh sách không được sắp xếp

3.2.2. Danh sách được sắp xếp

3.2.1. Danh sách không được sắp xếp

- Nằm trong cặp thẻ `...`
- Thẻ mở ``, thẻ đóng `` không bắt buộc

Cú pháp:

``

`` Mục 1 của danh sách ``

`` Mục 2 của danh sách ``

...

`` Mục n của danh sách ``

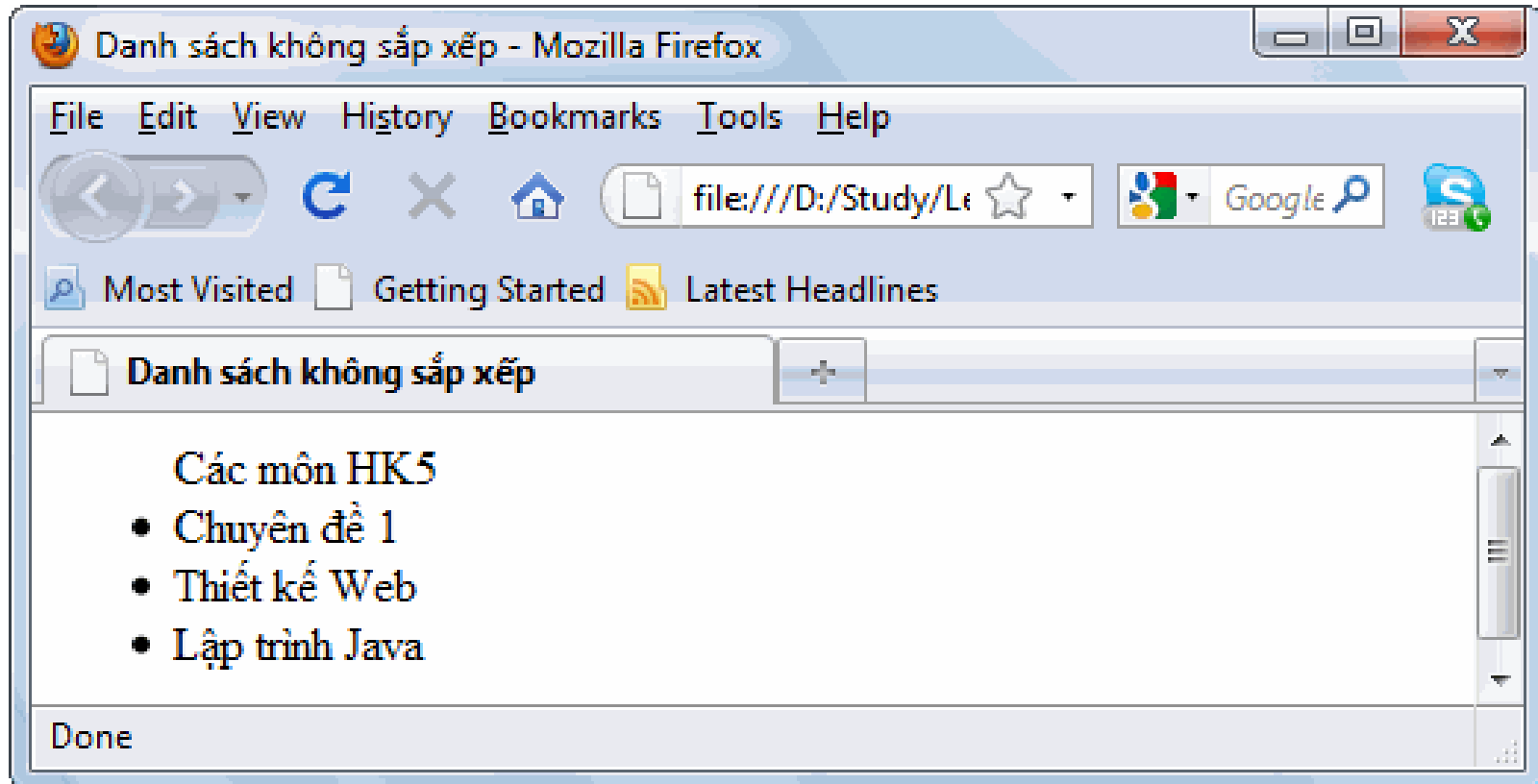
``

3.2.1. Danh sách không được sắp xếp

❑ Ví dụ:

```
<HTML>
  <BODY>
    <UL>Các môn HK5
      <LI>Chuyên đề 1
      <LI>Thiết kế Web
      <LI>Lập trình Java
    </UL>
  </BODY>
</HTML>
```

3.2.1. Danh sách không được sắp xếp



3.2.1. Danh sách không được sắp xếp

❑ Sử dụng thuộc tính **type** để định dạng bullet

<LI type = "square"> : bullet vuông

<LI type = "disc">: bullet hình cầu

<LI type = "circle">: bullet hình tròn

3.2.1. Danh sách không được sắp xếp

```
<HTML>
```

```
  <BODY>
```

```
    <UL>Các môn HK5 – V2
```

```
      <LI type = "square" >Chuyên đề 1
```

```
      <LI type = "disc" >Thiết kế Web
```

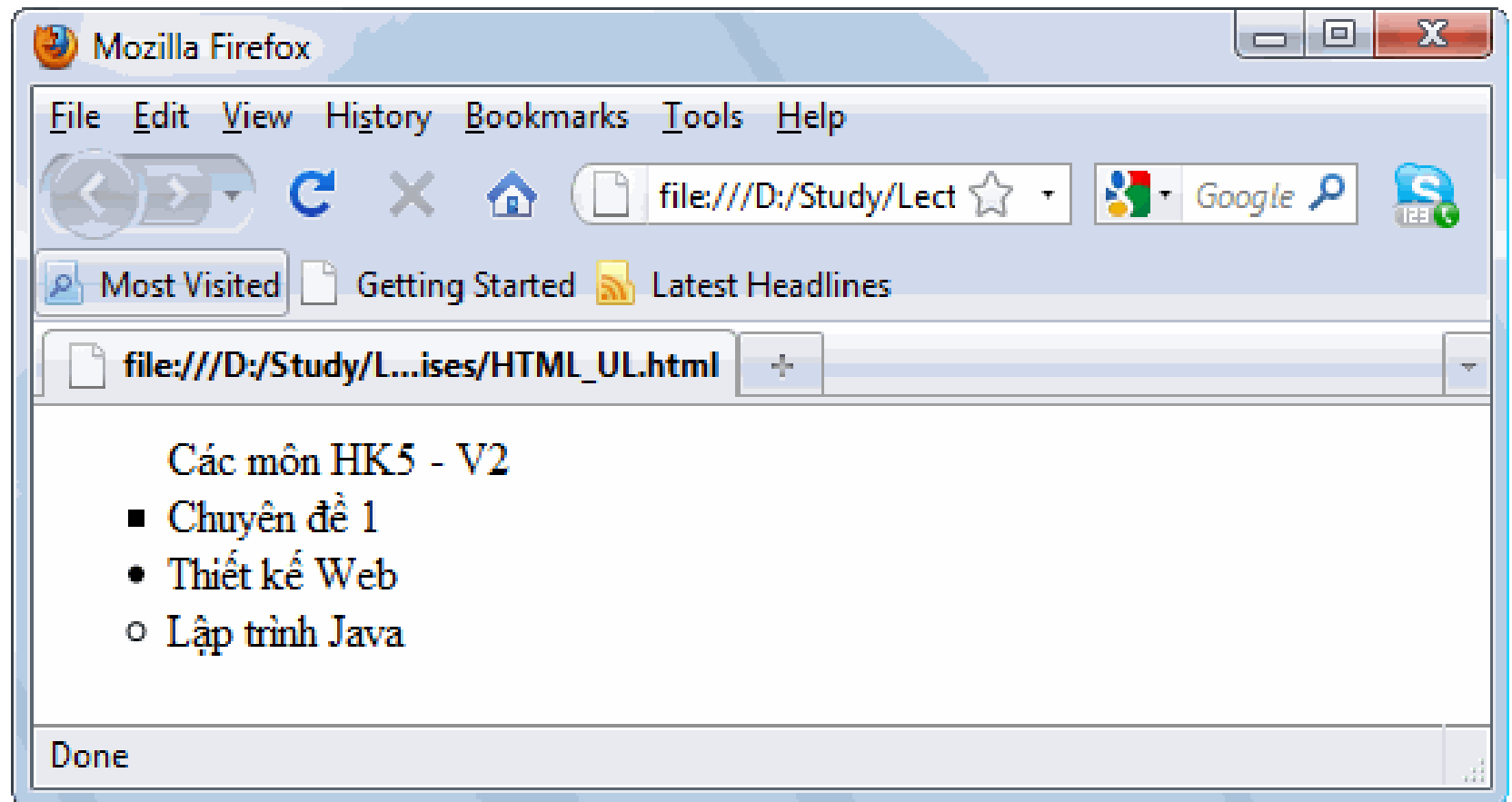
```
      <LI type = "circle">Lập trình Java
```

```
    </UL>
```

```
  </BODY>
```

```
</HTML>
```

3.2.1. Danh sách không được sắp xếp



3.2.2. Danh sách được sắp xếp

- Nằm trong thẻ `...`
- Thẻ mở ``, thẻ đóng `` không bắt buộc

Cú pháp:

``

`` Mục 1 của danh sách ``

`` Mục 2 của danh sách ``

...

`` Mục n của danh sách ``

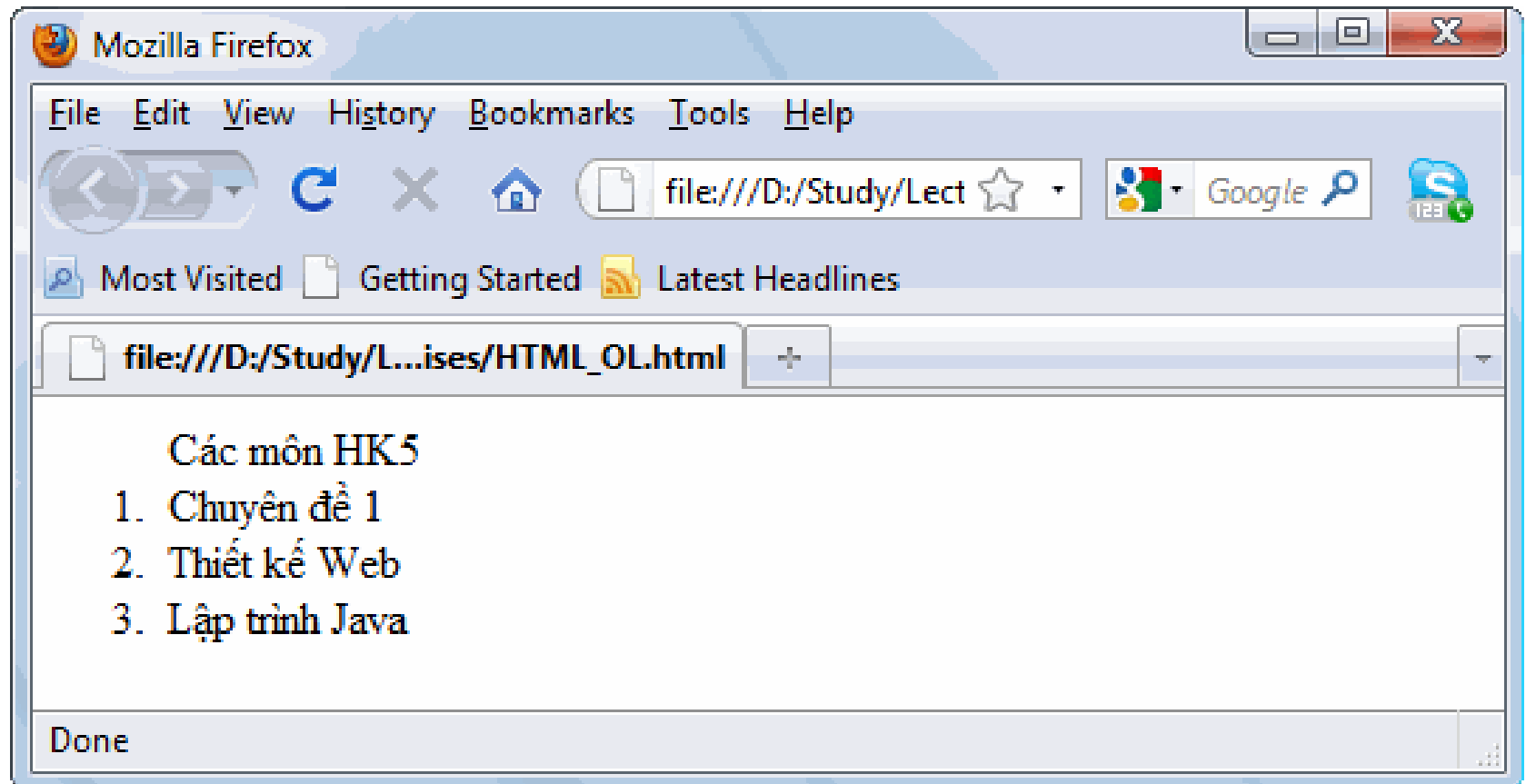
``

3.2.2. Danh sách được sắp xếp

□ Ví dụ:

```
<HTML>
  <BODY>
    <OL>Các môn HK5
      <LI>Chuyên đề 1
      <LI>Thiết kế Web
      <LI>Lập trình Java
    </OL>
  </BODY>
</HTML>
```

3.2.2. Danh sách được sắp xếp



3.2.2. Danh sách được sắp xếp

- Sử dụng thuộc tính **type** để định dạng
- Định dạng cho từng mục
 - `<LI type = "I">` kí tự LaMã: I, II,...
 - `<LI type = "i">` kí tự thường: i, ii,...
 - `<LI type = "A">` kí tự chữ hoa: A, B,...
 - `<LI type = "a">` kí tự chữ thường: a, b,...
 - `<OL start = "n">` bắt đầu với số n
- Định dạng cho toàn bộ danh sách
 - `<OL type = "Giá trị">...`

3.3. Các thẻ tạo bảng

- ❑ HTML coi một bảng gồm *nhiều dòng*, một dòng gồm *nhiều ô*, và chỉ có ô mới chứa dữ liệu của bảng.

Các thẻ:

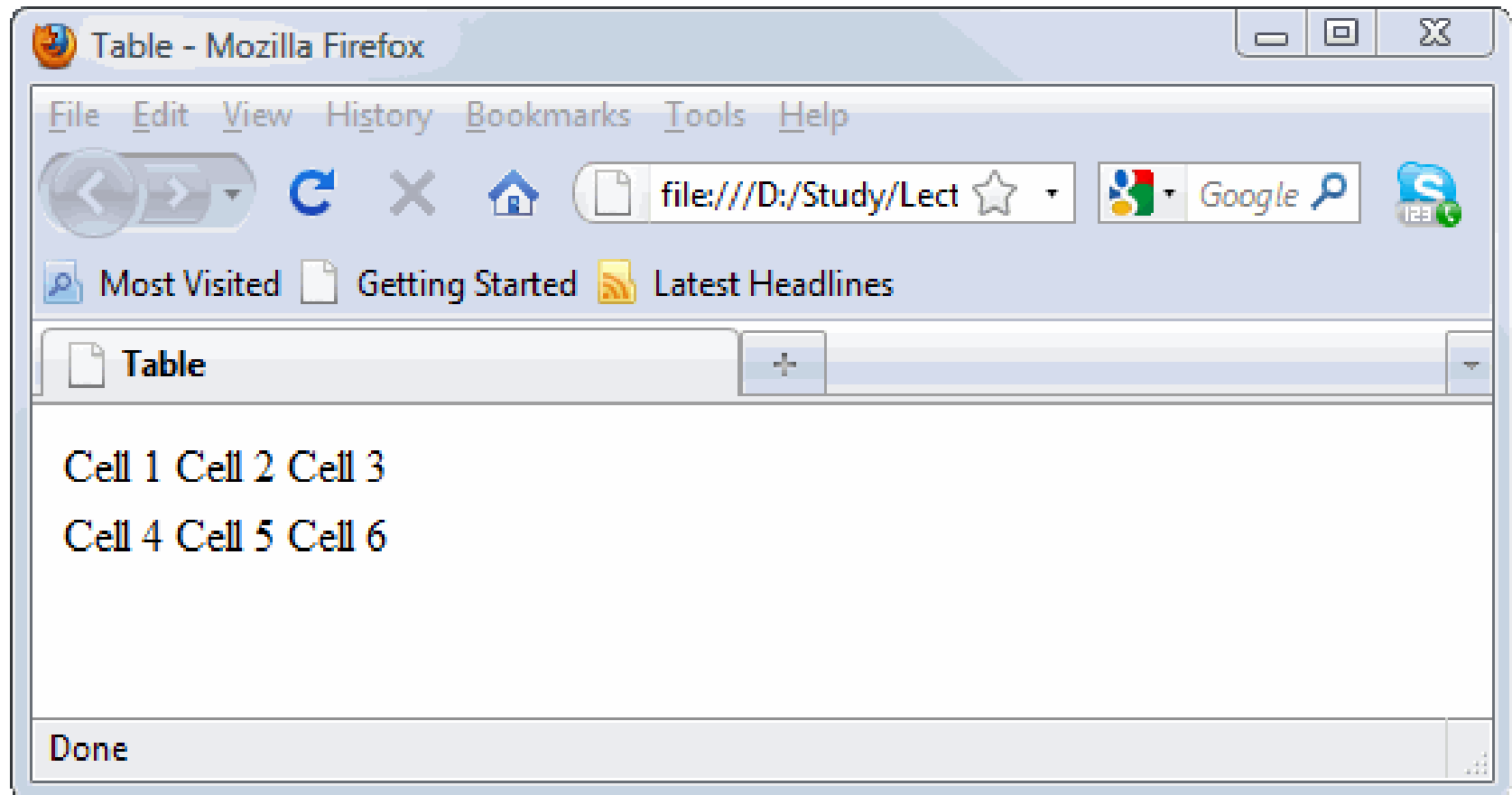
- Tạo bảng: `<TABLE>...</TABLE>`
- Tạo dòng: `<TR>...</TR>`
- Tạo ô:
 - ✓ Ô tiêu đề của bảng: `<TH>...</TH>`
 - ✓ Ô dữ liệu: `<TD>...</TD>`
- Nhóm các dòng tiêu đề bảng: `<thead>`
- Nhóm các dòng nội dung chính: `<tbody>`
- Nhóm các dòng chân bảng: `<tfoot>`

3.3. Các thẻ tạo bảng

❑ Ví dụ:

```
<TABLE>
  <TR>
    <TD>Cell 1</TD>
    <TD>Cell 2</TD>
    <TD>Cell 3</TD>
  </TR>
  <TR>
    <TD>Cell 4</TD>
    <TD>Cell 5</TD>
    <TD>Cell 6</TD>
  </TR>
</TABLE>
```

3.3. Các thẻ tạo bảng



3.3. Các thẻ tạo bảng

▪ Thuộc tính

- ✓ **Border** = "số": đường viền bảng. Mặc định 0
- ✓ **Width** = "độ rộng"
- ✓ **Height** = "độ cao"
- ✓ **Cellspacing** = "số": khoảng cách giữa 2 ô liên tiếp
- ✓ **Cellpadding** = "số": khoảng cách từ cạnh ô tới nội dung ô
- ✓ **Bgcolor** = "màu nền"
- ✓ **Background** = "địa chỉ ảnh nền"
- ✓ **Colspan** = "số": gộp các ô lại với nhau
- ✓ **Rowspan** = "số": gộp các hàng lại với nhau
- ✓ **Align**: căn theo chiều ngang (left, center, right)
- ✓ **Valign**: căn theo chiều dọc (top, middle, bottom)

3.3. Các thẻ tạo bảng

❑ Ví dụ:

```
<TABLE width = "400" height = "200" border = "1" bgcolor = "#CCCCCC"  
cellspacing = "10" cellpadding="40">
```

```
<TR>
```

```
<TD>Cell 1</TD>
```

```
<TD>Cell 2</TD>
```

```
<TD>Cell 3</TD>
```

```
</TR>
```

```
<TR>
```

```
<TD>Cell 4</TD>
```

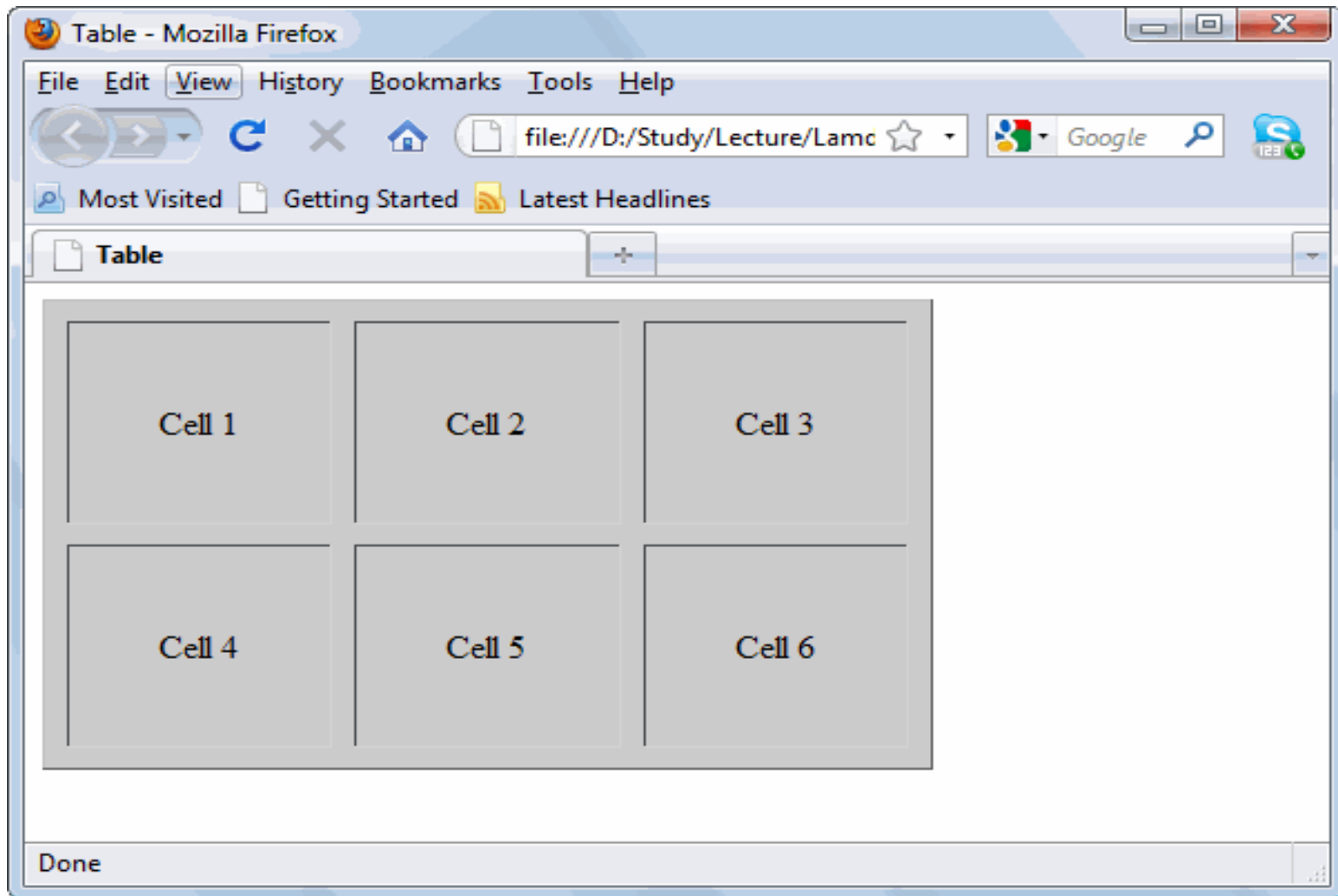
```
<TD>Cell 5</TD>
```

```
<TD>Cell 6</TD>
```

```
</TR>
```

```
</TABLE>
```

3.3. Các thẻ tạo bảng



3.4. Thẻ liên kết

- ❑ Siêu liên kết (hyperlink) cho phép bạn chuyển từ trang này sang trang khác, tải tập tin về máy...
- ❑ Tạo liên kết đến
 - Một phần khác của cùng tài liệu
 - Một tài liệu khác
 - Một phần của tài liệu khác

3.4. Thẻ liên kết

- ❑ Tạo siêu liên kết
 - Điểm cung cấp liên kết (hypermedia)
 - Địa chỉ của tài liệu được kết nối
- ❑ Địa chỉ
 - Sử dụng địa chỉ IP hoặc URL
 - Có 2 dạng URL
 - ✓ URL tuyệt đối: địa chỉ đầy đủ đến file
 - ✓ URL tương đối: URL thiếu một hay nhiều phần

3.4. Thẻ liên kết

❑ Cú pháp tạo liên kết

`Hypertext`

hoặc

` Hypertext `

Trong đó:

`protocol:` http, ftp

❑ Phân loại liên kết:

- Liên kết ngoài
- Liên kết trong
- Liên kết Email

3.4. Thẻ liên kết

❑ Liên kết ngoài

- Là liên kết đến một tài liệu khác
 - ✓ Tài liệu được liên kết nằm trong cùng một thư mục => chỉ cần xác định tên tài liệu

Ví dụ:

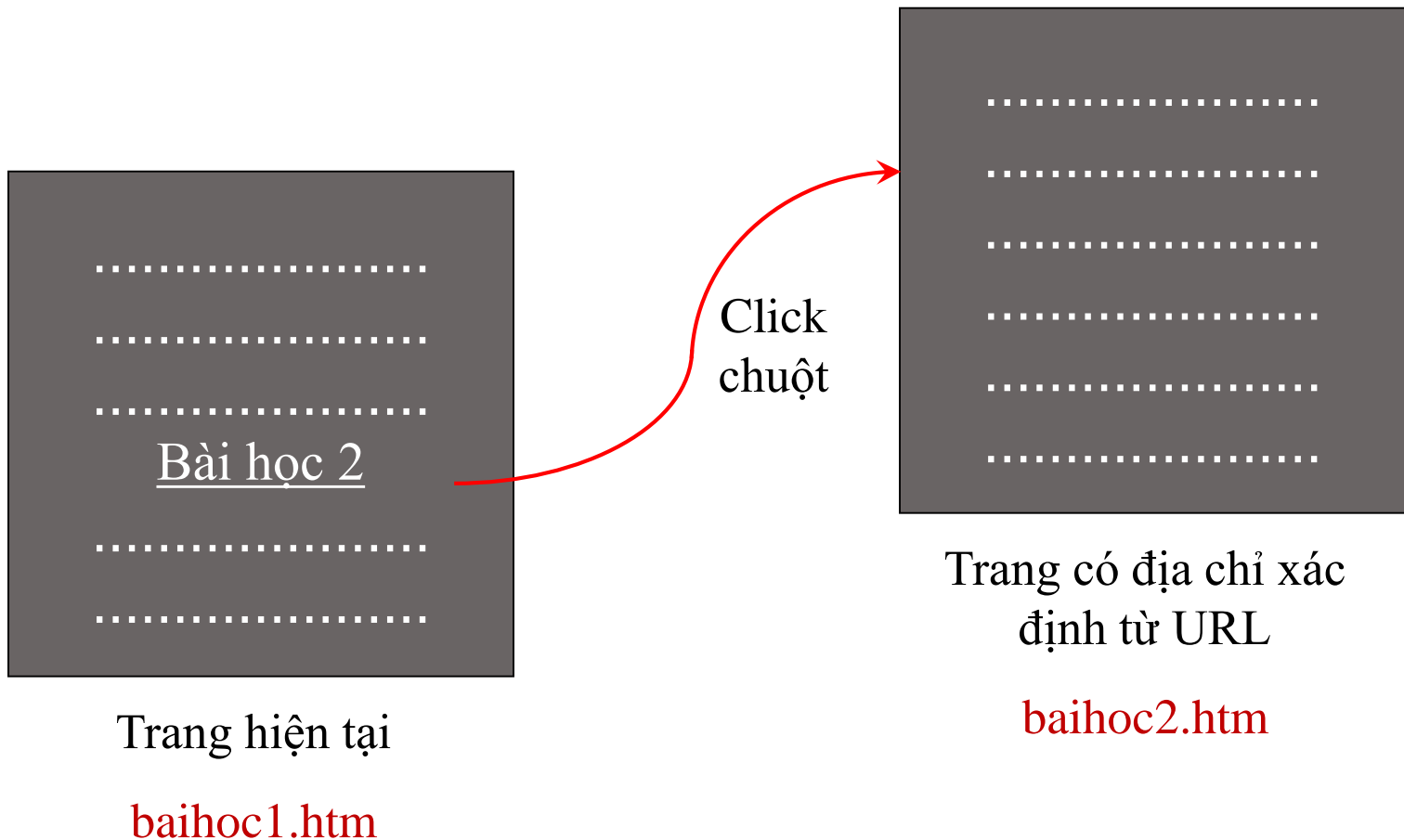
`Homepage`

- ✓ Tài liệu được liên kết thuộc thư mục khác => sử dụng đường dẫn tuyệt đối hoặc tương đối
 - Đường dẫn tuyệt đối: "C:/MyFolder/MyFile.htm"
 - Đường dẫn tương đối: "../MyFolder/MyFile.htm"

3.4. Thẻ liên kết

Cú pháp:

` Text đại diện `



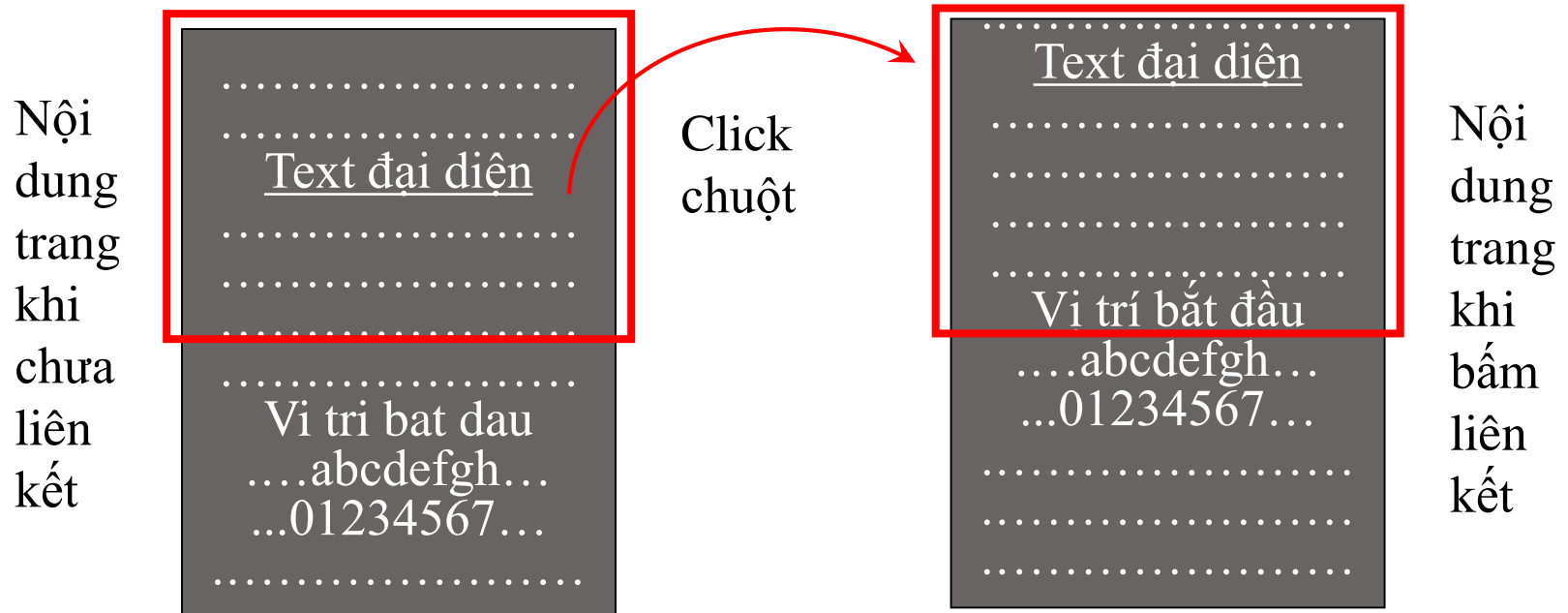
3.4. Thẻ liên kết

❑ Liên kết trong

- Cho phép người sử dụng chuyển đến các phần khác nhau của một tài liệu.

`Vị trí bắt đầu `

`Text đại diện`



3.4. Thẻ liên kết

❑ Liên kết Email

Cho phép người sử dụng gửi thư đến một địa chỉ.

Cú pháp:

`Text đại diện`

Ví dụ:

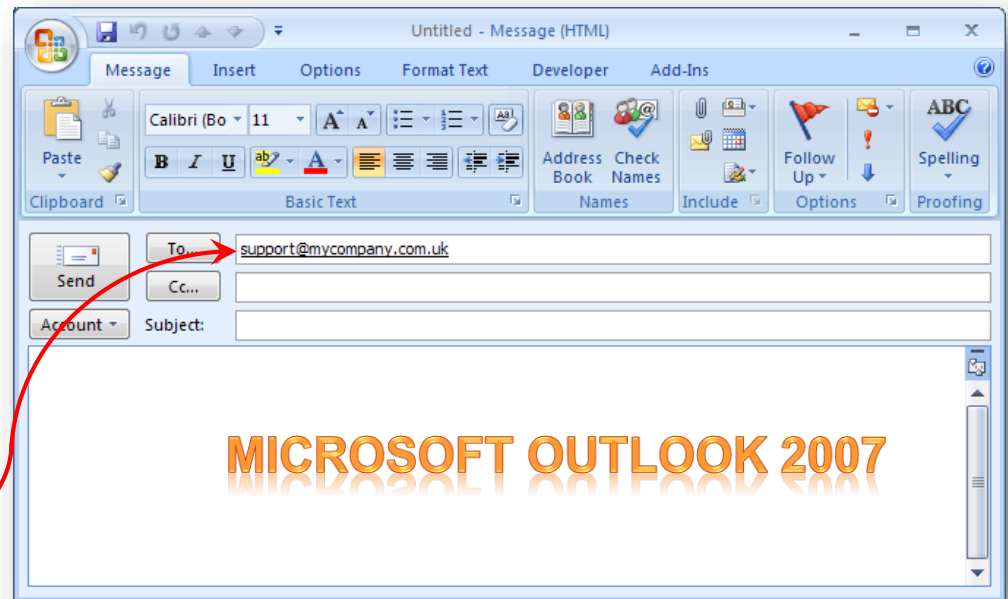
` Gửi thư `

3.4. Thẻ liên kết

`Liên hệ Admin`



Click
chuột



3.5. Các thẻ đa phương tiện

3.5.1. Hình ảnh

3.5.2. Âm thanh

3.5.1. Hình ảnh – Images

❑ Cú pháp:

<img SRC = url ALIGN = TOP/MIDDLE/BOTTOM ALT = text
BORDER = n WIDTH = width HEIGHT = height />

Thuộc tính:

- ✓ BORDER: Viền ảnh
- ✓ ALIGN: Căn lề ảnh so với văn bản xung quanh
- ✓ WIDTH: Độ rộng ảnh
- ✓ HEIGHT: Chiều cao ảnh
- ✓ ALT: Chú thích cho ảnh

3.5.1. Hình ảnh

Ví dụ:

```
<HTML>  
  <BODY>  
    <IMG align="bottom" src="car.gif">  
    <IMG align="middle" src="car.gif">  
    <IMG align="top" src="car.gif">  
  </BODY>  
</HTML>
```

3.5.2. Âm thanh

❑ Cách sử dụng thẻ object

```
<object data='GraduaMon.mp3' type='application/x-mplayer2'  
width='0' height='0'>
```

```
    <param name='filename' value='GraduaMon.mp3'>
```

```
    <param name='playcount' value='true'>
```

```
    <param name='autostart' value='true'>
```

```
</object>
```

- ❑ Chạy được trên nhiều trình duyệt IE, Fire Fox, Chrome, Safari...

4. Các thẻ tạo biểu mẫu

- Cho phép người sử dụng nhập dữ liệu trên trang web. Dữ liệu này có thể được gửi về server để xử lý.
- Người sử dụng nhập dữ liệu thông qua các điều khiển (controls).
- Có nhiều loại control.

4. Các thẻ tạo biểu mẫu

4.1. Form

4.2. Label

4.3. Textbox

4.4. Checkbox

4.5. Option Button

4.6. Button

4.7. ComboBox

4.8. ListBox

4.9. TextArea

4.1. Tag <form>

❑ Là container **chứa các thành phần nhập liệu khác.**

Cú pháp:

```
<form action = "url" method = "get | post" name = "name"  
      target = "frame_name | _blank | _self ">  
</form>
```

Thuộc tính:

action: chỉ định trang web nhận xử lý dữ liệu từ FORM này khi có sự kiện click của button SUBMIT

method: Phương thức gửi dữ liệu (POST, GET)

name: Tên form.

target: Chỉ định cửa sổ sẽ hiển thị kết quả sau khi gửi dữ liệu từ form đến server.

4.2. Label

Dùng tạo nhãn cho các điều khiển khác.

Cú pháp:

`<label for = "for">Nhãn</label>`

Thuộc tính:

`for`: có giá trị trùng với id thẻ `<input>`

4.3. Textbox

❑ Sử dụng để nhập các văn bản ngắn (trên 1 dòng) hoặc mật khẩu

Cú pháp:

```
<input name = "name" type = "text|password" value = "value">
```

Thuộc tính:

name: Tên hộp văn bản

type = "text": Ô nhập văn bản thường

type = "password": Ô nhập mật khẩu

value: giá trị mặc định

4.4. Checkbox

❑ Cho phép **chọn nhiều** lựa chọn trong một nhóm lựa chọn được đưa ra bằng cách đánh dấu ("tích").

Cú pháp:

```
<input name = "name" type = "checkbox" value = "value" checked>
```

Thuộc tính:

name: Tên hộp kiểm checkbox

type = "checkbox"

value: giá trị chương trình sẽ nhận được nếu người sử dụng chọn ô này.

checked: nếu có thì nút này mặc định được chọn

4.5. Radio Button

☐ Cho phép **chọn một** lựa chọn trong một nhóm lựa chọn được đưa ra.

Cú pháp:

```
<input name = "name" type = "radio" value = "value" checked>
```

Thuộc tính:

name: Tên hộp kiểm checkbox

type = "radio"

value: giá trị chương trình sẽ nhận được nếu người sử dụng chọn ô này.

checked: nếu có thì nút này mặc định được chọn

4.6. Button

- ❑ Sử dụng để người sử dụng ra lệnh thực hiện công việc.

Trên web có 3 loại nút:

submit: Tự động ra lệnh gửi dữ liệu

reset: đưa mọi dữ liệu về trạng thái mặc định

normal: người lập trình tự xử lý

Cú pháp:

```
<input name = "name" type = "submit|reset|button" value = "value">
```

Thuộc tính:

name: Tên nút

type = "submit": nút submit

type = "reset": nút reset

type = "button": nút thông thường

value = "tiêu đề nút"

4.7. ComboBox

□ Bao gồm một danh sách có nhiều phần tử. Tại một thời điểm chỉ có 1 phần tử được chọn

Cú pháp:

```
<select name = "name">
```

```
    <option value = "value1">Tên mục chọn thứ nhất</option>
```

```
    <option value = "value2">Tên mục chọn thứ hai</option>
```

```
    ...
```

```
</select>
```

Thuộc tính:

name: Tên combobox

value: Giá trị chương trình nhận được nếu phần tử được chọn

selected: nếu có thì phần tử này mặc định được chọn

4.8. Listbox

□ Bao gồm một danh sách có nhiều phần tử. Có thể nhìn thấy nhiều phần tử cùng lúc, có thể lựa chọn nhiều phần tử.

Cú pháp:

```
<select name = "name" size = "size" multiple>  
    <option value = "value1">Tên mục chọn thứ nhất</option>  
    <option value = "value2">Tên mục chọn thứ hai</option>  
    ...  
</select>
```

Thuộc tính:

name: Tên combobox

size: Số dòng nhìn thấy

multiple: Cho phép chọn nhiều phần tử

value: Giá trị chương trình nhận được nếu phần tử được chọn

4.9. TextArea

- ❑ Cho phép nhập văn bản dài trên nhiều dòng.

Cú pháp:

```
<textarea name = "name" rows = "rows" cols = "cols">
```

Nội dung mặc định

```
</textarea>
```

Thuộc tính:

name: Tên TextArea

rows: Số dòng hiển thị

cols: Số cột hiển thị

4.10. FieldSet

- ❑ Fieldset dùng để nhóm các thành phần form có liên quan

Cú pháp:

`<fieldset>`

`</fieldset>`

Thuộc tính:

4.11. Legend

❑ *Legend* nằm trong *fieldset* đóng vai trò là caption cho *Fieldset* đó

Cú pháp:

<fieldset>

<legend> </legend>

</fieldset>

Thuộc tính:

5. Phương thức GET/POST

Phương thức GET

- ❑ Các đối số của Form được ghi chêm theo vào đường dẫn URL của thuộc tính action trong tag <form>
- ❑ Khối lượng dữ liệu đối số được truyền đi của Form bị giới hạn bởi chiều dài tối đa của một URL trên Address bar.
 - IE : Tối đa của một URL là 2.048 ký tự
 - Firefox : Tối thiểu của một URL là khoảng 100.000 ký tự
 - Safari : Tối thiểu của một URL là 80.000 ký tự
 - Opera : Tối thiểu của một URL là 190.000 ký tự
 - Apache Server : Tối đa của một URL là 8.192 ký tự
 - IIS Server : Tối đa của một URL là 16.384 ký tự

Phương thức POST

- Các đối số của Form được truyền “ngầm” bên dưới
- Khối lượng dữ liệu đồ sộ được truyền đi của Form **không** phụ thuộc vào URL -> Không bị giới hạn
- Chỉ sử dụng được phương thức truyền POST khi action chỉ định đến trang web thuộc dạng trang web **có mã lệnh xử lý trên Server.**