TRUNG TÂM ĐÀO TẠO MẠNG MÁY TÍNH NHẤT NGHỆ
ĐỐI TÁC ĐÀO TẠO CỦA MICROSOFT TẠI VIỆT NAM
105 Bà Huyện Thanh Quan, Q3, TP. HCM
Tel: 3.9322.735-0913.735.906 Fax: 3.9322.734 Web: nhatnghe.com

Microsoft GOLD CERTIFIED Partner

----------📖----------

HTML, CSS & JQUERY

NGÔN NGỮ WEB

THÀNH PHỐ HỒ CHÍ MINH – 2011

1 HTML

1.1 Giới thiệu

"Internet","World Wide Web", và "Web page" không chỉ còn là các thuật ngữ. Giờ đây, các thuật ngữ này đã trở thành hiện thực. Internet là mạng máy tính lớn nhất trên thế giới, được xem như là mạng của các mạng. World Wide Web là một tập con của Internet. World Wide Web gồm các Web Servers có mặt khắp mọi nơi trên thế giới. Các Web server chứa thông tin mà bất kỳ người dùng nào trên thế giới cũng có thể truy cập được. Các thông tin đó được lưu trữ dưới dạng các trang Web.

Trong phần này, chúng ta sẽ học về Ngôn ngữ đánh dấu siêu văn bản (HTML), đây là một phần quan trọng trong lãnh vực thiết kế và phát triển thế giới Web.

1.1.1Giới thiệu HTML

Ngôn ngữ đánh dấu siêu văn bản chỉ rõ một trang Web được hiển thị như thế nào trong một trình duyệt. Sử dụng các thẻ và các phần tử HTML, bạn có thể:

- ✓ Điều khiển hình thức và nội dung của trang
- ✓ Xuất bản các tài liệu trực tuyến và truy xuất thông tin trực tuyến bằng cách sử dụng các liên kết được chèn vào tài liệu HTML
- ✓ Tạo các biểu mẫu trực tuyến để thu thập thông tin về người dùng, quản lý các giao dịch .....
- ✓ Chèn các đối tượng như audio clip, video clip, các thành phần ActiveX và các Java Applet vào tài liệu HTML

Tài liệu HTML tạo thành mã nguồn của trang Web. Khi được xem trên trình soạn thảo, tài liệu này là một chuỗi các thẻ và các phần tử, mà chúng xác định trang web hiển thị như thế nào. Trình duyệt đọc các file có đuôi .htm hay .html và hiển thị trang web đó theo các lệnh có trong đó.

Ví dụ, theo cú pháp HTML dưới đây sẽ hiển thị thông điệp "`Trang web đầu tiên của tôi`"

Ví dụ 1:

```
<HTML>
<HEAD>
    <TITLE>Chao ban</TITLE>
    <META http-equiv="Content-Type" content="text/html; charset=utf-8">
</HEAD>
<BODY>
      <H3>Trang web đầu tiên của tôi</H3>
</BODY>
</HTML>
```

Trình duyệt thông dịch những lệnh này và hiển thị trang web như hình sau

Hình 4.1: Kết quả ví dụ 1

1.1.2Cấu trúc của một tài liệu HTML

Một tài liệu HTML gồm 3 phần cơ bản:

- ✓ Phần <HTML>: Mọi tài liệu HTML phải bắt đầu bằng thẻ mở HTML <HTML> và kết thúc bằng thẻ đóng HTML </HTML>

  <HTML> …. </HTML>

Thẻ HTML báo cho trình duyệt biết nội dung giữa hai thẻ này là một tài liệu HTML

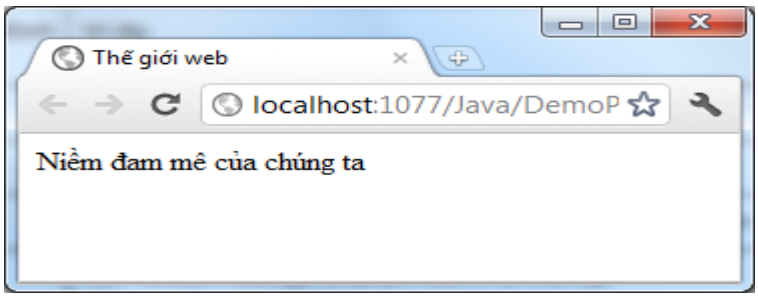
- ✓ Phần tiêu đề <HEAD>: Phần tiêu đề bắt đầu bằng thẻ <HEAD> và kết thúc bởi thẻ </HEAD>. Phần này chứa tiêu đề mà được hiển thị trên thanh điều hướng của trang Web. Tiêu đề nằm trong thẻ TITLE, bắt đầu bằng thẻ <TITLE> và kết thúc là thẻ </TITLE>.

Tiêu đề là phần khá quan trọng. Các mốc được dùng để đánh dấu một web site. Trình duyệt sử dụng tiêu đề để lưu trữ các mốc này. Do đó, khi người dùng tìm kiếm thông tin, tiêu đề của trang Web cung cấp từ khóa chính yếu cho việc tìm kiếm.

- ✓ Phần thân <BODY>: phần này nằm sau phần tiêu đề. Phần thân bao gồm văn bản, hình ảnh và các liên kết mà bạn muốn hiển thị trên trang web của mình. Phần thân bắt đầu bằng thẻ <BODY> và kết thúc bằng thẻ </BODY>

  Ví dụ 2:

```
<HTML>
    <HEAD>
        <TITLE>Thế giới web</TITLE>
    </HEAD>
    <BODY>
        <P>Niềm đam mê của chúng ta</P>
    </BODY>
</HTML>
```



Hình 4.2: Kết quả của ví dụ 2

1.2 Siêu liên kết

Siêu liên kết là kỹ thuật cho phép tạo ra sự liên kết từ trang web này đến trang web kác. Trong HTML người ta sử dụng thẻ <A> để tạo siêu liên kết. Thẻ <A> được sử dụng với các mục đích sau:

- ✓ Tạo liên kết đến trang
- ✓ Đánh dấu và tạo liên kết đến một vùng đã được đánh dấu trên một trang
- ✓ Chạy chương trình gửi email
- ✓ Gọi hàm javascript

1.2.1Liên kết trang

Thẻ <A> được sử dụng để xác định văn bản hay ảnh nào sẽ dùng làm siêu liên kết trong tài liệu HTML. Thuộc tính HREF (tham chiếu siêu văn bản) được dùng để chỉ địa chỉ hay URL của tài liệu hoặc file được liên kết.

Cú pháp của HREF là:

```
<A HREF = “<địa chỉ tài nguyên>”>Hypertext</A>
```

<địa chỉ tài nguyên> là địa chỉ của trang web bạn muốn đến. Có 2 loại địa chỉ là tương đối và tuyệt đối.

✪ Địa chỉ tuyệt đối

Địa chỉ tuyệt đối (bắt đầu với http://) là địa chỉ sử dụng để chỉ đến một tại nguyên cụ thể trên một máy chạy với port nào đó. Thông thường được sử dụng đến tham chiếu đến các tài nguyên bên ngoài. Hãy xem một số ví dụ sau:

Liên kết đến trang c.html đặt tại thư mục b của máy www.lycato.com chạy tại port 8080 sử dụng giao thức http.

```
<A HREF=”http://www.lycato.com:8080/b/c.html”>liên kết</A>
```

Liên kết đến trang c.html đặt tại thư mục b của máy www.yahoo.com chạy tại port 80 sử dụng giao thức http.

```
<A HREF=”http://www.yahoo.com/b/c.html”>liên kết</A>
```

✪ Địa chỉ tương đối:

Địa chỉ tương đối là địa chỉ của tài nguyên trong cùng một website nên người ta thông thường bỏ đi phần thông tin máy chủ và cổng chỉ còn lại phần đường dẫn đến tài nguyên tính từ gốc của website hoặc tính từ trang web hiện tại. Một vài ví dụ sau:

Trang c.html đặt tại thư mục “a/b” được tính từ gốc của web

```
<A HREF=”/a/b/c.html”>liên kết</A>
```

Trang c.html đặt tại thư mục “b” được tính từ vị trí của trang hiện tại

```
<A HREF=”b/c.html”>liên kết</A>
```

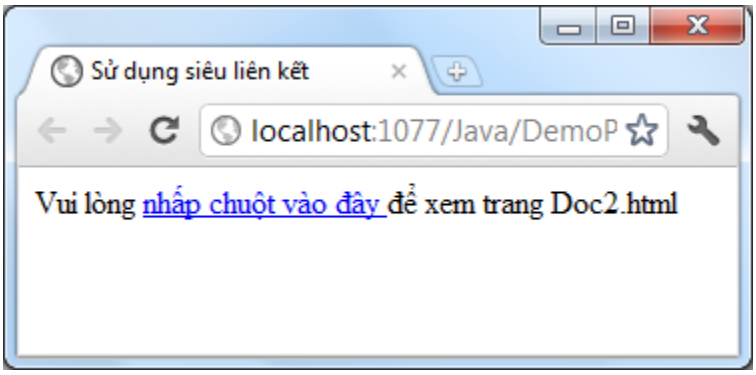
Trang c.html đặt tại thư mục “b” đồng cấp với thư mục chứa trang web hiện tại

```
<A HREF=”../b/c.html”>liên kết</A>
```

Giả sử có hai tài liệu HTML trên đĩa cứng cục bộ, Doc1.htm và Doc2.htm. Để tạo một liên kết từ Doc1.html đến Doc2.htm

Ví dụ 6:

```
<HTML>
    <HEAD>
        <TITLE>Sử dụng siêu liên kết</TITLE>
    </HEAD>
    <BODY>
        <P> Vui lòng <A HREF="Doc2.html">nhấp chuột vào đây </A>
        để xem trang Doc2.html
    </BODY>
</HTML>
```

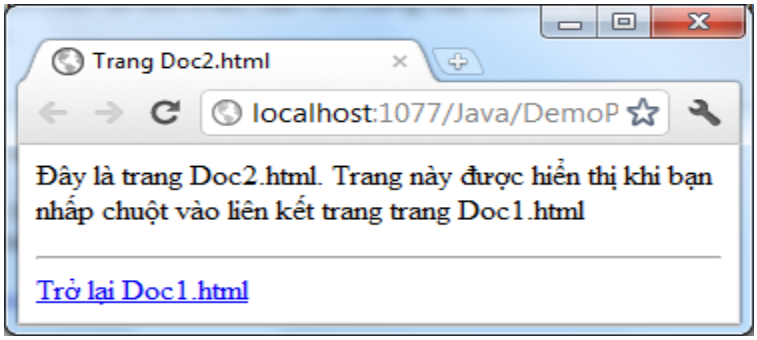


Hình 4.3: Kết quả của ví dụ 6

Khi người dùng "nhảy" đến một tài liệu khác, bạn nên cung cấp cách để quay trở lại trang chủ hoặc định hướng đến một file khác.

Ví dụ 7:

```
<HTML>
    <HEAD><TITLE>Trang Doc2.html</TITLE></HEAD>
    <BODY>
        <P> Đây là trang Doc2.html. Trang này được hiển thị khi
        bạn nhấp chuột vào liên kết trang trang Doc1.html
        <HR>
        <A HREF= Doc1.html>Trở lại Doc1.html</A>
    </BODY>
</HTML>
```



Hình 4.4: Kết quả ví dụ 7

Chú ý là các liên kết được gạch chân. Trình duyệt tự động gạch chân các liên kết. Nó cũng thay đổi hình dáng con trỏ chuột khi người sử dụng di chuyển chuột vào liên kết.

Ở ví dụ trên, các file nằm trong cùng một thư mục, vì vậy chỉ cần chỉ ra tên file trong thông số HREF là đủ. Tuy nhiên, để liên kết đến các file ở thư mục khác, cần phải cung cấp đường dẫn tuyệt đối hay đường dẫn tương đối

✪ Chú ý:

- Để tạo liên kết cho phép download các tài nguyên khác như pdf, word, excel… bạn chỉ việc tạo liên kết như bình thường và chỉ đến địa chỉ của tại nguyên muốn download. Ví dụ:

```
<a href="TaiLieu.pdf">Download</a>
```

- Thẻ <a> còn có thuộc tính quan trọng khác là target. Thuộc tính này qui định cửa sổ hiển thị trang đích. Có 4 giá trị hợp lệ là: _self, _blank, _parent và _top.
  - _self: chỉ ra trang đích hiển thị lên trang hiện tại
  - _blank: chỉ ra trang đích hiển thị trên cửa sổ mới
  - _parent, _top: qui định trang đích hiển thị trên cửa sổ cha và cửa sổ ở mức cao nhất trong trường hợp sử dụng frame.

  Ví dụ: hiển thị trang mail yahoo trên cửa sổ mới

```
<a href="http://mail.yahoo.com" target="_blank">Yahoo Mail</a>
```

1.2.2Liên kết vùng

Vài trường hợp tài liệu web rất dài, bạn muốn phân chia thành nhiều phần và cung cấp cho người đọc các liên kết để nhảy đến các vùng đó. Muốn làm được điều này bạn cần 2 việc là đánh dấu vùng và liên kết vùng.

Đánh dấu vùng: bạn đánh dấu vùng với thẻ <A> có cú pháp sau còn gọi là thả neo

```
<A NAME = “<tên vùng>”>Hiển thị tại vùng</A>.
```

Bạn sử dụng thuộc tính name để chỉ ra tên vùng. Có hoặc không có phần giữa thẻ <A> điều được.

Ví dụ: `<a name="Internet">Internet</a>`

Liến kết đến vùng đã đánh dấu: bạn sử dụng cú pháp sau để liên kết đến vùng đã đánh dấu:

```
<A HREF = “<địa chỉ trang>#<tên vùng>”>Hypertext</A>
```

Địa chỉ trang và tên vùng phân cách bởi dấu #. Dấu # dùng để chỉ ra tên vùng cần liên kết đến. Nếu bạn liên kết đến các vùng cùng một trang thì không cần chỉ ra <địa chỉ trang>. Nghĩa là:

```
<A HREF = “#<tên vùng>”>Hypertext</A>
```

Ví dụ: `<A HREF="DemoPage.htm#HTML">Giới thiệu Internet</A>`

Bây giờ mời bạn hãy xem một ví dụ liên kết vùng dẫn đến các đoạn khác nhau trong cùng một trang.

```
<HTML>
    <HEAD><TITLE>Liên kết vùng</TITLE></HEAD>
    <BODY>
        <A HREF="DemoPage.htm#Internet">Internet</A><BR>
        <A HREF="DemoPage.htm#HTML">Giới thiệu Internet</A><BR>
        <A HREF="DemoPage.htm#Design">Thiết kế web</A><BR>
```

```
        <hr />
        <p>
            <a name="Internet">Internet</a>
            Internet Internet Internet Internet Internet Internet Internet
Internet Internet Internet Internet Internet Internet Internet Internet
Internet Internet Internet Internet Internet Internet
        </p>
        <p>
            <a name="HTML">Giới thiệu Internet</a>
            Giới thiệu Internet Giới thiệu Internet Giới thiệu Internet Giới
thiệu Internet Giới thiệu Internet Giới thiệu Internet Giới thiệu Internet
        </p>
        <p>
            <a name="Design">Thiết kế web</a>
            Thiết kế web Thiết kế web Thiết kế web Thiết kế web Thiết kế web
Thiết kế web Thiết kế web Thiết kế web Thiết kế web Thiết kế web Thiết kế web
        </p>
    </BODY>
</HTML>
```

Kết quả chạy trang. Khi click vào liên kết sẽ dẫn đến vùng đã liên kết

Hình 4.5: Liên kết vùng

1.2.3Sử dụng e-mail

Nếu bạn muốn liên kết đến ứng dụng mail và đặt các thông tin về email sẵn vào nó để có thể gửi mail thì bạn có thể tạo liên kết như sau.

✓ Chỉ cung cấp email

```
<a href="mailto:youremailaddress">Email Me</a>
```

✓ Cung cấp thêm tiêu đề mail

```
<a href="mailto:email@echoecho.com?subject=SweetWords">Send Email</a>
```

✓ Cung cấp cả nội dung mail

```
<a href="mailto:email@echoecho.com?subject=SweetWords&body=Please send me a copy of your new program!">Email Me</a>
```

1.2.4Gọi java script

Bạn có thể sử dụng thẻ a để gọi hàm javascript để thực hiện mục đích riêng nào đó khi người dùng click vao liên kết. Cú pháp

```
<a href="javascript:<biểu thức javascript>">Send Email</a>
```

Thông thường bạn ứng dụng vào các việc sau:

Vô hiện hóa liên kết

```
        <!--Click vào đây không làm gì cả-->
        <a href="javascript:;">Do nothing</a> |
```

Submit form thay cho nút submit

```
        <!--Submit form myForm-->
        <a href="javascript:{document.myForm.submit()}">Submit my form</a>
```

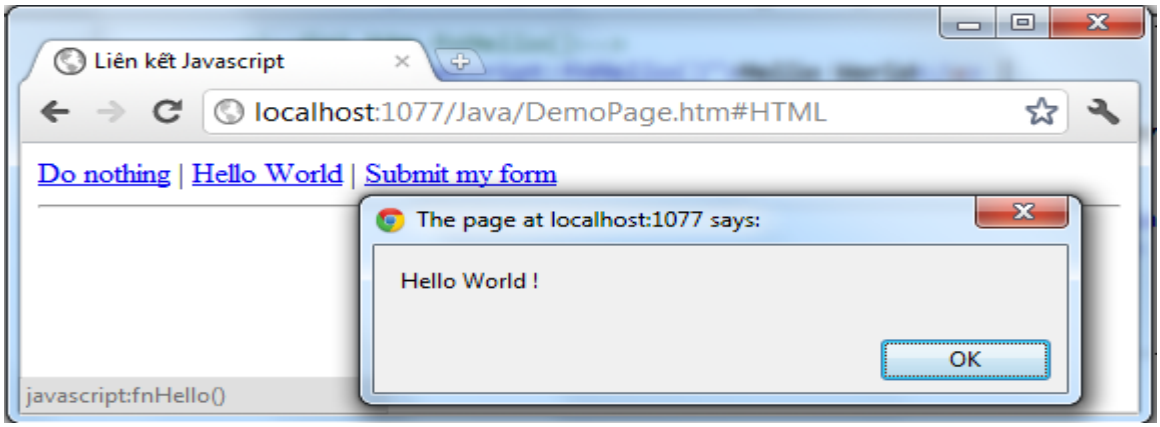
Gọi hàm để thục hiện một công việc khác (làm thay đổi giao diện, tương tác ajax)

```
        <!--Gọi hàm fnHello()-->
        <a href="javascript:fnHello()">Hello World</a> |
```

Sau đây là ví dụ hoàn chỉnh

```
<HTML>
    <HEAD>
        <TITLE>Liên kết Javascript</TITLE>
        <script>
            function fnHello() {
                alert("Hello World !");
            }
        </script>
    </HEAD>
    <BODY>
        <!--Click vào đây không làm gì cả-->
        <a href="javascript:;">Do nothing</a> |
        <!--Gọi hàm fnHello()-->
        <a href="javascript:fnHello()">Hello World</a> |
        <!--Submit form myForm-->
        <a href="javascript:{document.myForm.submit()}">Submit my form</a>
        <hr />
        <form name="myForm" action="MyPage.jsp">
            <input name="txtProductId" type="hidden" value="Nokia" />
            <input name="txtQuantity" type="hidden" value="2" />
        </form>
    </BODY>
</HTML>
```

Kết quả thực hiện. Khi click vào liên kết "Hello World"



Hình 4.6: liên kết gọi javascript

1.3 Các thẻ cơ bản

Chương này thảo luận về những thẻ cơ bản của HTML như các thẻ tiêu đề (Header), các thẻ đoạn và các thẻ khối. Phần này cũng nói về danh sách (Lists) và làm thế nào để sử dụng các ảnh trong tài liệu HTML

1.3.1Các thẻ định dạng

Phần này ta sẽ học ba thẻ mức đoạn văn bản: <BLOCKQUOTE> và <PRE>

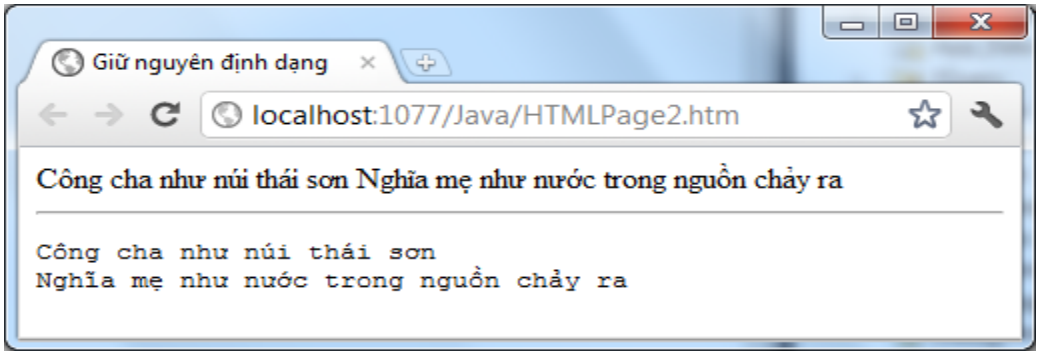
1.3.1.1Thẻ giữ nguyên định dạng

Thẻ <PRE> sẽ giữ nguyên định dạng văn bản như trên trình soạn thảo. Ví dụ sau sẽ cho bạn thấy điều đó.

```
<html>
<head>
    <title>Giữ nguyên định dạng</title>
</head>
<body>

Công cha như núi thái sơn
Nghĩa mẹ như nước trong nguồn chảy ra
<hr />
<pre>
Công cha như núi thái sơn
Nghĩa mẹ như nước trong nguồn chảy ra
</pre>

</body>
</html>
```



Hình 4.7: Giữ nguyên định dạng

1.3.1.2 Các thẻ tiêu đề

Với định dạng in đậm và kích thước to nhỏ khác nhau nên các thẻ <H1>….<H6> thường dùng làm tiêu đề. <H1> có kích thước to nhất và <H6> có kích thước nhỏ nhất.

Ví dụ 1:

```
<HTML>
    <HEAD><TITLE>Cac the co ban</TITLE></HEAD>
    <BODY>
        <H1>The tieu de H1</H1>
        <H2>The tieu de H2</H2>
        <H3>The tieu de H3</H3>
        <H4>The tieu de H4</H4>
        <H5>The tieu de H5</H5>
        <H6>The tieu de H6</H6>
    </BODY>
</HTML>
```

Hình 4.8: Thẻ thiêu đề

1.3.1.3Các thẻ định dạng thường dùng

Thẻ	Mô tả	Ví dụ	
<B>, <STRONG>	In đậm	<P><B>This is good fun</B></P>	**This is good fun**
<I>, <EM>	In nghiêng	<P><EM>This is good fun</EM></P>	*This is good fun*
^{. . .}	Chỉ số trên	<P>pi*r²</P>	pi*r^2
_{. . .}	Chỉ số dưới	<P>H₂O</P>	H_2O

1.3.1.4Màu và ký tự đặc biệt

✪ **Sử dụng ký tự đặc biệt**

Bạn có thể chèn các ký tự đặc biệt vào văn bản của tài liệu HTML. Để đảm bảo trình duyệt không nhầm chúng với thẻ HTML, bạn phải gán mã định dạng cho các ký tự đặc biệt này.

Ký tự đặc biệt	Mã định dạng	Ví dụ
Lớn hơn (>)	&gt;	If A > B Then A = A + 1
Nhỏ hơn (<)	&lt;	If A < B Then A = A + 1
Trích dẫn(““)	&quot;	" To be or not to be ? " That is the question
Ký tự “&”	&amp;	<P> William & Graham went to the fair

✪ **Sử dụng màu sắc**

Màu được sử dụng nhiều để thiết lập màu sắc cho các thành phần trên trang web. Để thiết lập màu chúng ta có 2 cách là tên màu (red, yellow, green…) hoặc trộn từ 3 màu cơ bản (#RRGGBB) (đỏ, xanh và xanh da trời). Mỗi màu chính được xem như một bộ hai số của hệ 16.

Số thập lục phân 00 chỉ 0% của màu trong khi đó số thập lục phân FF chỉ 100% của màu. Giá trị cuối cùng là một mã sáu chữ số chỉ màu.

Mã thập lục phân	Màu
#FF0000	Red
#00FF00	Green
#0000FF	Blue
#000000	Black
#FFFFFF	White

1.3.2Các thẻ multimedia

1.3.2.1Chèn ảnh

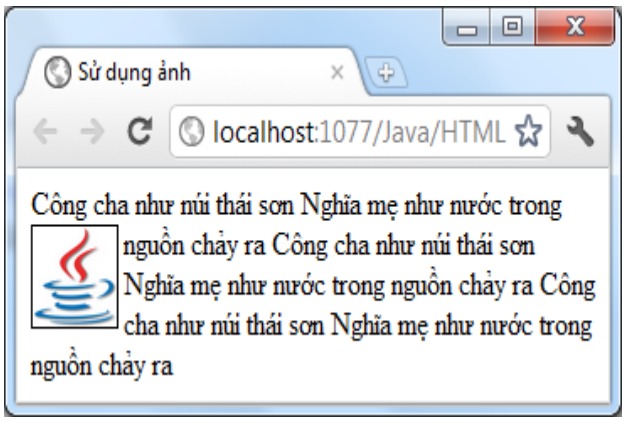
Thẻ <img> được sử dụng để chèn ảnh vào trang web. Sau đây là cú pháp thẻ và các thuộc tính thường dùng.

```
<img align="left" width="50px" height="50px" border="1" src="java.png">
```

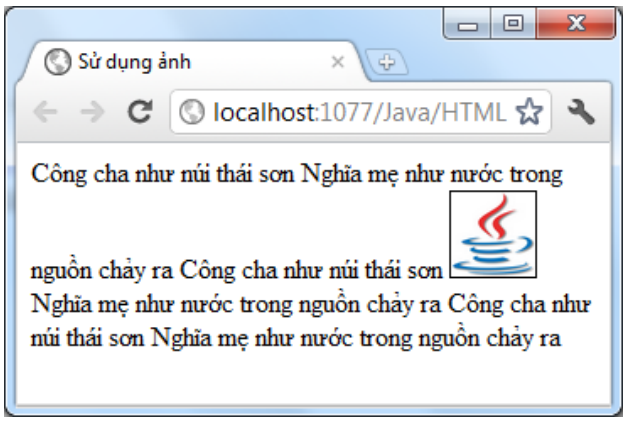
Hầu hết các thuộc tính đều rõ nghĩa, riêng thuộc tính align bạn cần lưu ý cách sử dụng của nó. Để hiểu rõ bạn hãy xem ví dụ sau:

```
<html>
<head>
    <title>Sử dụng ảnh</title>
</head>
<body>
    Công cha như núi thái sơn Nghĩa mẹ như nước trong nguồn chảy ra Công cha như núi
    thái sơn
    <img align="left" width="50px" height="50px" border="1" src="java.png">
    Nghĩa mẹ như nước trong nguồn chảy ra Công cha như núi thái sơn Nghĩa mẹ như nước
    trong nguồn chảy ra
</body>
</html>
```

Điều gì se xảy ra nếu bỏ thuộc tính align của thẻ. Hai hình sau tương ứng với 2 tình huống.



Thuộc tính align=”left”



Không sử dụng thuộc tính align

Hình 4.9: Chèn hình

Chú ý: ngày nay kỹ thuật sử dụng hình ảnh thật sự đa dạng và phong phú. Người ta có thể sử dụng CSS để đưa ảnh làm nền hầu hết các thẻ trên web. Và vì vậy bất ký thể nào cũng có thể sử dụng để làm ảnh.

Ví dụ:

```
<span style=”background-image: url(java.png);width:50px;height:50px”/>
```

1.3.2.2Chèn âm thanh và video

Bạn có thể sử dụng thẻ <EMBED> để chèn nhạc, video hay flash.

Cú pháp của thẻ:

<EMBED SRC=”tập tin video hay nhạc” WIDTH=”chiều rộng” HEIGHT=”chiều cao”>

Ví dụ chèn nhạc hay video:

```
<EMBED SRC="love.mp3" HEIGHT=”100” WIDTH=”100”/>
```

Ví dụ chèn nhạc nền:

```
<EMBED SRC="love.mid" HIDDEN=”true”/>
```

Thuộc tính HIDDEN=”true” có nghĩa là không hiện trên trang web, chi nghe.

Ví dụ chèn flash

```
<EMBED TYPE="application/x-shockwave-flash" SRC="flash.swf"
WMODE="transparent"></EMBED>
```

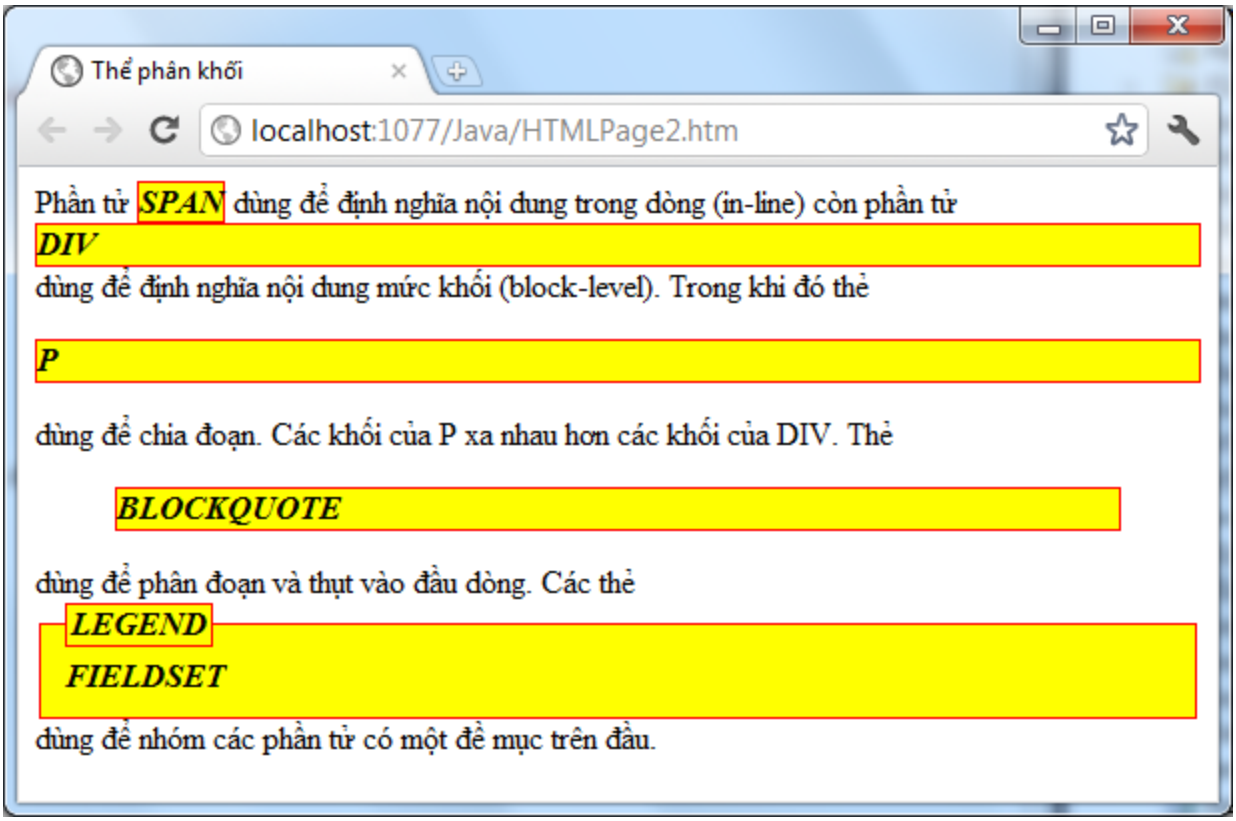
Thuộc tính **WMODE=”transparent”** cho biết flash trong suốt tức bỏ màu nền của flash.

1.3.3Thẻ phân khối

Phần tử **<SPAN>** dùng để định nghĩa nội dung trong dòng (in-line) còn phần tử **<DIV>** dùng để định nghĩa nội dung mức khối (block-level). Trong khi đó thẻ **<P>** dùng để chia đoạn. Các khối của P xa nhau hơn các khối của DIV. Thẻ **<BLOCKQUOTE>** dùng để phân đoạn và thụt vào đầu dòng. Các thẻ **<FIELDSET>** và **<LEGEND>** dùng để nhóm các phần tử có một đề mục trên đầu

Ví dụ sau cho thấy qui luật phân khối của các thẻ trên

```
<html>
<head>
    <title>Thể phân khối</title>
    <style type="text/css">
        span, div, p, blockquote, fieldset, legend
        {
            border: 1px solid red;
            background-color: Yellow;
            font-weight: bold;
            font-style:italic;
        }
    </style>
</head>
<body>
Phần tử <span>SPAN</span> dùng để định nghĩa nội dung trong dòng (in-line)
còn phần tử <div>DIV</div> dùng để định nghĩa nội dung mức khối (block-
level). Trong khi đó thẻ <p>P</p> dùng để chia đoạn. Các khối của P xa nhau
hơn các khối của DIV. Thẻ <blockquote>BLOCKQUOTE</blockquote> dùng để phân
đoạn và thụt vào đầu dòng. Các thẻ <fieldset><legend>LEGEND</legend>
FIELDSET</fieldset> dùng để nhóm các phần tử có một đề mục trên đầu.
</body>
</html>
```



Hình 4.10: Thẻ mức khối

✪ Ngắt dòng và thước kẻ ngang

Trong HTML để ngắt dòng bạn cần sử dụng thẻ <BR/>. Khi đó nội dung trang ở phía sau thẻ <br> sẽ xuống dòng dưới.

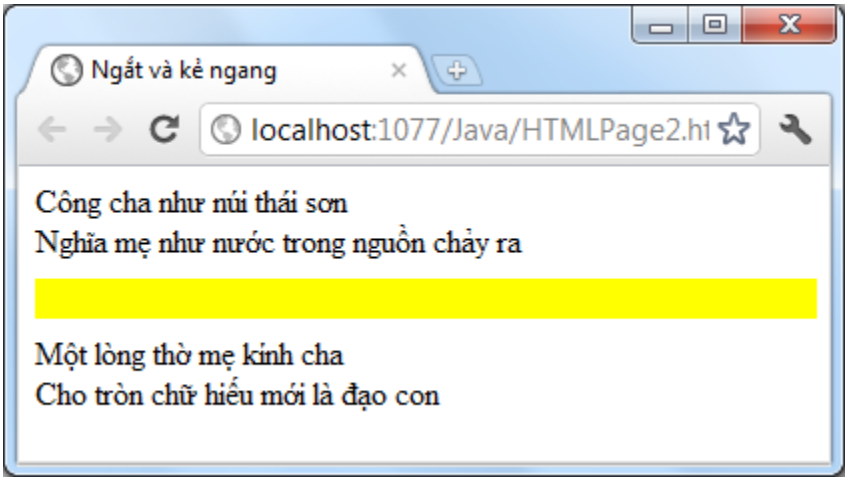
Thẻ <HR>(horizontal rule) được dùng để kẻ một đường ngang trên trang. Những thuộc tính sau giúp điều khiển các đường nằm ngang. Nó chỉ có thẻ bắt đầu, không có thẻ kết thúc và không có nội dung.

Thuộc tính	**Mô tả**
align	Chỉ định vị trí của đường nằm ngang. Chúng ta có thể canh lề center(giữa)|right(phải)|left(trái). Ví dụ align=center
width	Chỉ độ dài của đường thẳng. Nó có thể xác định bằng các pixel hoặc tính theo phần trăm. Mặc định là 100%, nghĩa là toàn bộ bề ngang của tài liệu.
size	Chỉ độ dày của đường thẳng và được xác định bằng các pixel.
noshade	Chỉ đường được hiển thị bằng màu đặc thay vì có bóng.

Ví dụ:

```
<html>
<head>
    <title>Ngắt và kẻ ngang</title>
```

```
</head>
<body>
    Công cha như núi thái sơn <br/>Nghĩa mẹ như nước trong nguồn chảy ra
    <hr width="100%" size="20px" align="left" color="yellow"/>
    Một lòng thờ mẹ kính cha <br/>Cho tròn chữ hiếu mới là đạo con
</body>
</html>
```



Hình 4.11: thước kẻ ngang và ngắt dòng

1.3.4Danh sách

Danh sách dùng để nhóm dữ liệu một cách logic. Chúng ta có thể thêm các danh sách vào tài liệu HTML để nhóm các thông tin có liên quan lại với nhau. Để tạo danh sách người ta dùng các thẻ <ul> (unorder list), <ol> (order list) và <li> (list item). Danh sách được chia làm 2 loại:

- ✓ Danh sách không thứ tự sử dụng <ul> và
- ✓ Danh sách có thứ tự sử dụng <ol> và

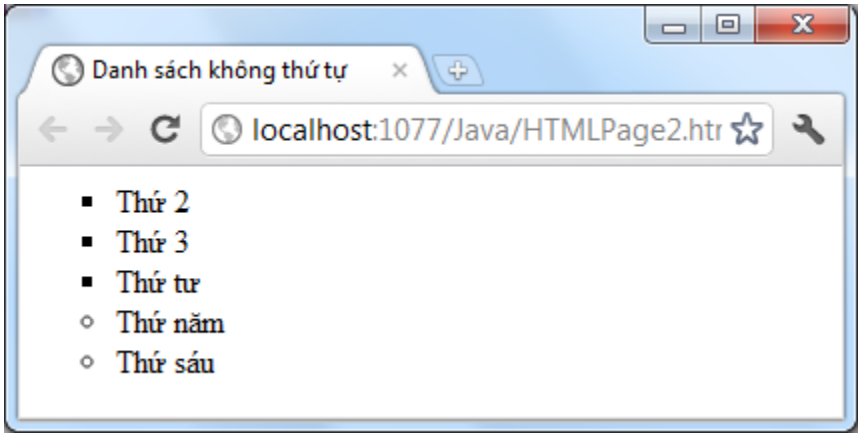
1.3.4.1Danh sách không thứ tự

Danh sách không thứ tự được nằm trong cặp thẻ <UL>…</UL>. Mỗi mục trong danh sách được đánh dấu bằng thẻ <LI>. Thể hiện của danh sách này là chấm tròn đặc, rỗng hoặc chấm vuông tùy vào kiểu lựa chọn. Cả 2 thẻ <UL> và <LI> để có thể thiết lập thuộc tính type. Giá trị của thuộc tính này (square, circle, disc) qui định kiểu thể hiện. Ví dụ sau tạo một danh sách toàn chấm vuông nhưng 2 mục cuối của danh sách là chấm tròn.

Ví dụ:

```
<HTML>
    <HEAD>
        <TITLE>Danh sách không thứ tự</TITLE>
    </HEAD>
    <BODY>
    <UL TYPE="square">
        <LI>Thứ 2</LI>
        <LI>Thứ 3</LI>
        <LI>Thứ tư</LI>
```

```
        <LI TYPE="circle">Thứ năm</LI>
        <LI TYPE="circle">Thứ sáu</LI>
    </UL>
    </BODY>
</HTML>
```



Hình 4.12: Danh sách không thứ tự

1.3.4.2Danh sách có thứ tự

Danh sách có thứ tự nằm trong cặp thẻ <OL>…</OL>. Danh sách có thứ tự cũng hiển thị các mục danh sách. Sự khác nhau là các mục danh sách hiển thị theo thứ tự được tạo ra một cách tự động. Loại của số thứ tự được qui định bởi thuộc tính TYPE của cả <OL> và <LI>.

Giá trị của thuộc tính TYPE

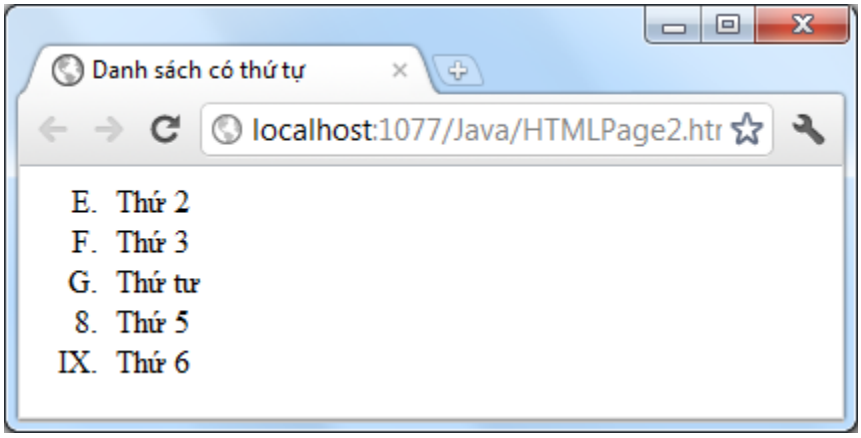
Thuộc tính TYPE	**Biểu hiện**
TYPE=”1”	1, 2, 3, 4…
TYPE=”a”	a, b, c, d…
TYPE=”A”	A, B, C, D…
TYPE=”i”	i, ii, iii, iv…
TYPE=”I”	I, II, II, IV…

Ngoài thuộc tính TYPE, <OL> còn có thuộc tính quan trọng khác là START qui định số bắt đầu của các mục trong danh sách.

Ví dụ:

```
<HTML>
<HEAD>
      <TITLE>Danh sách có thứ tự</TITLE>
</HEAD>
<BODY>
    <OL TYPE="A" START="5">
```

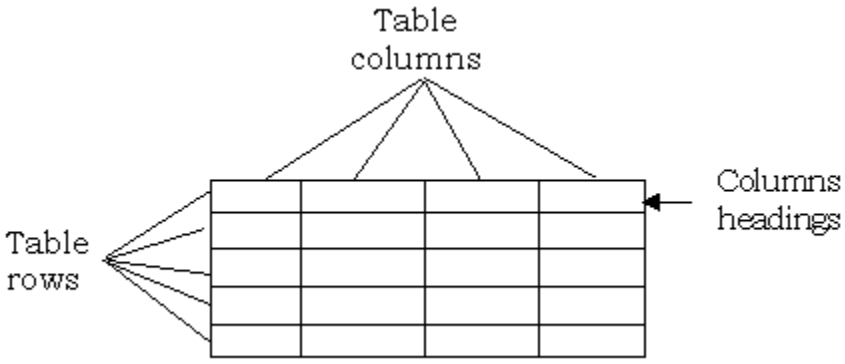
```
        <LI>Thứ 2</LI>
        <LI>Thứ 3</LI>
        <LI>Thứ tư</LI>
        <LI TYPE="1">Thứ 5</LI>
        <LI TYPE="I">Thứ 6</LI>
    </OL>
</BODY>
</HTML>
```



Hình 4.13: danh sách có thứ tự

1.4 Sử dụng bảng

Bảng không những dùng để hiển thị dữ liệu mà còn dùng để canh chỉnh, sắp xếp các thành phần web tuân theo sử trang trí của người thiết kế. Bảng có các hàng, các cột như mô hình sau.



Hình 4.14: Mô hình của bảng

Trong HTML để tạo bảng, bạn cần nắm 4 thẻ là:

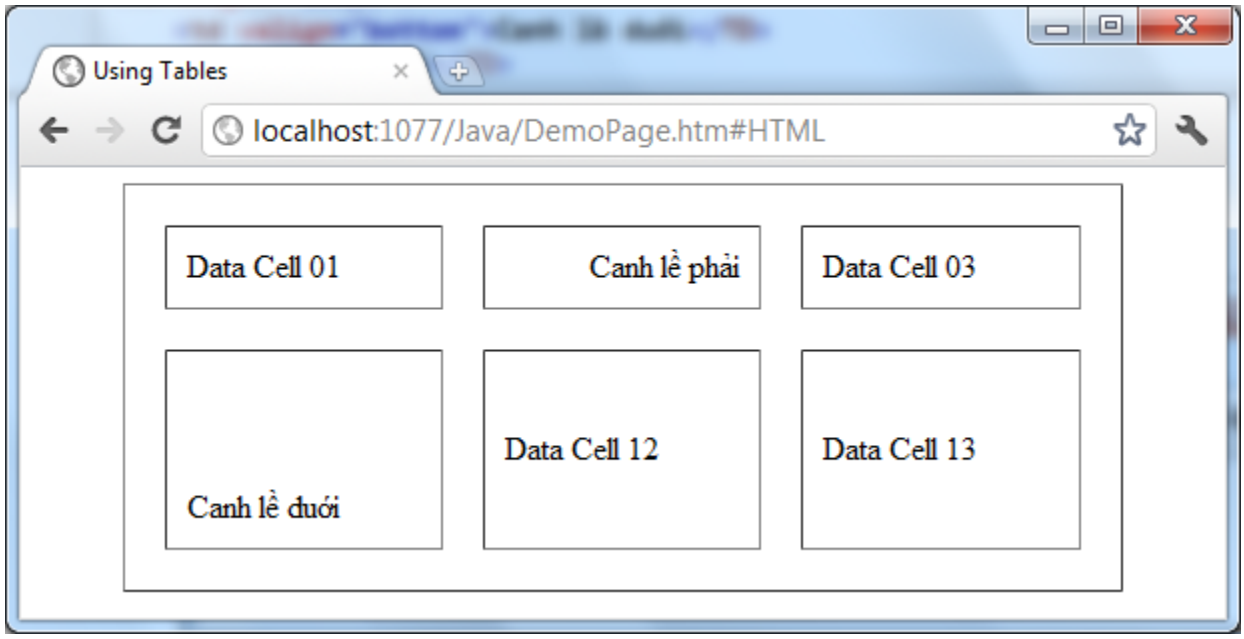
- ✓ <TABLE>: tạo bảng
- ✓ <TR>: tạo một hàng
- ✓ <TD>: tạo ô
- ✓ <TH>: tạo ô tiều đề (giống <TD> chỉ khác là nội dung được canh giữa và in đậm)

Bên cạnh học cách tạo bảng, bạn cũng cần nắm một số thuộc tính về bảng như border, cellspacing, cellpadding, align… và cách để gộp các ô bên trong bảng.

✪ **Tạo bảng**

Ví dụ sau tạo ra bảng 2 hàng 3 cột.

```
<HTML>
<HEAD>
    <TITLE>Using Tables</TITLE>
</HEAD>
<BODY>
<TABLE    BORDER="1"    width="500px"    cellpadding="10"    cellspacing="20"
align="center">
<TR>
    <TD>Data Cell 01</TD>
    <TD align="right">Canh lề phải</TD>
    <TD>Data Cell 03</TD>
</TR>
<TR height="100px">
    <td valign="bottom">Canh lề dưới</TD>
    <TD>Data Cell 12</TD>
    <TD>Data Cell 13</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```



Hình 4.15: Kết quả của ví dụ

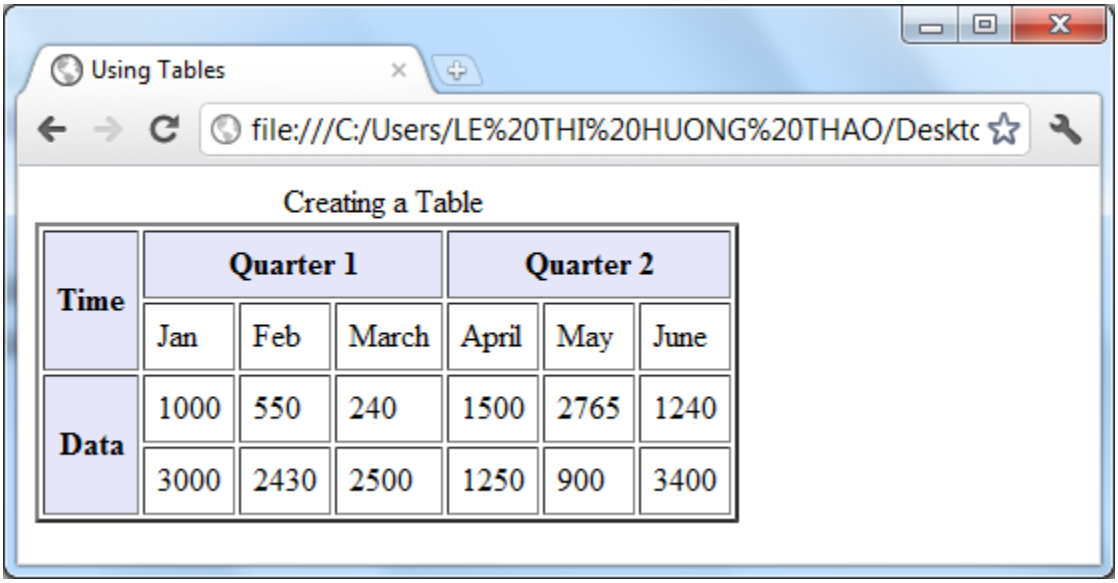
Chú ý một số đặc điểm kỹ thuật được sử dụng trên đây:

- ✓ Thuộc tính CELLSPACING qui định khoảng hở giữa các ô kề nhau.
- ✓ Thuộc tính CELLPADDING qui định khoảng đệm từ mép lề đến nội dung trông ô.
- ✓ Thuộc tính ALIGN qui định vị trí đặt của bảng.
- ✓ Thuộc tính BORDER qui định độ dày của đường kẻ.

✓ Thuộc tính ALIGN và VALIGN của mỗi ô qui định canh lề nội dung bên trong ô.

✪ **Trộn các ô**

Bảng sau cho chúng ta thấy các ô đã được trộn theo hàng (Time, Data) và cột (Quarter 1, Quarter 2)



Hình 4.16: Trộn ô

Đoạn HTML sau đây đã sinh ra bảng trên

```
<HTML>
<HEAD>
      <TITLE>Using Tables</TITLE>
</HEAD>
<BODY>
    <TABLE BORDER=2 CELLSPACING=2 CELLPADDING=6>
        <CAPTION>Creating a Table</CAPTION>
        <TR BGCOLOR=lavender>
            <TH ROWSPAN=2>Time</TH>
            <TH ALIGN=CENTER COLSPAN=3>Quarter 1</TH>
            <TH ALIGN=CENTER COLSPAN=3>Quarter 2</TH>
        </TR>
        <TR>
            <TD>Jan</TD>
            <TD>Feb</TD>
            <TD>March</TD>
            <TD>April</TD>
```

```
            <TD>May</TD>
            <TD>June</TD>
        </TR>
        <TR>
            <TH ROWSPAN="2" BGCOLOR=lavender>Data</TH>
            <TD>1000</TD>
            <TD>550</TD>
            <TD>240</TD>
            <TD>1500</TD>
            <TD>2765</TD>
            <TD>1240</TD>
        </TR>
        <TR>
            <TD>3000</TD>
            <TD>2430</TD>
            <TD>2500</TD>
            <TD>1250</TD>
            <TD>900</TD>
            <TD>3400</TD>
        </TR>
    </TABLE>
</BODY>
</HTML>
```

Chú ý: khi trộn 2 ô trên 2 hàng lại với nhau thì hàng sau sẽ bị mất một ô

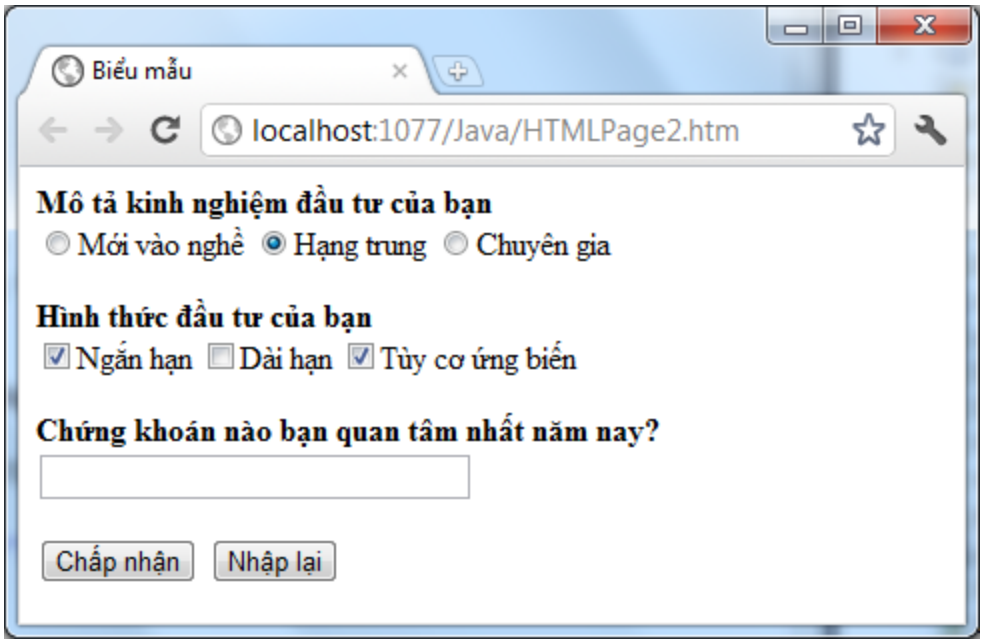
1.5 Sử dụng biểu mẫu

Biểu mầu được sử dụng để tiếp nhận dữ liệu từ người dùng hoặc hiển thị dữ liệu để người dùng xem. Đây là thành phần cực kỳ quan trong đối với lập trình web. Bạn cần phải nắm rõ các thẻ và thuộc tính của nó để phục vụ cho việc viết mã lập trình điều khiển dữ liệu.

Nói đến biểu mẫu trong HTML người ta thường nhắc đến 4 thẻ quan trọng là

- ✓ <FORM>: khung chứa các phần tử nhập liệu
- ✓ <INPUT>: sinh 10 phần tử nhập liệu và nút nhấn tùy vào thuộc tính type
- ✓ <SELECT>: sinh các phần tử combo box và list box tùy vào thuộc tính multiple và size
- ✓ <TEXTAREA>: sinh ô nhập nhiều dòng

Ví dụ giao diện và mã HTML tương ứng



Hình 4.17: Biểu mẫu

```
<HTML>
<HEAD>
    <TITLE>Biểu mẫu </TITLE>
</HEAD>
<BODY>
    <FORM ACTION="MyServlet.do" METHOD="POST">
        <p><B>Mô tả kinh nghiệm đầu tư của bạn</B><BR>
        <INPUT TYPE="RADIO" NAME="rdoInvExp" VALUE="0">Mới vào nghề
        <INPUT TYPE="RADIO" NAME="rdoInvExp" VALUE="1" checked>Hạng trung
        <INPUT TYPE="RADIO" NAME="rdoInvExp" VALUE="2">Chuyên gia
        <p><B>Hình thức đầu tư của bạn</B><BR>
        <INPUT TYPE="CHECKBOX" NAME="chkTypeInv" VALUE="0" checked>Ngắn hạn
        <INPUT TYPE="CHECKBOX" NAME="chkTypeInv" VALUE="1">Dài hạn
        <INPUT TYPE="CHECKBOX" NAME="chkTypeInv" VALUE="2”>Tùy cơ ứng biến
        <p><B>Chứng khoán nào bạn quan tâm nhất năm nay?</B><BR>
        <INPUT TYPE="TEXT" NAME="txtStockPick" SIZE="30" MAXLENGTH="30">
        <p>
        <INPUT TYPE="SUBMIT" NAME="Submit" VALUE="Chấp nhận">
        <INPUT TYPE="RESET" NAME="Reset" VALUE="Nhập lại">
    </FORM>
</BODY>
</HTML>
```

1.5.1Phần tử FORM

Phần tử <FORM> được sử dụng để tạo một vùng trên trang như một biểu mẫu. Nó chỉ ra cách bố trí của biểu mẫu. Các thuộc tính bao gồm:

Thuộc tính	Mô tả

ACCEPT	Thuộc tính này xác định danh sách các kiểu MIME được máy chủ nhận ra.
ACTION	Thuộc tính này xác định trang web nhận và xử lý dữ liệu trên biểu mẫu
METHOD	Thuộc tính này xác định phương thức dữ liệu được gửi đến máy chủ. Thông thường có 2 hình thức là GET và POST. Dữ liệu được ghép sau địa chỉ URL của thuộc tính ACTION để chuyển đến server nếu bạn sử dụng METHOD=”GET”. Nếu bạn dùng METHOD=”POST” thì dữ liệu sẽ được trình duyệt truyền cho trang web thông qua một luồng ngầm.

Ví dụ, để đưa một biểu mẫu đến chương trình “xử lý biểu mẫu” sử dụng theo phương thức POST

```
<FORM action=”http://mysite.com/mypage” METHOD=”post”>
...nội dung form...
</FORM>
```

1.5.2Các phần tử nhập của HTML

Khi tạo một biểu mẫu, ta có thể đặt các điều khiển lên biểu mẫu để nhận dữ liệu nhập vào từ người dùng. Các điều khiển này được sử dụng với phần tử <FORM>. Tuy nhiên, ta cũng có thể sử dụng chúng ở bên ngoài biểu mẫu để tạo các giao diện người dùng.

1.5.2.1Phần tử INPUT

Thuộc tính	**Mô tả**
TYPE	Thuộc tính này xác định loại phần tử. Ta có thể chọn một trong các lựa chọn: TEXT, PASSWORD, CHECKBOX, RADIO, FILE, HIDDEN, BUTTON, SUBMIT, RESET và IMAGE. Mặc định là TEXT
NAME	Bên cạnh để phân biệt trong lập trình Javascript, thuộc tính này còn được trang web sử dụng để nhận dữ liệu của phần tử.
VALUE	Chứa giá trị của phần tử. Không có ý nghĩa với IMAGE
SIZE	Độ rộng của phần tử, chỉ có ý nghĩa với TEXT, PASSWORD
MAXLENGTH	Số ký tự tối đa được nhập vào, chỉ có ý nghĩa với TEXT, PASSWORD
CHECKED	Xác định nút có được chọn hay không, chỉ có ý nghĩa với RADIO hay CHECKBOX
SRC	Chỉ ra địa chỉ của ảnh, chỉ có ý nghĩa với IMAGE

- **Ô nhập một dòng**

```
<input type="text" name="txtAbc" value="0" maxlength="" size="">
```

- **Ô nhập mật khẩu: người đứng sau lưng không nhìn thấy**

```
<input type="password" name="txtAbc" value="0" maxlength="" size="">
```

- **Phần tử ẩn: thường dùng cho lập trình hoặc các phần tử đã biết trước dữ liệu**

```
<input type="hidden" name="txtAbc" value="0">
```

- **Upload file: biểu mẫu chứa thẻ này phải là** `<form enctype="multipart/form-data">`

```
<input type="file" name="filAbc">
```

- **Ô nhập checkbox**

```
<input type="checkbox" name="chkAbc" value="0" checked>
```

- **Ô nhập radio**

```
<input type="radio" name="rdoAbc" value="0" checked>
```

- **Nút nhấn thực hiện lệnh JavaScript**

```
<input type="button" name="btnAbc" value="OK" onclick=”alert(‘Hello’)”>
```

- **Nút chuyển dữ liệu đến trang xử lý**

```
<input type="submit" name="btnAbc" value="OK">
```

- **Nút phục hồi trạng thái ban đầu của form**

```
<input type="reset" name="btnAbc" value="Cancel">
```

- **Nút cùng chức năng với SUBMIT chỉ khác về diện mạo là hình ảnh**

```
<input type="image" name="btnAbc" src="ok.gif" width=”80px” height=”25px”>
```

1.5.2.2Phần tử TEXTAREA

Thẻ <TEXTAREA> cho phép người dùng nhập nhiều dòng văn bản. Các thuộc tính quan trọng của TEXTAREA gồm:

Thuộc tính	**Mô tả**
COLS	Độ rộng của textarea
ROWS	Độ cao của textarea
TYPE	Luôn có giá trị là textarea
VALUE	Phần văn bản nhập vào hoặc phần thân của thẻ

Ví dụ:

```
<TEXTAREA NAME=”txtGhiChu" COLS="50" ROWS="5"> </TEXTAREA>
```

1.5.2.3Phần tử BUTTON (Nút bấm)

Thẻ này được sử dụng để thay thế chi BUTTON, RESET, SUBMIT và IMAGE tùy vào thuộc tính type của nó. Nhã của nút được thiết kế tùy ý tại phần thân của thẻ. Sau đây là các thuộc tính thường dùng

Thuộc tính	**Mô tả**

NAME	Gán tên cho nút
VALUE	Gán giá trị cho nút
TYPE	Xác định loại nút. Các giá trị có thể là: **Submit** – có tác dụng như <input type=”submit”> **Button** – có tác dụng như <input type=button”> **Reset** – có tác dụng như <input type=”reset”>

Ví dụ sau minh họa cho việc sử dụng phần tử BUTTON

```
<button type="submit" name="submit" value="submit">
    <img src="OK.png" />Send
</button>
```

1.5.2.4Phần tử lựa chọn (Select)

Phần tử SELECT được sử dụng để hiển thị một danh sách các lựa chọn cho người dùng. Mỗi lựa chọn được biểu diễn bởi phần tử OPTION.

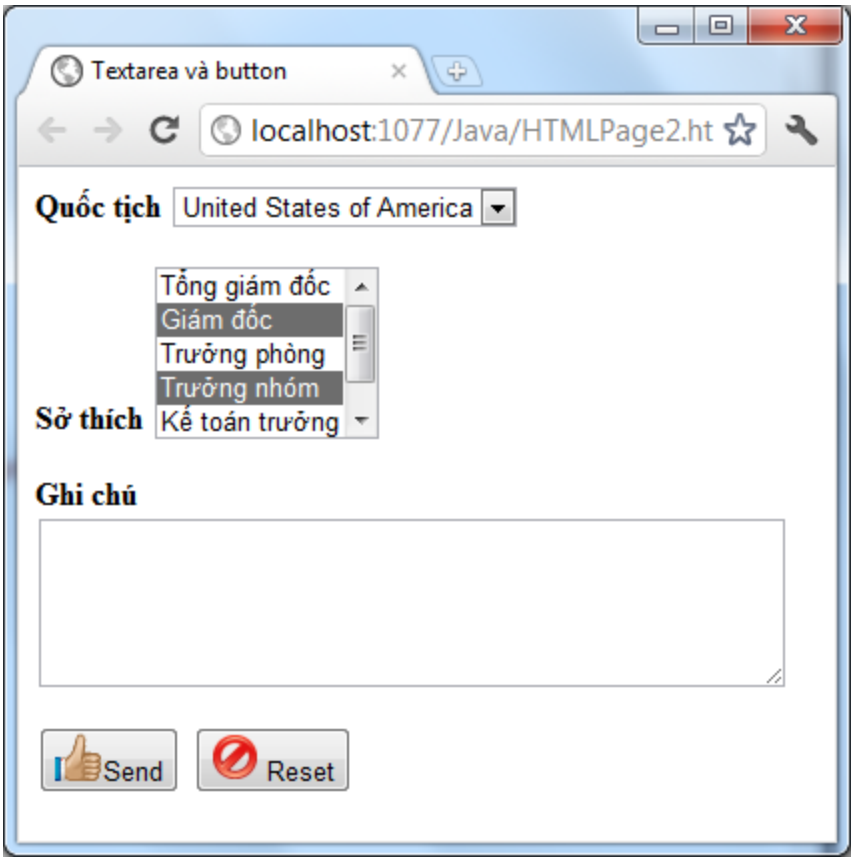
Thuộc tính	**Mô tả**
NAME	Tên của phần tử
SIZE	Số mục nhìn thấy, các mục còn lại phải cuộn mới thấy được.
MULTIPLE	Xuất hiện thuộc tính này thì cho phép chọn nhiều mục, ngược lại chỉ cho chọn một

Ví dụ sau minh họa việc sử dụng các thẻ <button>, <textarea> và <select>

```
<html>
<head>
    <title>Textarea và button </title>
    <style>
    b{width: 100px;}
    </style>
</head>
<body>
    <form action="http://www.mysite.com/mypage" method="POST">

    <p><b>Quốc tịch</b>
    <SELECT NAME="cboQuocTich">
```

```
        <OPTION value="VN">Viêt Nam</OPTION>
        <OPTION value="US" selected>United States of America</OPTION>
        <OPTION value="SG">Singapore</OPTION>
        <OPTION value="UK">United Kindom</OPTION>
    </SELECT>
    <p><b>Sở thích</b>
    <SELECT NAME="lstSoThich" multiple size="5">
        <OPTION value="TGD">Tổng giám đốc</OPTION>
        <OPTION value="GD">Giám đốc</OPTION>
        <OPTION value="TP">Trưởng phòng</OPTION>
        <OPTION value="TN">Trưởng nhóm</OPTION>
        <OPTION value="KTT">Kế toán trưởng</OPTION>
        <OPTION value="NV">Nhân viên</OPTION>
    </SELECT>
    <p><b>Ghi chú</b>
    <textarea name="txtAdvice" rows="5" cols="44"></textarea>
    <p>
        <button type="submit" name="submit" value="submit">
            <img src="OK.png" />Send
        </button>
        <button type="reset" name="reset">
            <img src="No.png" /> Reset
        </button>
    </p>
    </form>
</body>
</html>
```



Hình 4.18: <select>, <button> và <textarea>

Bạn hãy lưu ý thẻ <OPTION> có 2 thuộc tính quan trọng. **VALUE** dùng để chứa dữ liệu chuyển đến server trong khi đó phần thân của thẻ là để hiển thị cho người dùng nhìn. **SELECTED** là thuộc tính dùng để xác định mục được chọn.

1.5.2.5Phần tử LABEL (Nhãn)

Thẻ **<LABEL for="id của phần tử">** được sử dụng để gắn nhãn cho một phần tử giao diện (INPUT, TEXTAREA, SELECT…). Bạn chỉ cần xác định id của phần tử trong thuộc tính FOR của thẻ <LABEL>. Khi bạn click vào nhãn thì phần tử được chỉ định sẽ tích cực.

```
<LABEL for="lastname">Last name: </LABEL>
<INPUT type="text" name="txtFirstName" id="lastname">
```

1.5.3Các thuộc tính quan trọng của phần tử giao diện

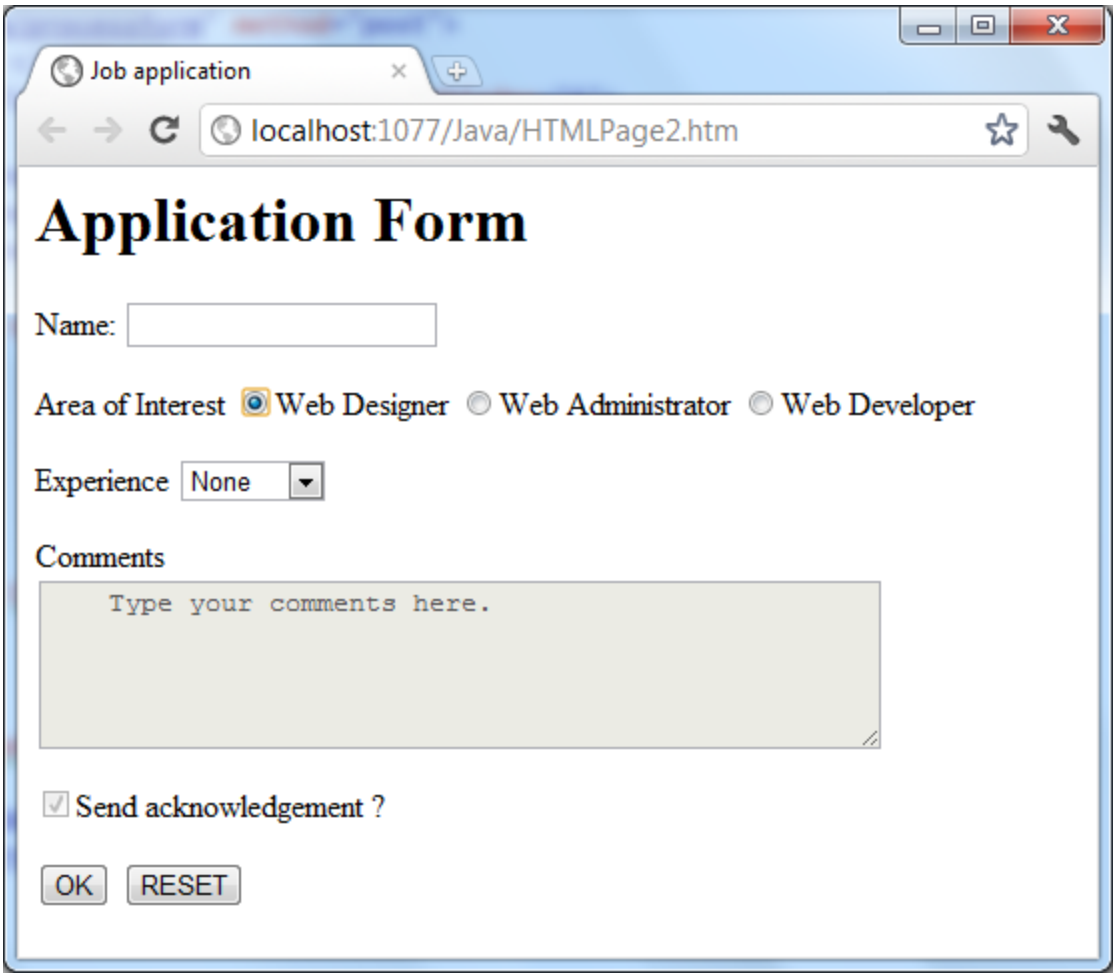
Bên cánh các thuộc tính của các phần tử giao diện đã nêu trên, bạn cũng cần lưu ý các thuộc tính không kém phần quan trọng khác được liệt kê dưới đây.

Thuộc tính	Mô tả	Ví dụ
TABINDEX	Thứ tự của phím TAB	<input tabindex=2>
ACCESSKEY	Ký tự nóng để truy xuất (ALT+hotkey)	<select accesskey="A">

READONLY	Chỉ đọc, không cho sửa	<textarea readonly>
DISABLED	Hủy phần tử khỏi form	<input disabled>

Ví dụ sau ứng dụng các thuộc tính trên. Chú ý phần tử READONLY và DISABLED sẽ không nhận phím TAB.

```
<HTML>
<HEAD>
    <TITLE>Job application</TITLE>
</HEAD>
<BODY>
<H1> Application Form</H1>
<FORM action="http://somesite.com/processform" method="post">
    <LABEL for="firstname">Name: </LABEL>
    <INPUT   type="text"   name="txtFirstName"   id="firstname"   tabindex="1"
accesskey="N">
    <P>Area of Interest
    <INPUT  TYPE=RADIO  NAME="rdoArea"  VALUE="0"  CHECKED  tabindex="2">Web
Designer
    <INPUT TYPE=RADIO NAME="rdoArea" VALUE="1">Web Administrator
    <INPUT TYPE=RADIO NAME="rdoArea" VALUE="2">Web Developer
    <P>Experience
    <SELECT NAME="cboExperience" tabindex="5">
        <OPTION>None</OPTION>
        <OPTION>1 Year</OPTION>
        <OPTION>3 Years</OPTION>
        <OPTION>5 Years</OPTION>
    </SELECT>
    <P>Comments<bR>
    <TEXTAREA NAME="txtComments" COLS="50" ROWS="5" disabled>
    Type your comments here.
    </TEXTAREA>
    <P>
    <input type="checkbox" checked readonly />Send acknowledgement ?
    <P>
    <INPUT TYPE="SUBMIT" VALUE="OK" tabindex="7">
    <INPUT TYPE="RESET" VALUE="RESET" tabindex="8">
</FORM>
</BODY>
</HTML>
```

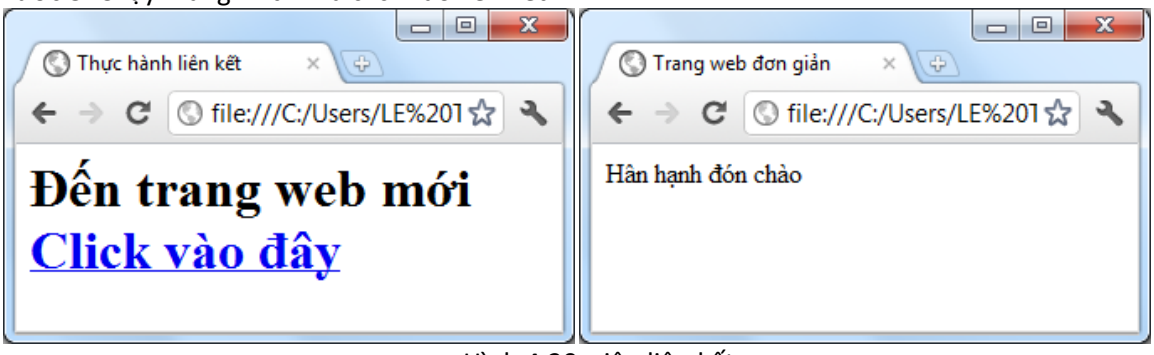


Hình 4.19: tabindex, accesskey, readonly và disabled

Khi chạy bạn hãy cố dùng phím TAB và ALT+N để kiểm tra phản ứng của giao diện.

1.6 Bài tập và thực hành

1. Liên kết giữa 2 trang
   Tạo trang web TrangA.html chứa liên kết đến TrangB.htm có giao diện như sau
   Bước 3: Chạy TrangA.html và click vào liên kết

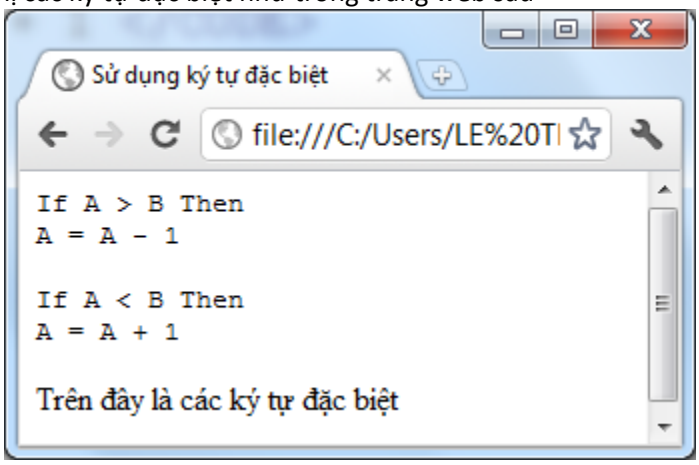


Hình 4.20: siêu liên kết

2. Tạo trang web có nhiều vùng (Trò chơi, Âm nhạc, Bưu thiếp và Gửi mail). Đặt các liên kết vùng lên đầu trang. Khi click chuột tại liên kết sẽ dẫn đến vùng tương ứng của liên kết.

Hình 4.21: liên kết vùng

3. Tạo trang web hiển thị các ký tự đặc biệt như trong trang web sau



Hình 4.22: Ký tự đặc biệt

4. Viết một trang web mô tả về gia đình và các vật nuôi của bạn. Kích vào liên kết "Mô tả gia đình" nó sẽ đưa bạn đến phần mô tả về gia đình . Khi kích vào liên kết "Mô tả vật nuôi" nó sẽ đưa bạn đến phần mô tả về vật nuôi của bạn.

   Gợi ý: Sử dụng neo "anchor" và những phần tử "đoạn"

5. Viết câu lệnh HTML để hiển thị dòng sau trong một trang web.

   For more information, please send an e-mail to me, Garywilson@MyCompany.com

   Gợi ý: Đặt câu lệnh sau vào trong phần <BODY> của file HTML.

   <A HREF=MAILTO:Garywilson@MyCompany.com"> Garywilson@MyCompany.com </A>

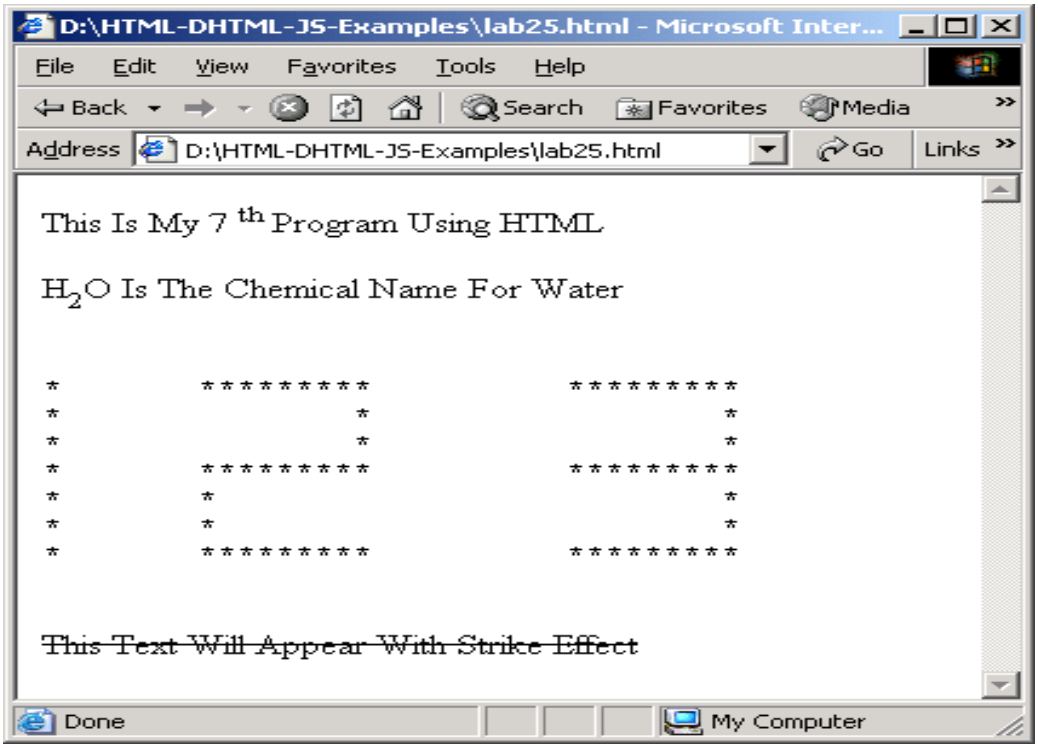
6. Viết đoạn mã HTML để kết hợp tất cả các thẻ định dạng được thảo luận sau đây:

```
<HTML>
<BODY>
```

```
<P> This Is My 7 <SUP> th </SUP> Program Using HTML </P>
<P> H<SUB>2</SUB>O Is The Chemical Name For Water</P>
<PRE>

 *         *********           *********
 *                 *                 *
 *                 *                 *
 *         *********           *********
 *         *                         *
 *         *                         *
 *         *********           *********

</PRE>
<STRIKE> This Text Will Appear With Strike Effect </STRIKE>
</BODY>
</HTML>
```



Hình 4.23: Kết quả

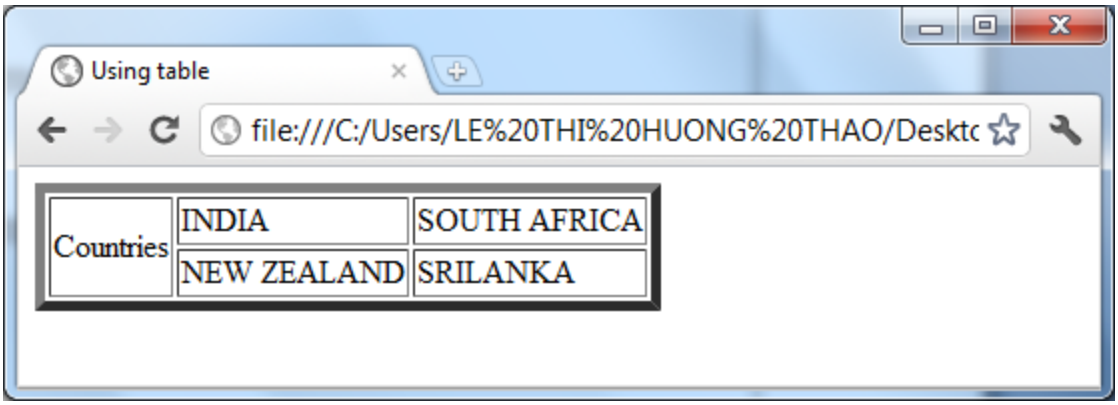
7. **Tạo trang web chứa danh sách như sau**

Hình 4.24: Kết quả

8. Viết một đoạn HTML cho trang Web với tiêu đề, “Using headings”. Thêm vào tiêu đề mức ba <H3> dòng chữ “My First HTML document”
9. Tạo một danh sách như sau:
    1. Introduction to HTML
        a. Introduction to the World Wide Web
        b. Introduction to HTML tags
            - Formatting text
            - Enhancing the text
    2. Designing a Web Site
        i. Designing the page
        ii. Designing navigation
        iii. Creating Hyperlinks
3. Tạo một trang web chứa một hình ảnh, một video và nhạc nền
5. Chèn ba ảnh vào tài liệu. Chèn ba đường thẳng xen kẽ giữa mỗi ảnh.
6. Tạo bảng như dưới:

Employee Name	**Designation**	**Salary**
John	Manager	4000
David	Officer	2500
Graham	Executive	3000

7. Tạo bảng sử dụng kỹ thuật gộp 2 hàng như hình sau



Hình 4.25: Trộn ô

8. Tạo một bảng để thống kê quá trình bán hàng hàng năm của công ty "My Company Limited". Hiển thị các số liệu hằng tháng được nhóm theo quí. Các sản phẩm là: kẹp giấy, đinh kẹp và bút
9. Viết đoạn mã HTML để tạo bảng như hình 6.3 dưới đây, bảng có đường viền dày 5 pixels

TAG NAME	DESCRIPTION	
TABLE	*To place a table in a web page*	
	TD	*To add columns*

Hình 4.26: Trộn ô

10. Tạo một biểu mẫu để nhận các thông tin chi tiết sau từ người dùng:

    Name – tên

    Address and telephone number – Địa chỉ và số điện thoại

    Sex, Age – Giới tính, tuổi

    Spoken languages – Ngôn ngữ (có ba chọn lựa)

    Cho người dùng chọn reset hay submit

11. Tạo một biểu mẫu hiển thị danh sách các cuốn sách. Nhận thông tin đặt hàng chi tiết từ người dùng:

    Name - tên

    Postal Address and telephone number – Địa chỉ đường bưu điện và số điện thoại

    Payment options – demand draft or credit card -Phương thức thanh toán – tiền mặt hay thẻ tài khoản

    Cho người dùng xác nhận hoặc hủy bỏ đơn đặt hàng

12. Tạo một biểu mẫu đặt mua báo dài hạn có các thông tin sau:

    Personal Information – thông tin người dùng

        Last Name, First Name

Address – Địa chỉ

City, State, Pin code -

Telephone number

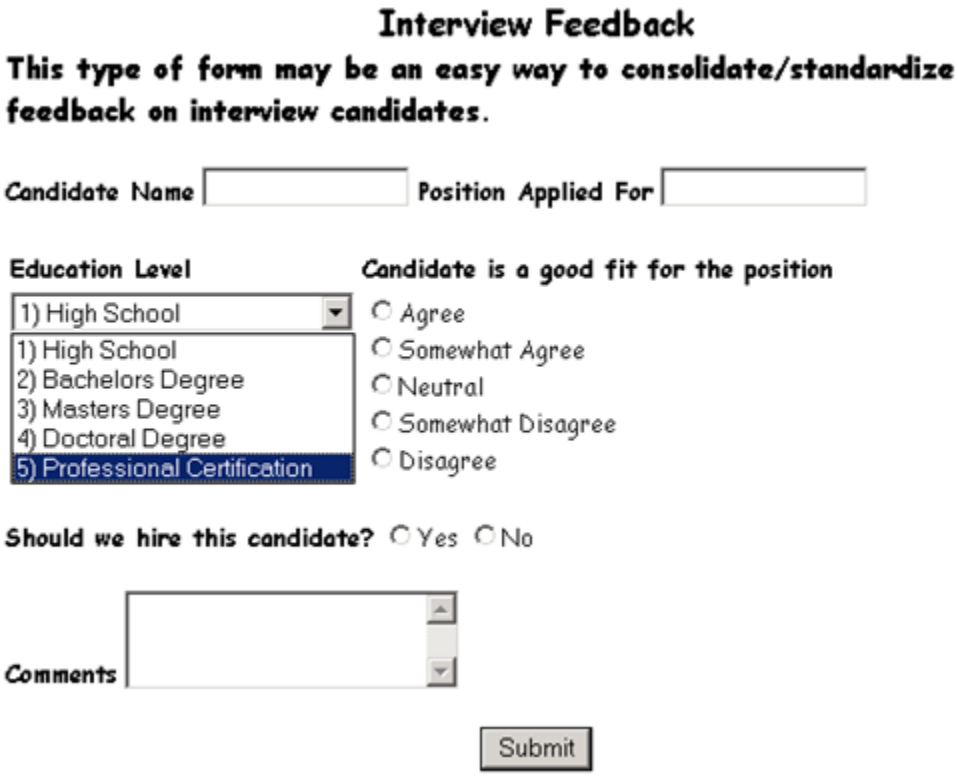
Báo dài hạn có 5 chọn lựa. Người dùng có thể chọn hơn một chọn lựa.

Thời gian đặt báo – 1 năm, 3 năm, 5 năm

Phương thức thanh toán – tiền mặt hay thẻ

Cho người dùng xác nhận hoặc hủy bỏ đơn đặt hàng

13. Viết một tài liệu HTML để tạo một biểu mẫu như hình sau



Hình 4.27: biểu mẫu

Gợi ý: Sử dụng phần tử TABLE, tạo bảng, để hiển thị *Education Level* và *Candidate is a Good Fit for the Position*

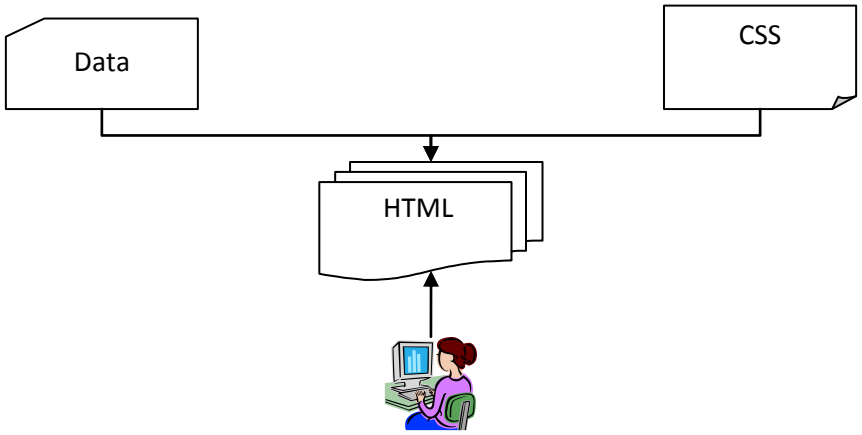
14. Tạo một biểu mẫu đển nhập thông tin họ và tên từ người dùng. Ngoài ra, hiển thị một danh sách ngày, tháng và năm, để người dùng chọn ngày sinh nhật của mình từ danh sách này.

2 BẢNG ĐỊNH KIỂU - CSS

2.1 Giới thiệu

Mong muốn của những người làm web của chúng ta là làm thế nào để bóc thông tin về kiểu dáng, diện mạo ra khỏi nội dung của các trang web. Nếu làm được điều này, chúng ta có được các lợi điểm sau:

- Dễ quản lý (đội ngũ và bảo trì). Khi cần thay đổi, chúng ta chỉ cần cập nhật có một nơi.
- Tái sử dụng. Một qui luật kiểu dáng có thể áp dụng cho nhiều thành phần web khác nhau
  - Trên mỗi trang
  - Trên các trang
  - Thậm chí trên các website.
- Cải thị tốc độ.
  - Giảm lượng thông tin kiểu dáng tải về: nhiều thành phần cùng kiểu dáng chỉ tải có một lần
  - Cách hiển thị của trình duyệt



Hình 5.1: mô hình hoạt động CSS

2.2 Khởi động nhanh

Để không gặp khó khăn của các qui luật của CSS, hãy cùng chúng tôi tìm hiểu một ví dụ đơn giản về nó. Qua đó chúng ta có dịp làm quen với các khái niệm của CSS để chuẩn bị cho cuộc đại khám phá CSS sắp đến.

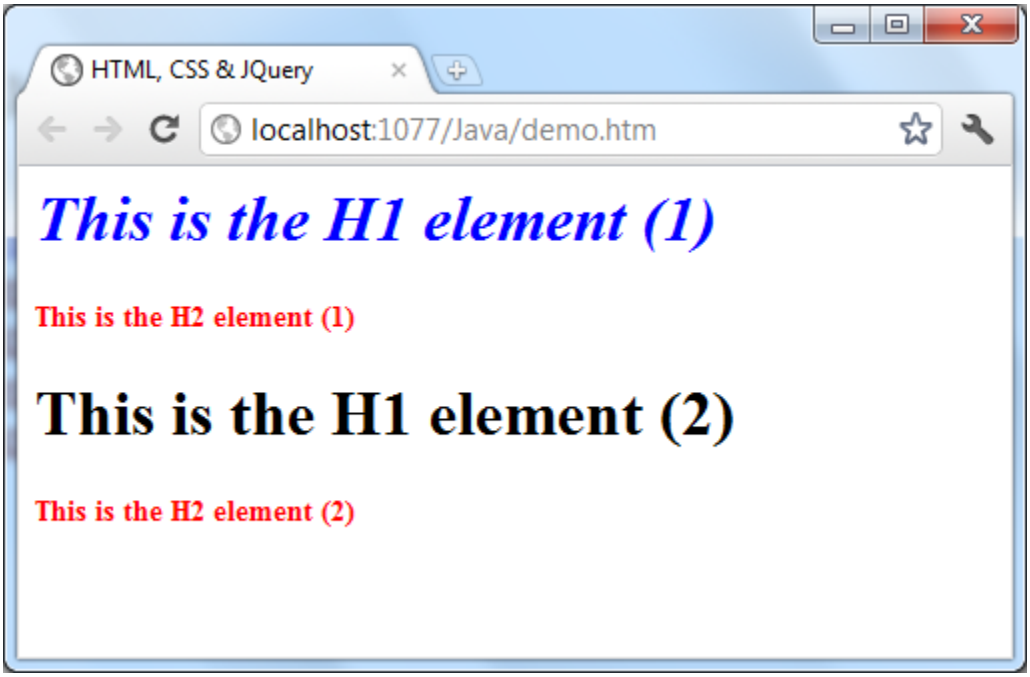
Chúng ta hãy xem và phân tích ví dụ sau đây

Mã nguồn HTML

```
<HTML>
<HEAD>
    <title>HTML, CSS & JQuery</title>
    <STYLE TYPE="text/css">
          H2{color:red;font-size: 14px;}
    </STYLE>
</HEAD>
    <H1 style="color:blue;font-style:italic;">
```

```
        This is the H1 element (1)</H1>
    <H2>This is the H2 element (1)</H2>
    <H1>This is the H1 element (2)</H1>
    <H2>This is the H2 element (2)</H2>
</HTML>
```

Kết quả thực hiện



Hình 5.2: Áp dụng CSS đơn giản

Phân tích ví dụ:

Ví dụ sau sẽ cho chúng ta dòng chữ "This is the H1 element (1)" lớn (do thẻ <h1>) màu xanh (do thuộc tính color có giá trị blue) và nghiêng (do thuộc tính font-style: italic).

```
<H1 style="color:blue;font-style:italic"> This is the H1 element (1)</H1>
```

Ví dụ trên chỉ cho một thẻ <H1> có qui luật kiểu dáng riêng mà thôi. Nếu chúng ta muốn tất cả các thể <H1> có cùng kiểu dáng như trên thì sao. Đơn giản thôi, chúng ta chép phần kiểu dáng (thuộc tính style) cho tất cả các thẻ <H1> trên trang là được chứ gì?. Không, đừng làm vậy bạn chẵn có lợi lọc gì cả so với phương pháp truyền thống trước đây.

Hãy xem phần định nghĩa cho thẻ <H2>. Tất cả những gì bạn phải làm ở đây là tách rời phần kiểu dáng của <H2> và định nghĩa riêng đặt giữa thẻ <style> thế là đủ. Trình duyệt sẽ tự động tìm kiếm các thể <H2> và áp dụng kiểu dáng cho bạn.

Với cách làm như định nghĩa cho thẻ <H2> chúng ta đã tiết kiệm được một lượng dữ liệu do không phải truyền kiểu dáng cho tất cả các thể <H2> do đó sẽ tăng tốc độ download của trang web. Nếu bạn đặt ra câu hỏi là "Tôi muốn áp dụng các luật CSS của <H2> cho tất cả các thẻ <H2> trên tất cả các trang web thì sao?". Thất đơn giản bạn chỉ việc tách phần định nghĩa CSS trên một tập tin kiểu dáng *.css sau đó liên kết vào bất kỳ trang nào bạn muốn.

Định nghĩa CSS trên một tập tin riêng (demo.css)

```
H2{color:red;font-size: 14px;}
```

Liên kết và áp dụng CSS

```
<HTML>
<HEAD>
    <title>HTML, CSS & JQuery</title>
    <link href="demo.css" rel="stylesheet" type="text/css" />
</HEAD>
    <H1 style="color:blue;font-style:italic;">
        This is the H1 element (1)</H1>
    <H2>This is the H2 element (1)</H2>
    <H1>This is the H1 element (2)</H1>
    <H2>This is the H2 element (2)</H2>
</HTML>
```

2.3 Kết luận

2.3.1Nên làm

- ✓ Bạn có thể sử dụng thuộc tính style của tất cả các thẻ HTML để định nghĩa kiểu dáng cho một lần sử dụng duy nhất.
- ✓ Bạn nên sử dụng thẻ <style> để định nghĩa cho các CSS dùng nhiều lần trong trang hiện tại
- ✓ Bạn nên sử dụng tập tin CSS để định nghĩa CSS cho các CSS được sử dụng nhiều nơi trên các trang của site.

2.3.2Lợi ích của các stylesheet

- ✓ **Nạp chồng trình duyệt:** - Mỗi trình duyệt đều có thể hiển thị các trang web theo cách riêng của nó. Trước đây các nhà phát triển không kiểm soát được các trang web hiển thị trên trình duyệt. Suy cho cùng bạn không biết trình duyệt nào mà người dùng cách nửa vòng trái đất sử dụng. Nhờ có các stylesheet bạn có thể nạp chồng các qui ước của trình duyệt và đặt theo cách riêng của chúng ta. Chẳng hạn, bạn có thể xác định kiểu mà trong đó một phần tử <H1> cần hiển thị:

  <H1><FONT SIZE=3 COLOR=AQUA>

  <B>Overriding the browser</B></FONT></H1>

- ✓ **Bố cục trang (Page layout) –** Những stylesheet có thể dùng để hiển thị font thay đổi màu mà không làm thay đổi cấu trúc của trang web. Điều này có nghĩa là với tư cách là một nhà thiết kế bây giờ bạn có thể tách biệt những yêu cầu về thiết kế hình ảnh trực quan từ cấu trúc logic của trang web và địa chỉ là hai chuyện hoàn toàn khác nhau.

  Khi sử dụng các biện pháp liên quan trong stylesheet của bạn, bạn có thể thể hiện các tài liệu sao cho đẹp mắt trên bất kỳ màn hình nào và theo bất kỳ độ phân giải nào.

- ✓ **Sử dụng lại các stylesheet –** Một khi đã định nghĩa kiểu thông tin, chúng ta có thể nhúng stylesheet bên trong tài liệu HTML. Lần lượt thay thế, chúng ta có thể kết nối tất cả các trang trên website đến stylesheet. Điều này chắc chắn rằng các trang web của chúng ta đều có cùng diện mạo thông tin khi được hiển thị. Vì vậy, bạn có thể có được nền chung của trang ví dụ như logo của trang và một số thông tin chuẩn(cho các trang) trong một stylesheet. Điều này sẽ đảm bảo được cách nhìn và cảm

nhận thông dụng về trang website. Cứ thử hình dung xem có vài trăm trang web và bạn phải xác định kiểu của mỗi trang một cách độc lập.

- ✓ **Chỉ cần làm một lần thật tốt –** Chúng ta có thể tạo một stylesheet và được liên kết đến nhiều tài liệu. Tất cả những tài liệu sẽ có diện mạo giống nhau. Tuy nhiên, quan trọng nhất là khi bạn thực hiện thay đổi stylesheet thì tất cả các tài liệu được kết nối vào stylesheet sẽ bị thay đổi theo.

2.4 Các qui ước trong CSS

2.4.1Qui tắc stylesheet

Stylesheet phân cấp(cascading style sheet) định nghĩa các kiểu có thể được áp dụng vào các trang hoặc các phần tử của trang.

- ✓ **Qui tắc kiểu (Style Rule)**- Stylesheet phân cấp là một tập hợp các qui tắc. Qui tắc định nghĩa kiểu của tài liệu. Ví dụ, chúng ta có thể tạo ra một qui tắc kiểu được xác định cho tất cả phần tiêu đề <H1> hiển thị màu vàng xanh. Qui tắc kiểu có thể ứng dụng vào các thành phần được chọn của trang web. Ví dụ, chúng ta có thể xác định một đoạn văn bản bất kỳ in đậm và in nghiêng trên trang. Điều này được gọi là **khai báo kiểu có sẵn** mà nhờ đó các kiểu được áp dụng vào các phần tử HTML đơn lẻ trên một trang web.
- ✓ **Style Sheet –** Là một danh mục các qui tắc kiểu và có thể nhúng vào bên trong tài liệu HTML. Trong trường hợp đó, nó được gọi là stylesheet nhúng. Stylesheet cũng có thể được tạo ra bằng một file bên ngoài và được liên kết với tài liệu HTML.
- ✓ **Các qui tắc (Rules) –** Bảng kiểu có thể có một hay nhiều qui tắc. Phần đầu của qui tắc gọi là bộ chọn (Selector). Mỗi bộ chọn có các thuộc tính và các giá trị liên quan đến nó.

  RuleSelector {Declarations property: value; property: value; ... }

  Phần của qui tắc được kèm theo trong phạm vi các dấu ngoặc móc được gọi là **khai báo**. Khai báo có hai phần, phần trước dấu hai chấm (:) là thuộc tính và phần nằm sau dấu hai chấm là giá trị của thuộc tính đó.

  Các khai báo được phân cách bởi dấu chấm phẩy (;). Ta nên đặt dấu chấm phẩy sau lần khai báo cuối cùng. Ví dụ như

```
H1 {color: blue}
```

  Ở đây, **H1** là ***bộ chọn***, **color: blue** là ***khai báo***, trong phạm vi khai báo: {Property: Value} thì c**olor** là ***thuộc tính***, **blue** là ***giá trị***.

2.4.2Định nghĩa các bộ chọn (Selector)

Bộ chọn (selector) là nơi định nghĩa các qui luật kiểu dáng để áp dụng cho các thành phần trên trang web. Ở đây bạn có một số cách định nghĩa khác nhau làm ảnh hưởng đến cách áp dụng ở các thẻ trên trang web. Trong phần này bạn sẽ học cách định nghĩa các bộ chọn và cách áp dụng của nó.

2.4.2.1Bộ chọn HTML (HTML Selector)

- ✓ Định nghĩa: định nghĩa kiểu dáng bổ sung cho các thẻ HTML

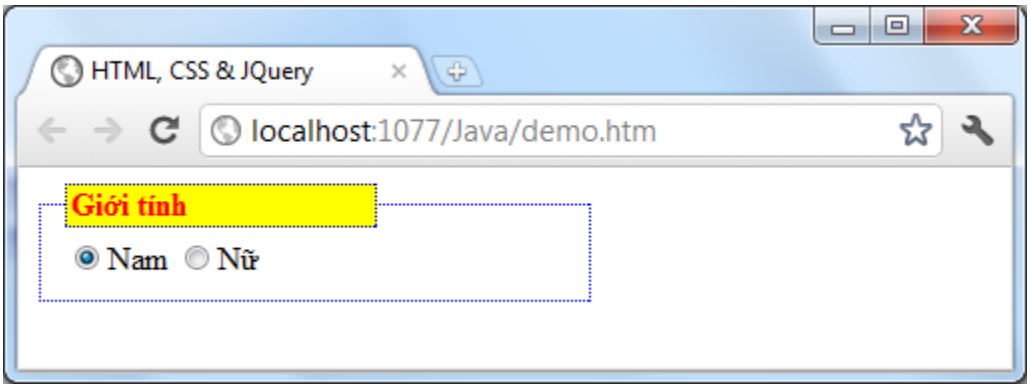
  <tên thẻ>{<khai báo các thuộc tính css>}

- ✓ Áp dụng: Tự động áp dụng các qui luật css trong phần khai báo cho tất cả các thẻ có tên là <tên thẻ>

✓ Ví dụ sau đây định nghĩa lại thẻ <fieldset> và legend với các thuộc tính kích thước (width), đường kẻ (border), màu chữ (color), màu nền (background-color).

```
<HTML>
<HEAD>
    <title>HTML, CSS & JQuery</title>
    <STYLE TYPE="text/css">
    FIELDSET{
        width: 250px;
        border: 1px dotted #0000FF;
    }
    LEGEND{
        font-weight: bold;
        color: #FF0000;
        background-color: #FFFF00;
        border: 1px dotted #0000FF;
        width: 150px;
    }
    </STYLE>
</HEAD>
<body>
<fieldset>
      <legend>Giới tính</legend>
      <input type="radio" name="rdoGioiTinh" checked/>Nam
    <input type="radio" name="rdoGioiTinh" />Nữ
</fieldset>
</body>
</HTML>
```

Kết quả hiển thị



Hình 5.3: Bộ chọn HTML

2.4.2.2Bộ chọn lớp (Class Selector)

✓ Định nghĩa: định nghĩa một lớp được bắt đầu bởi dấu chấm (.) bên trong khai báo nhiều thuộc tính css để áp dụng cho bất kỳ thẻ nào chỉ định bởi thuộc tính class của nó.

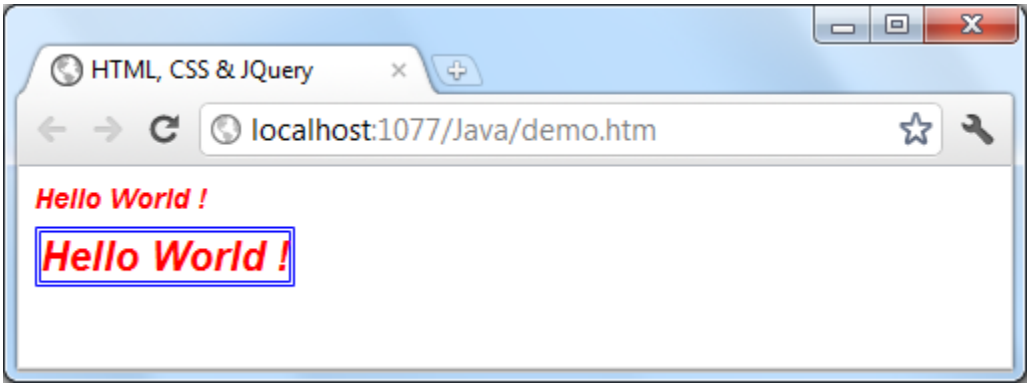
.<tên lớp>{<khai báo các thuộc tính css>}

- ✓ Áp dụng: tất cả các thẻ sử dụng thuộc tính class có giá trị là <tên lớp>. Chú ý thuộc tính class của mỗi thể có thể chỉ đến nhiều class cùng một lúc (cách nhau khoản trắng).
- ✓ Ví dụ sau định nghĩa 2 bộ chọn lớp sau đó thẻ <H1> áp dụng một còn thẻ <DIV> áp dụng cả hai để tận dụng các đặc điểm tổng hợp.

```
<HTML>
<HEAD>
    <title>HTML, CSS & JQuery</title>
    <STYLE TYPE="text/css">
    .MyHeader{
          font-family: Arial, Helvetica, sans-serif;
          font-weight: bold;
          font-style: italic;
          font-size: 14px;
          color: #FF0000;
    }
    .MyBorder{
        border: 3px double blue;
        font-size: 20px;
        display: inline;
    }
    </STYLE>
</HEAD>
<body>
    <h1 class="MyHeader">Hello World !</h1>
    <div class="MyHeader MyBorder">Hello World !</div>
</body>
</HTML>
```

- ✓ font-family: tên font chữ
- ✓ font-weight: độ đậm
- ✓ font-style: kiểu chữ
- ✓ font-size: kích thước chữ
- ✓ color: màu chữ

Kết quả thực hiện



Hình 5.4: Bộ chọn lớp

2.4.2.3Bộ chọn định danh (ID Selector)

✓ Định nghĩa: giống như bộ chọn lớp nhưng khởi đầu với dấu rào (#)

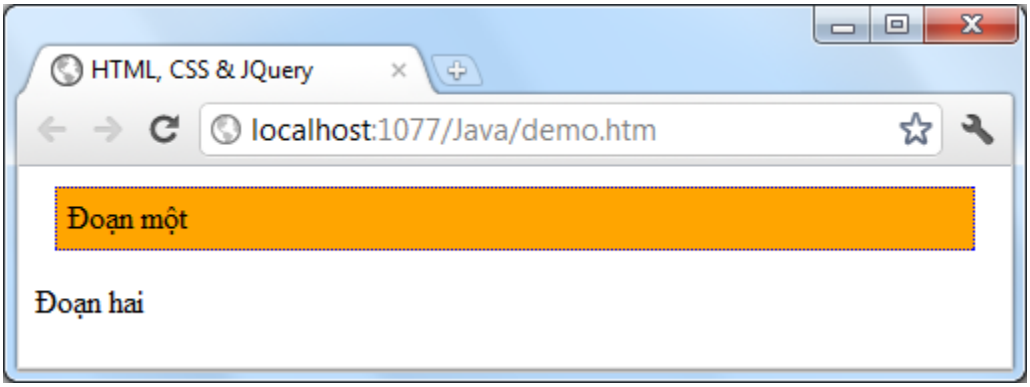
#<tên định danh>{<khai báo các thuộc tính css>}

✓ Áp dụng: tất cả các thẻ sử dụng thuộc tính id với giá trị là <tên định danh>

✓ Ví dụ sau định nghĩa bộ chọn định danh tên là #MyPara sau đó áp dụng cho một thẻ <P> trong trang web. Chú ý thẻ <P> còn lại không hề bị ảnh hưởng gì.

```
<HTML>
<HEAD>
    <title>HTML, CSS & JQuery</title>
    <STYLE TYPE="text/css">
    #MyPara
    {
          background-color: orange;
          background-image: url(images/abc.gif);
          text-align: justify;
          margin: 10px;
          padding: 5px;
          border: 1px dotted #0000FF;
    }
    </STYLE>
</HEAD>
<body>
    <p id="MyPara">Đoạn một</p>
    <p>Đoạn hai</p>
</body>
</HTML>
```

Kết quả thực hiện



Hình 5.5: Bộ chọn định danh

2.4.2.4Bộ chọn cho liên kết

✓ Định nghĩa: định nghĩa css cho siêu liên kết. Với liên kết có bốn trạng thái sử dụng là chưa thăm (chưa click), đã thăm, có chuột và tích cực (đang chọn). Vì vậy để định nghĩa CSS áp dụng cho liên kết bạn không chỉ định nghĩa CSS cho thẻ <A> mà còn định nghĩa cả 4 trạng thái của nó. Sau đây là cú pháp chung định nghĩa css cho siêu liên kết

A: {<khai báo các thuộc tính css>}

A:link {<khai báo các thuộc tính css>}

A: visited{<khai báo các thuộc tính css>}

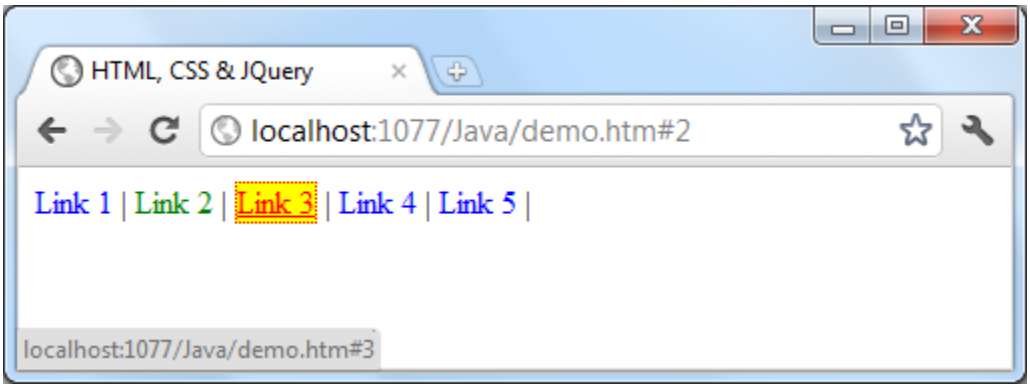
A: hover{<khai báo các thuộc tính css>}

A: active{<khai báo các thuộc tính css>}

- ✓ Áp dụng: Tất cả các liên kết trong trang có định nghĩa CSS cho liên kết
- ✓ Ví dụ

```
<HTML>
<HEAD>
    <title>HTML, CSS & JQuery</title>
    <STYLE TYPE="text/css">
    A{
        font-family: Arial
        font-size: 16px;
        text-decoration: none;
    }
    A:link{
        color: Blue;
    }
    A:visited {
        color: Green;
    }
    A:hover{
        text-decoration: underline;
        color: Red;
        border: 1px dotted Red;
        background-color: Yellow;
    }
    A:active {
        color: Orange;
    }
    </STYLE>
</HEAD>
<body>
    <a href="#1">Link 1</a> |
    <a href="#2">Link 2</a> |
    <a href="#3">Link 3</a> |
    <a href="#4">Link 4</a> |
    <a href="#5">Link 5</a> |
</body>
</HTML>
```

Kết quả thực hiện: Link3 đang có chuột, Link2 đã bị click trước đó



Hình 5.6: Bộ chọn liên kết

2.4.2.5Nhiều bộ chọn cùng kiểu

✓ Định nghĩa: định nghĩa nhiều bộ chọn đồng một số kiểu dáng.

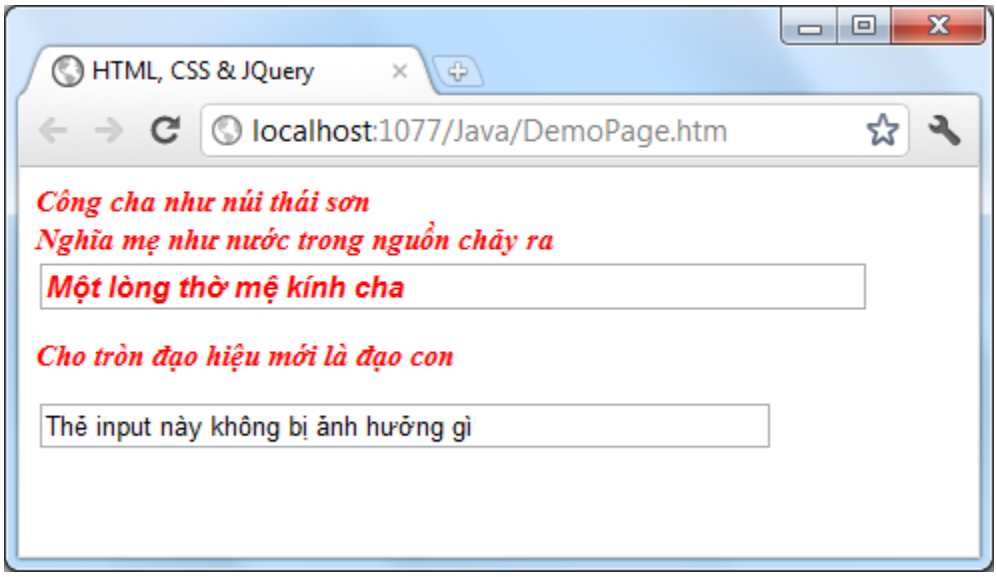
<bộ chọn 1>, <bộ chọn 2>,…,<bộ chọn n>{<khai báo các thuộc tính css>}

✓ Áp dụng: áp dụng các khai báo css cho tất cả các thẻ có chỉ định sử dụng css thõa mãn với các bộ chọn được liệt kê cách nhau dấu phẩy.

✓ Ví dụ

```
<html>
<head>
    <title>HTML, CSS & JQuery</title>
    <style type="text/css">
    #A, .B, DIV INPUT, H2
    {
          font-weight: bold;
          font-style: italic;
          color: #FF0000;
          font-size: 11pt;
    }
    </style>
</head>
<body>
    <div class="B">Công cha như núi thái sơn</div>
    <div id="A">Nghĩa mẹ như nước trong nguồn chảy ra</div>
    <div><input value="Một lòng thờ mệ kính cha" size="55" /></div>
    <h2>Cho tròn đạo hiếu mới là đạo con</h2>
    <input value="Thẻ input này không bị ảnh hưởng gì" size="55" />
</body>
</html>
```

Kết quả thực hiện



Hình 5.7: nhiều bộ chọn cùng kiểu

Lưu ý: "DIV INPUT" có nghĩa là định nghĩa css cho các thẻ <INPUT> đặt trong các thẻ <DIV>. Vì vậy trong bài này thẻ <input> không đặt trong không hề chịu tác dụng của css đã định nghĩa.

2.4.2.6Bộ chọn khoanh vùng

✓ Định nghĩa: định nghĩa CSS cho các vùng khác nhau trên trang. Như vậy chúng ta cần xác định vùng cần áp dụng và bộ chọn chứa các CSS để áp dụng.

<vùng> <bộ chọn>{<khai báo các thuộc tính css>}

✓ Áp dụng: Áp dụng CSS của bộ chọn cho các thẻ đặt trong <vùng> và chỉ định áp dụng bộ chọn
✓ Ví dụ

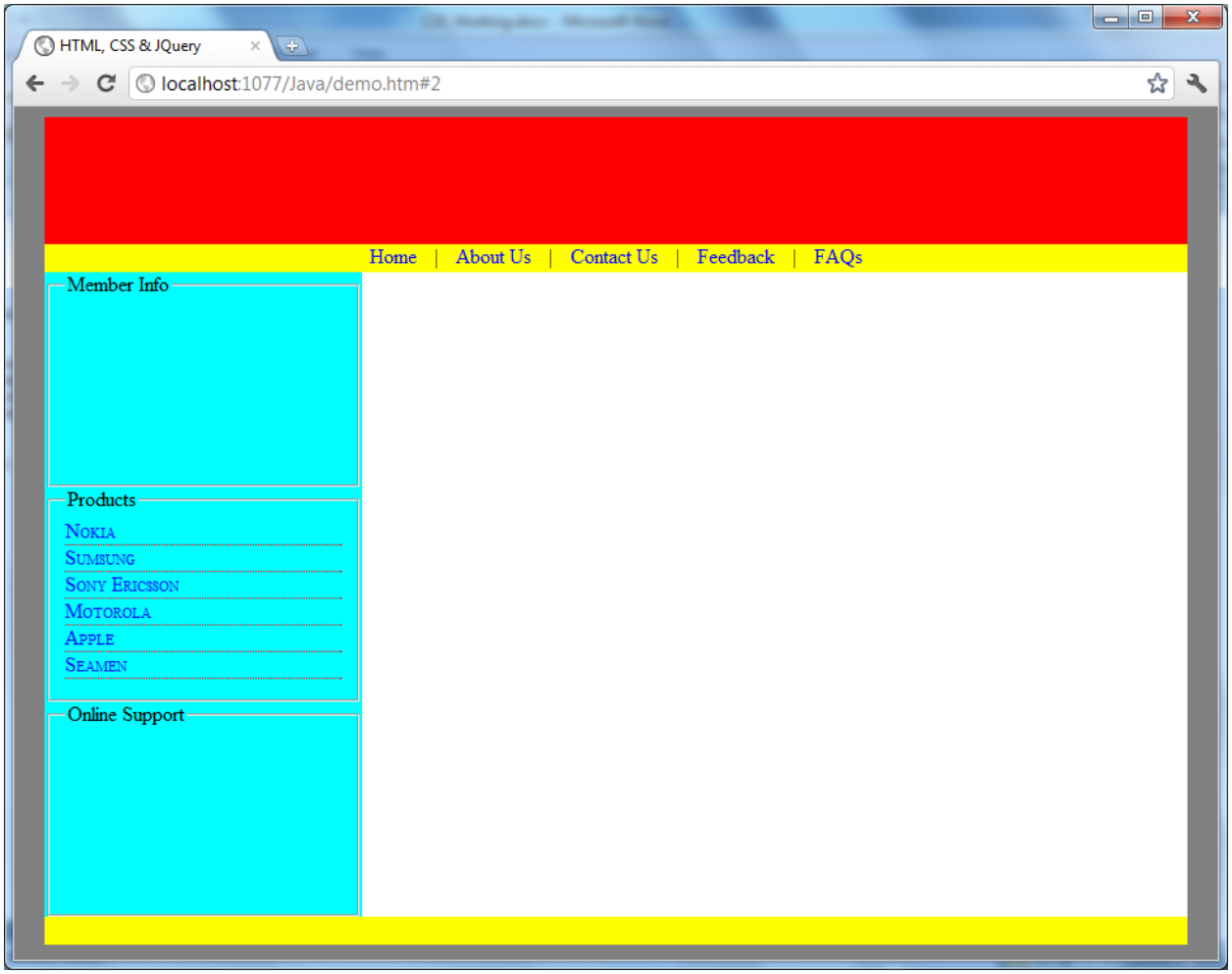
```
<HTML>
<HEAD>
    <title>HTML, CSS & JQuery</title>
    <STYLE TYPE="text/css">
        /*--vỏ bọc bên ngoài rộng 900px, canh giữa, nền trắng--*/
        .container{width:900px; margin: 0px auto; background-color: White;}
        /*--đầu trang cao 100px--*/
        .top{height: 100px; background-color: Red;}
        /*--menu trang cao 22px, canh giữa--*/
        .menu{height: 22px; background-color: Yellow; text-align:center}
        /*--giữa trang cao tối thiểu 400px--*/
        .middle{min-height: 400px;}
        /*--giữa-trái cao như middle, rộng 250px, gâm trái--*/
        .middle_left{
            float:left;width: 250px;min-height:inherit;
            background-color: Aqua;
        }
        /*--giữa-phải cao như middle, rộng 650px, gâm phải--*/
        .middle_right{
```

```
            float:right;width: 645px;min-height:inherit;
            background-color: White;
        }
        /*--chân trang không gâm, cao 22px--*/
        .bottom{clear:both;height: 22px; background-color: Yellow;}
        /*--fieldset trong .middle_left cao tối thiểu 150--*/
        .middle_left fieldset{min-height: 150px;}
        /*--li trong .middle_left không dùng dấu, kẽ chân--*/
        .middle_left li{list-style-type:none; border-bottom: 1px dotted red;}
        /*--liên kết trong .menu cách nhau 20px--*/
        .menu a{padding: 0px 10px 0px 10px;}
        /*--liên kết trong .middle_left chữ HOA nhỏ--*/
        .middle_left a{font-variant:small-caps;}
        /*--liên kết có chuột trong .middle_left in đậm--*/
        .middle_left a:hover{font-weight: bold;}
        /*--liên kết chung cho toàn trang--*/
        a{text-decoration: none;}
        a:link, a:active, a:visited{color: Blue;}
        a:hover{color: Red;}
        body{background-color: Gray;}

    </STYLE>
</HEAD>
<body>
    <div class="container">
        <div class="top"></div>
        <div class="menu">
            <a href="#1">Home</a> |
            <a href="#2">About Us</a> |
            <a href="#3">Contact Us</a> |
            <a href="#4">Feedback</a> |
            <a href="#5">FAQs</a>
        </div>
        <div class="middle">
            <div class="middle_left">
                <fieldset>
                    <legend>Member Info</legend>
                </fieldset>
                <fieldset>
                    <legend>Products</legend>
                    <li><a href="#1">Nokia</a></li>
                    <li><a href="#2">Sumsung</a></li>
                    <li><a href="#3">Sony Ericsson</a></li>
                    <li><a href="#4">Motorola</a></li>
                    <li><a href="#5">Apple</a></li>
                    <li><a href="#5">Seamen</a></li>
                </fieldset>
                <fieldset>
                    <legend>Online Support</legend>
```

```
                </fieldset>
            </div>
            <div class="middle_right"></div>
        </div>
        <div class="bottom"></div>
    </div>
</body>
</HTML>
```

Kết quả thực hiện



Hình 5.8: layout

Các bạn lưu ý các điểm sau:

Khoanh vùng liên kết cho 2 vùng khác nhau là .menu và .middle_left:

```
        /*--liên kết trong .menu cách nhau 20px--*/
        .menu a{padding: 0px 10px 0px 10px;}
        /*--liên kết trong .middle_left chữ HOA nhỏ--*/
        .middle_left a{font-variant:small-caps;}
        /*--liên kết có chuột trong .middle_left in đậm--*/
```

```
.middle_left a:hover{font-weight: bold;}
```

Chúng ta cũng khoanh vùng cho các <fieldset> và <li> đặt trong middle_left

```
/*--fieldset trong .middle_left cao tối thiểu 150--*/
.middle_left fieldset{min-height: 150px;}
/*--li trong .middle_left không dùng dấu, kẽ chân--*/
.middle_left li{list-style-type:none; border-bottom: 1px dotted red;}
```

2.4.2.7Qui tắc nạp chồng

Nếu 2 bộ chọn cùng được áp dụng cho cùng một thẻ thì phần khác nhau sẽ được gộp lại và phần giống nhau thì bộ chọn có độ ưu tiên cao hơn sẽ nạp chồng lên các bộ chọn có độ ưu tiên thấp hơn.

Các loại bộ chọn cũng như vị trí định nghĩa của chúng có ảnh hưởng đến độ ưu tiên áp dụng của các chúng. Hãy ghi nhớ 2 qui tắc sau:

- ✓ Khác phạm vi định nghĩa:
  - **Nội tuyến (thuộc tính style)->nhúng (thẻ <style>)->liên kết ngoài (tập tin css)**
- ✓ Cùng phạm vi định nghĩa:
  - **bộ chọn định danh->bộ chọn lớp->bộ chọn html**

Ví dụ sau thẻ <H1> chịu ảnh hưởng cả 3 bộ chọn khác nhau. Khi chạy sẽ cho dòng chữ "Nạp Chồng" màu vàng

```
<HTML>
<HEAD>
    <TITLE>Job application</TITLE>
    <style type="text/css">
        H1{color: green;}
        #xyz{color: yellow;}
        .abc{color: Red;}
    </style>
</HEAD>
<BODY>
<H1 class="abc" id="xyz">Nạp chồng</H1>
</BODY>
</HTML>
```

2.5 Các thuộc tính CSS

2.5.1Định dạng văn bản

Các thuộc tính CSS thường sử dụng để định nghĩa cho văn bản trên trang web như font chữ, màu sắc, chế độ hiển thị…

Thuộc tính	**Mô tả**
font-family: Verdana, Geneva, sans-serif;	Chỉ định tên font. Các font được liệt kê các nhau dấu phẩy. Áp dụng cho font được tìm thấy trước
font-size: 12px;	Kích thước font

font-style: italic;	Kiểu font
line-height: 12px;	Độ cao của mỗi hàng
font-weight: bold;	Độ đậm của font chữ: bold (đậm), 100 (độ đậm 100)
font-variant: small-caps;	Chữ hoa nhỏ, ký tự đầu lơn hơn
text-transform: uppercase;	Chữ hoa, chữ thường lowercase(thường), uppercase(HOA)
color: #F00;	Màu sắc có thể dùng mã (Red, Green, Blue) hoặc tên màu
text-decoration: none;	Trang trí chữ: none(không làm gì), underline(gạch chân), throuth-line(gạch ngang giữa)

2.5.2Nền

Nền chỉ có 2 loại là màu và ảnh. Nếu là ảnh thì cần điều chỉnh chế độ lặp lại (lát). Trong trường hợp không lặp bạn cần điều chỉnh vị trí đặt ảnh nền.

Thuộc tính	**Mô tả**
background-attachment: fixed;	Chế độ khi cuốn: fixed, scroll
background-color: #F00;	Màu nền
background-image: url(anh/abc.jpg);	Ảnh nền
background-repeat: repeat;	Chế độ lặp lại: none(không lặp), repeat(lặp cả 2 chiều), repeat-x(lặp chiều ngang), repeat-y(lặp chiều đứng)
background-position: left center;	Vị trí đặt ảnh nền trường hợp không lặp

2.5.3Định nghĩa toàn khối văn bản

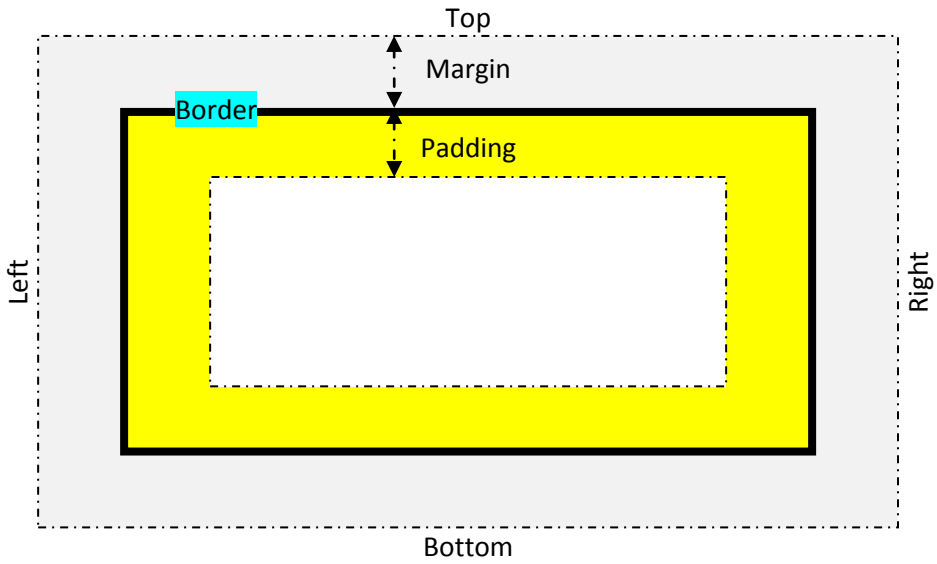
Điều khiển các thông số toàn khối văn bản là rất cần thiết để áp dụng cho một vùng nào đó trên trang web. Bạn có thể điều chỉnh khoản cách giữa các ký tự, các từ, thụt đầu dòng hay canh lề theo chiều đúng, chiều ngang. Đặc biết là chế độ hiển thị của khối.

Thuộc tính	**Mô tả**
letter-spacing: 2em;	Khoản cách giữa các ký tự
text-align: justify;	Canh lề: left, right, center, justify
text-indent: 5px;	Khoản thụt vào đầu dòng
vertical-align: middle;	Canh lề đứng: top, bottom, middle, base-line

word-spacing: 4em;	Khoản cách giữ các từ
display: inline;	Chế độ hiển thị: none (ẩn khối), block (một khối tách biệt), inline (trong hàng cho phép ngắt khối khi xuống hàng), inline-block (xuống hàng nguyên khối)

2.5.4Định nghĩa thuộc tính hộp

Mô hình của một hộp gồm kích thước, lề, phần đệm, được viềng, chế độ xếp hộp.



Hình 5.9: Mô hình hộp CSS

Thuộc tính	**Mô tả**
height: 222px;	Chiều cao
width: 111px;	Chiều rộng
margin: 6px;	Khoản lề xung quanh hộp. Sử dụng margin-top, margin-right, margin-bottom, margin-left nếu muốn định nghĩa riêng mỗi cạnh.
padding: 4px;	Phần đệm bên trong hộp. Sử dụng padding -top, padding -right, padding -bottom, padding -left nếu muốn định nghĩa riêng mỗi cạnh.
border: medium dotted #F00;	Đường kẻ theo thứ tự **độ dài, kiểu, màu**. Sử dụng border -top, border -right, border -bottom, border -left nếu muốn định nghĩa riêng mỗi cạnh.
float: left;	Gâm (chế độ xếp hộp) vào trái: left (gâm trái), right(gâm phải)
clear: right;	Hủy bỏ chế độ gâm: left(xóa gâm trái), right(xóa gâm phải), both(xóa gâm cả 2 bên)

2.5.5Định nghĩa danh sách

Để điều chỉnh <OL>, <UL> và <LI> bạn cận sử dụng các thuộc tính css sau đây.

Thuộc tính	**Mô tả**
list-style-position: inside;	Vị trí đặt dấu danh sách
list-style-image: url(xyz/abc.gif);	Hình ảnh thay dấu danh sách
list-style-type: square;	Kiểu dấu danh sách: square, none-list, circle

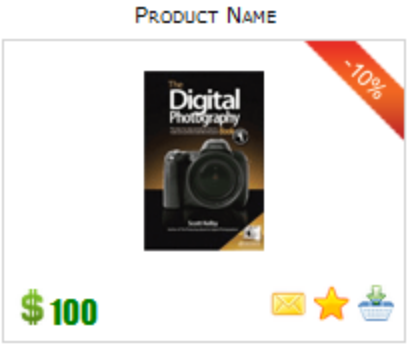
2.5.6Layer

Để tạo ra và điều chỉnh các thông số của nó, bạn cần học các thuộc tính css sau đây là đủ.

Thuộc tính	**Mô tả**
overflow: scroll;	Điều kiển chế độ tràn: scroll, visible, hidden
position: relative;	Chế độ vị trí của layer. Absolute (vị trí tuyệt đối so với layer mẹ), relative (vị trí tương đối tức đặt tại vị trí đặt thẻ)
visibility: visible;	Ẩn hiện layer
z-index: 111;	Chiều z hướng từ trong màn hình ra người dùng. Layer nào có z-index cao hơn sẽ nằm trên.
left: 0px;	Vị trí layer tính từ bên trái
top: 0px;	Vị trí layer tính từ bên trên
right: 0px;	Vị trí layer tính từ bên phải
bottom: 0px;	Vị trí layer tính từ bên dưới
clip: rect(0px,100px,100px,0px);	Vùng nhìn thấy trên layer

2.6 Bài tập và thực hành

1. Thiết kết layout để trình bày một hàng hóa như hình sau



Hình 5.10: Layout của một hàng hóa

Trong hình trên chúng ta đã có ảnh chính, tên hàng hóa, giá và các biểu tượng là dolar.png, letter.png, favorites.png, add.png và promo_icon.gif.

Để có được lay out như trên, bạn cần thực hiện theo các bước sau:

Bước 1: thiết kế HTML cho khối hàng hóa:

```
<div class="prod">
    <div class="name">Product Name</div>
    <div class="image">
        <a href="detail.jsp?id=1"><img src="thumb1.gif" /></a>
    </div>
    <div class="price"><img src="Dollar.png" />100</div>
    <div class="buttons">
        <a href="?send=1"><img src="Letter.png" /></a>
        <a href="?mark=1"><img src="Favourites.png" /></a>
        <a href="?add=1"><img src="Add.png" /></a>
    </div>
    <img class="icon" src="promo_icon.gif" />
</div>
```

Bước 2: thiết kế CSS cho các thành phần đã định sẵn trong HTML

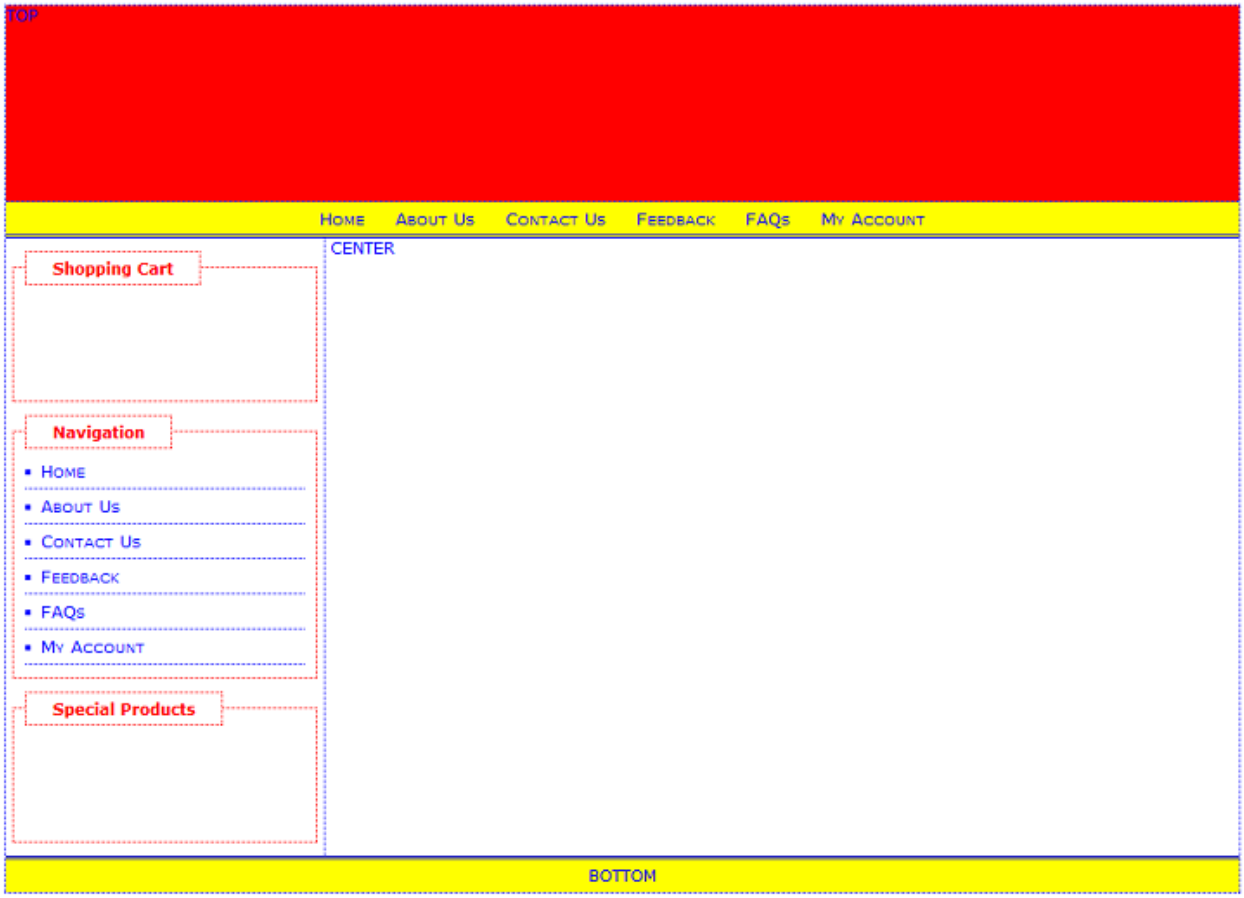
```
<style type="text/css">
.prod {
    width: 200px; height: 150px;
    border: 1px solid lightgrey;

    display: inline-block;
    margin: 30px 10px 0px 10px;
    position: relative;
```

```
}
.prod > * { /*--tất cả các thẻ con của .box--*/
    position:absolute;
    text-align:center;
}
.prod .icon {
    right:0px;
    top:0px;
}
.prod .name {
    top:-20px; /*--hiện ngoài .box--*/
    width:inherit; /*--cùng chiều rộng với .box--*/
    font-family: Verdana;
    font-size: 11px;
    font-variant:small-caps;
}
.prod .image {
    margin-top:15px;
    width:inherit;
}
.prod .price {
    left: 5px;
    bottom: 5px;
    font-family: Impact;
    color: Green;
}
.prod .buttons {
    right: 5px;
    bottom: 5px;
}
.prod .price img, .prod .buttons img {
    width: 18px;
    height: 18px;
}
```

```
</style>
```

2. Thiết kế layout như sau:



Hình 5.11: Layout

Bước 1: Tạo trang MyLayout.html có nội dung như sau

```
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>CSS Lab</title>
    <link href="MyLayout.css" rel="stylesheet" type="text/css" />
</head>

<body>
      <div id="container">
      <div id="top">TOP</div>
        <div id="menu">
```

```
            <ul>
                <li><a href="?#1">Home</a></li>
                <li><a href="?#2">About Us</a></li>
                <li><a href="?#3">Contact Us</a></li>
                <li><a href="?#4">Feedback</a></li>
                <li><a href="?#5">FAQs</a></li>
                <li><a href="?#6">My Account</a></li>
            </ul>
        </div> <!--end menu-->
        <div id="middle">
            <div id="middle-left">
                <fieldset>
                  <legend>Shopping Cart</legend>
                </fieldset>
                <fieldset>
                  <legend>Navigation</legend>
                    <ul>
                        <li><a href="?#1">Home</a></li>
                        <li><a href="?#2">About Us</a></li>
                        <li><a href="?#3">Contact Us</a></li>
                        <li><a href="?#4">Feedback</a></li>
                        <li><a href="?#5">FAQs</a></li>
                        <li><a href="?#6">My Account</a></li>
                    </ul>
                </fieldset>
                <fieldset>
                  <legend>Special Products</legend>
                </fieldset>
            </div> <!--end middle-left-->
            <div id="middle-center">CENTER</div>
        </div>
        <div id="bottom">BOTTOM</div>
    </div>
</body>
```

```
</html>
```

Bước 2: Tạo tập tin MyLayout.css định nghĩa CSS cho các vùng trong layout.

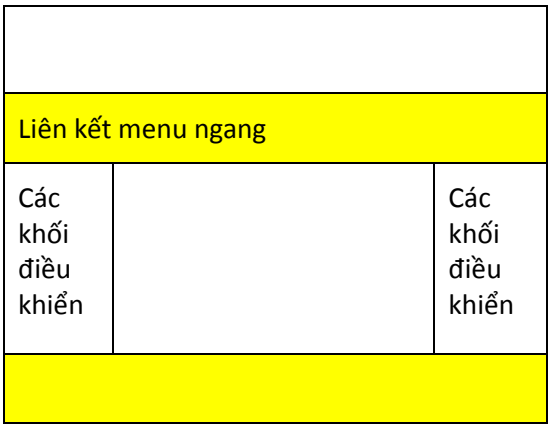
```
/*--định nghĩ chung cho toàn trang--*/
body,td,th {
	font-family: Verdana, Geneva, sans-serif;
	font-size: 12px;
	color: blue;
}
body {
	background-color: White;
	background-image: url(abc.gif);
	background-repeat: repeat;
	background-attachment:fixed;
}
a{
	text-decoration: none;
	font-variant:small-caps;
	font-size: 13px;
}
a:link, a:active, a:visited {
	color: blue;

}
a:hover {
	color: red;
}
h1, h2, h3 {
	color: blue;
	padding: 2px;
	margin: 0px;
}
/*--định nghĩa css cho các vùng của layout--*/
#container {
	margin: 0px auto;
	width: 950px;
	border: 1px dotted blue;
}
#top {
	height: 150px;
	background-color: red;
	border-bottom: 1px dotted blue;
}
#menu {
	height: 20px;
	background-color: yellow;
	text-align: center;
	padding-top: 5px;
	border-bottom: medium double blue;
```

```
}
#middle > div {
	min-height: 400px;
}
#middle-left {
	width: 245px;
	float:left;
	border-right: 1px dotted blue;
}
#middle-center {
	width: 700px;
	float:right;
}
#bottom {
	height: 20px;
	background-color: yellow;
	text-align: center;
	padding-top: 5px;
	font-variant:small-caps;
	border-top: medium double blue;
	clear:both;
}
/*--định nghĩa ul, li trong #menu ngang--*/
#menu ul{
	margin: 0px;
	padding: 0px;
}
#menu ul li{
	display: inline-block;
	padding: 0px 10px 0px 10px;
}
/*--định nghĩa legend, fieldset, ul, li trong #middle-left--*/
#middle-left fieldset{
	margin: 10px 5px 10px 5px;
	border: 1px dotted red;
	padding-bottom: 10px;
	min-height: 100px;
}
#middle-left legend{
	padding: 5px 20px 5px 20px;
	border: 1px dotted red;
	font-weight: bold;
	color: red;
}
#middle-left ul{
	margin: 0px;
	padding: 0px;
	list-style:inside;
	list-style-type:square;
```

```
}
#middle-left ul li{
	padding: 5px 0px 5px 0px;
	border-bottom: 1px dotted blue;
}
#middle-left li:hover{
	background-color: yellow;
	border-bottom: medium double red;
}
```

3. Hãy thiết kế layout có thiết kế sau



Hình 5.12: Mô hình layout

4. Hãy thiết kế CSS áp dụng cho tất cả các thẻ <input type='text|password'> hay textarea theo yêu cầu sau:
   - ✓ Đường viền xanh, đứt nét, dày 1px
   - ✓ Ảnh nền không lặp lại hiện ngay bên phải của ô nhập
   - ✓ Rộng 100%

3 JQUERY

- ✓ Giới thiệu về JQuery
- ✓ JQuery Selector
- ✓ JQuery Filter
- ✓ Thay đổi nội dung document
- ✓ Xử lý sự kiện
- ✓ Kiểm tra hợp lệ
- ✓ Ajax
- ✓ Giao diện
- ✓ Hiệu ứng & hoạt ảnh

3.1 Giới thiệu về JQuery

Jquery là thư viện javascript mã nguồn mở, miễn phí được sử dụng rộng rãi trên toàn cầu để tăng cường xử lý web phía client và khả năng tương tác với server thông qua ajax. Khi sử dụng Jquery bạn có thể yên tâm về sự tương thích với trình duyệt và tính chuẩn hóa mã JavaScript điều này tạo cơ hội hội nhập với cộng đồng lập trình viên quốc tế.

- ✓ Jquery truy xuất các thành phần nội dung trang web với cú pháp tương tự css (thông qua các bộ chọn selector). Dù sao với Jquery bộ chọn được tăng cường để dễ dàng tìm kiếm các thẻ trong việc xử lý chúng.
- ✓ Hỗ trợ nhiều thao tác xử lý trên tập các element chỉ bằng một dòng lệnh. Để làm được điều này, Jquery định nghĩa một đối tượng mới gọi là Jquery và gần như kết quả xử lý của mọi hàm trong Jquery đều trả lại chính đối tượng Jquery hiện tại nên mới có chuỗi lời gọi $("selector").func1().func2().func3()…;
- ✓ Đơn giản hóa cách viết mã nguồn javascript (write less, do more). Tách biệt mã xử lý javascript và thành phần thể hiện HTML.

✪ Khởi động nhanh với Jquery

Để dễ hiểu Jquery, chúng ta cùng nhau xem qua một ví dụ về Jquery. Qua đó chúng ta làm quen với một vài khái niệm quan trọng khi làm việc với Jquery.

Ví dụ Jquery đơn giản

```
<html>
<head>

<!--Nạp thư viện JQuery-->
<script src="Scripts/jquery-1.4.1.min.js" type="text/javascript"></script>

<script type="text/javascript">

    /*sau khi trang web được tải thành công thì sự kiện ready sẽ xảy ra*/
    $(document).ready(function () {

        /*đăng ký sự kiện click cho tất cả các thẻ input có type='button'*/
        $(":button").click(function () {

            /*ẩn nút bị click vào kéo dài 1 giây--*/
```

```
            $(this).hide(1000, function () {

                /*hộp thoại thông báo sẽ được hiện ra sau khi nút được ẩn*/
                alert("Nút đã được ẩn");
            });

        });

    });
</script>
</head>
<body>
    JQuery demo
    <input type="button" value="Button 1" />
    <input type="button" value="Button 2" />
</body>
</html>
```

Phân tích ví dụ trên:

✪ Nạp thư viện: Để làm việc với Jquery, trước hết bạn cần nạp thư viện Jquery.

```
<script src="Scripts/jquery-1.4.1.min.js" type="text/javascript"></script>
```

Bạn download Jquery tại http://jquery.com để sử dụng phiên bản mới nhất được cung cấp miễn phí bởi cộng đồng phát triển Jquery. Nếu làm việc với các chức năng mở rộng chưa được cung cấp bởi Jquery bạn phải download thêm các plugin của nó. Hiện tại chúng ta chưa cần đến, lúc cần chúng tôi sẽ hướng dẫn để bạn download đúng plugin cần.

✪ Hoạt động của đoạn mã lệnh

Sau khi thư viện đã được nạp, bạn có thể viết mã Jquery. Vì Jquery là JavaScript nên mã của nó cũng được đặt giữ cặp thẻ <script> và </script>.

Sự kiện `$(document).ready(function (){…})` xảy ra ngay sau khi trang web được tải xong, nghĩa là sau khi trang web được tải thành công thì nội dung của hàm truyền vào ready() sẽ được thực hiện.

Lợi dụng vào điều này chúng ta đăng ký xử lý sự kiến click cho các thẻ <input type='button'> với câu lệnh `$(":button").click(function () {…});` Vì vậy khi click vào các nút thì đoạn mã lệnh sau sẽ chạy `$(this).hide(1000, function () {…});` Đoạn mã này có tác dụng làm ấn nút vừa được click vào với thời gian ẩn kéo dài 1000 mili giây (hay 1 giây). Sau khi ẩn hoàn toàn hàm truyền ở tham số thứ 2 của hide() sẽ thực hiện và vì vậy ta nhận được hộp thoại thông báo "Nút đã được ẩn".

✪ Các khái niệm
- $(document), $(":button"), $(this): được gọi là bộ chọn (selector). Như vậy bộ chọn dùng để chọn đối tượng cần xử lý.
- Ready(), click(): được gọi là sự kiện, được kích hoạt tự động do hệ thống hoặc con người tương tác vào.
- Hide(): phương thức của JQuery dùng để xử lý hiệu ứng ẩn
- 1000 và các function là các tham số cho các sự kiện được gọi là handler. Riêng hàm làm tham số cho phương thức hide() gọi là hàm gọi ngược (hay callback) vì hàm này được chạy sau sự chờ đợi kết thúc của một hành động khác.

✪ Các thành phần trong JQuery

- Core functionality: các phương thức core của JQuery và các hàm tiện ích được sử dụng thường xuyên.
- Selector & Traveral: chọn, tìm kiếm element, duyệt qua các element trong document.
- Manipulation & CSS: thay đổi nội dung các element trong document, làm việc với css.
- Event: đơn giản hóa việc xử lý event. Cung cấp event helper function đăng ký nhanh các event.
- Effect & Animation: cung cấp các hàm hỗ trợ tạo animation & effect.
- Ajax: sử dụng để tương tác với server
- User interface: tập widget với các control: accordion, datepicker, dialog, progressbar, slider, tab
- Extensibility: hỗ trợ tạo plugin bổ sung thêm các chức năng mới vào core library.

3.2 Bộ chọn (JQuery Selector)

Như vậy vấn đề quan trọng bậc nhất khi học Jquery là bộ chọn (selector). Để xử lý các thẻ bạn cần phải biết cách xác định nó. Bộ chọn gồm 2 phần là bộ chọn cơ bản và bộ lọc.

- ✓ Truy xuất nội dung (element) trong document dựa trên biểu thức selector cung cấp. Selector sử dụng cú pháp tương tự CSS.
- ✓ Tập kết quả do Selector và Filter trả về: Jquery objects (không phải DOM objects).
- ✓ Cú pháp và cách chọn tương tự CSS

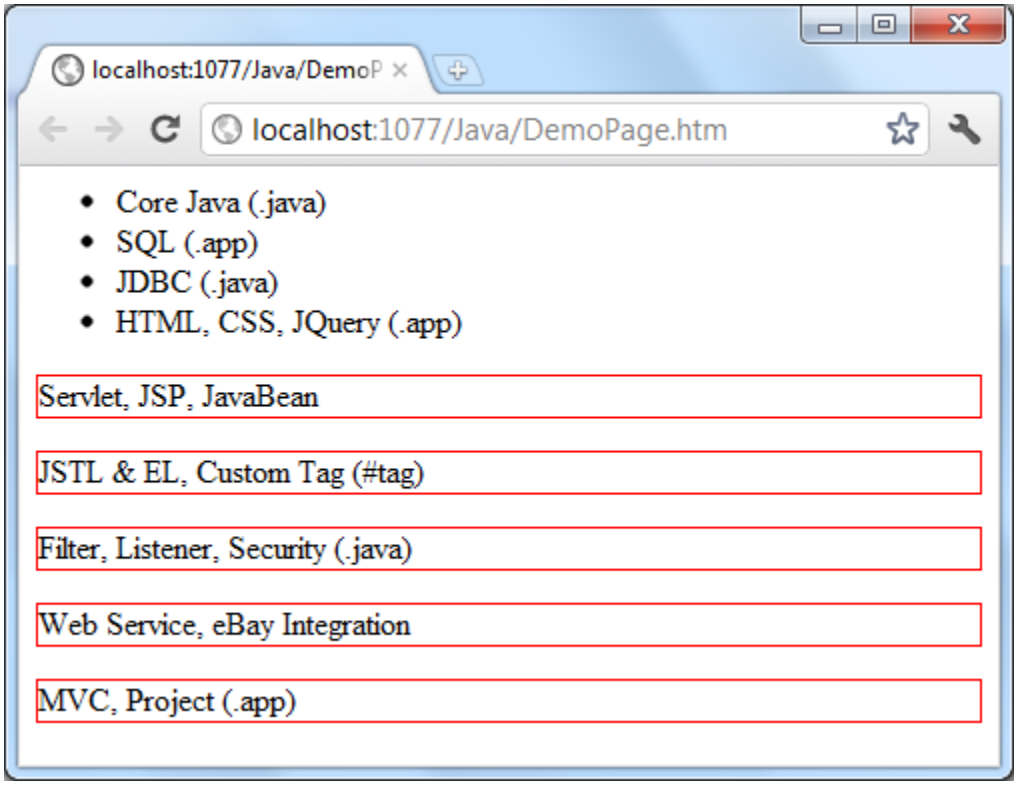
3.2.1Các bộ chọn cơ sở

SELECTOR	VÍ DỤ	Ý NGHĨA
<tên thẻ>	$(“p”)	Chọn tất cả các thẻ có tên là <tên thẻ>
#<định danh>	$(“#basic”)	Chọn tất cả các thẻ có ID là <định danh>
.<tên lớp>	$(“.java”)	Chọn tất cả các thẻ với thuộc tính class có giá trị là <tên lớp>
<tên thẻ>.<tên lớp>	$(“li.app”)	Chọn tất cả các thẻ có tên thẻ là <tên thẻ> với thuộc tính class có giá trị là <tên lớp>
*	$(“*”)	Chọn tất cả các element trên document.

Ví dụ bọc đường bao cho tất cả các thẻ <p>

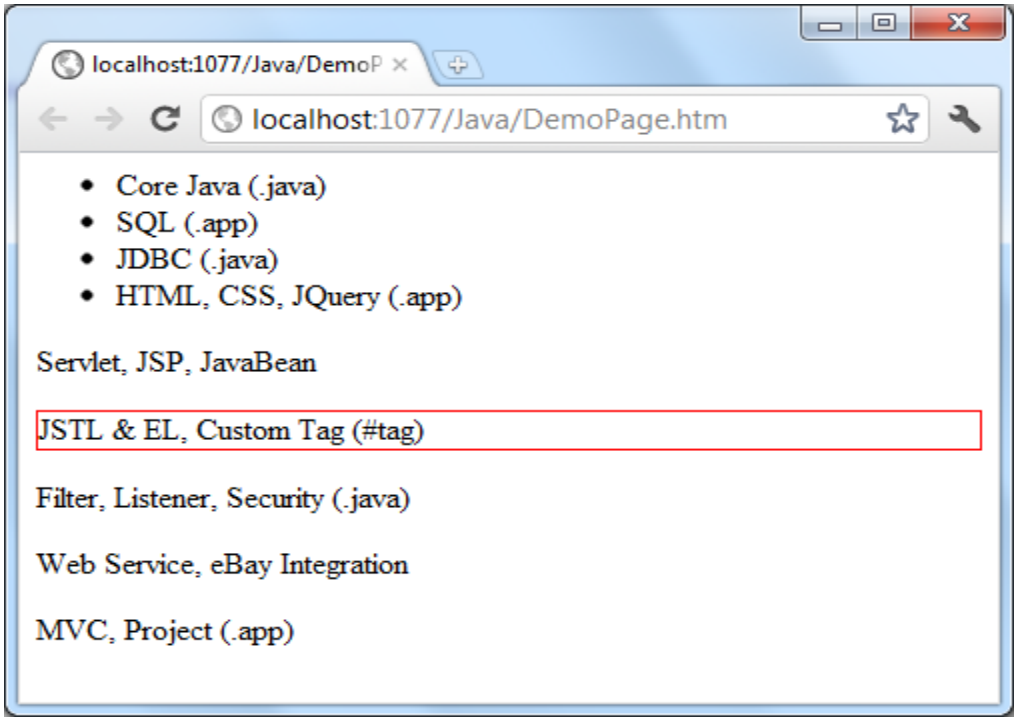
```
<html>
<head>
    <script                                    src="Scripts/jquery-1.4.1.min.js"
type="text/javascript"></script>
<script type="text/javascript">
    $(document).ready(function () {
        /*kẻ được bao (1px liền nét màu đỏ) cho tất cả thẻ <p>*/
        $("p").css("border", "1px solid red");
```

```
    });
</script>
</head>
<body>
    <ul id="basic">
        <li class="java">Core Java (.java)</li>
        <li class="app">SQL (.app)</li>
        <li class="java">JDBC (.java)</li>
        <li class="app">HTML, CSS, JQuery (.app)</li>
    </ul>
    <p>Servlet, JSP, JavaBean</p>
    <p id="tag">JSTL & EL, Custom Tag (#tag)</p>
    <p class="java">Filter, Listener, Security (.java)</p>
    <p>Web Service, eBay Integration</p>
    <p class="app">MVC, Project (.app)</p>
</body>
</html>
```



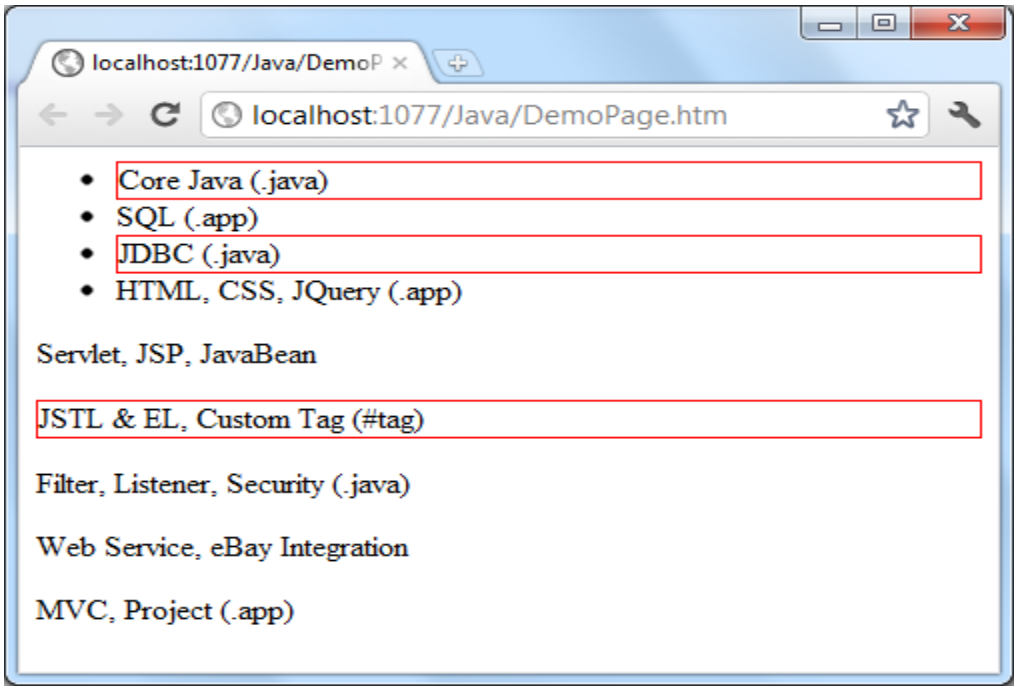
Hình 6.1: thẻ <p>

Ví dụ chỉ kẻ đường bao cho các thẻ có id là "tag". Mã nguồn như trên chỉ khác phần phần bộ chọn thẻ là `$("#tag")`.



Hình 6.2: Chọn thẻ có id="tag"

Ví dụ: `$("li.java, #tag")` – xác định tất cả các thẻ <li class='java'> và các thẻ có id='#tag'



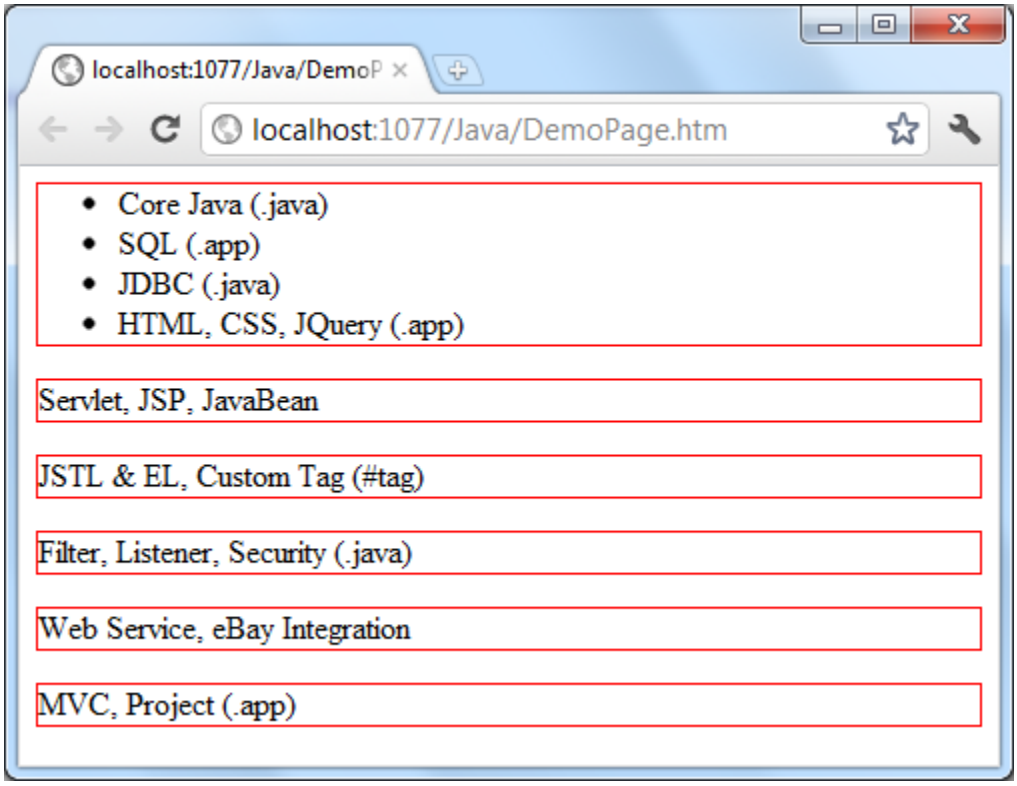
Hình 6.3: Chọn thẻ <li class="java"> hoặc thẻ có id="tag"

3.2.2Các bộ chọn quan hệ phân cấp

Chọn element dựa trên mối quan hệ phân cấp giữa các thẻ. Mối quan he

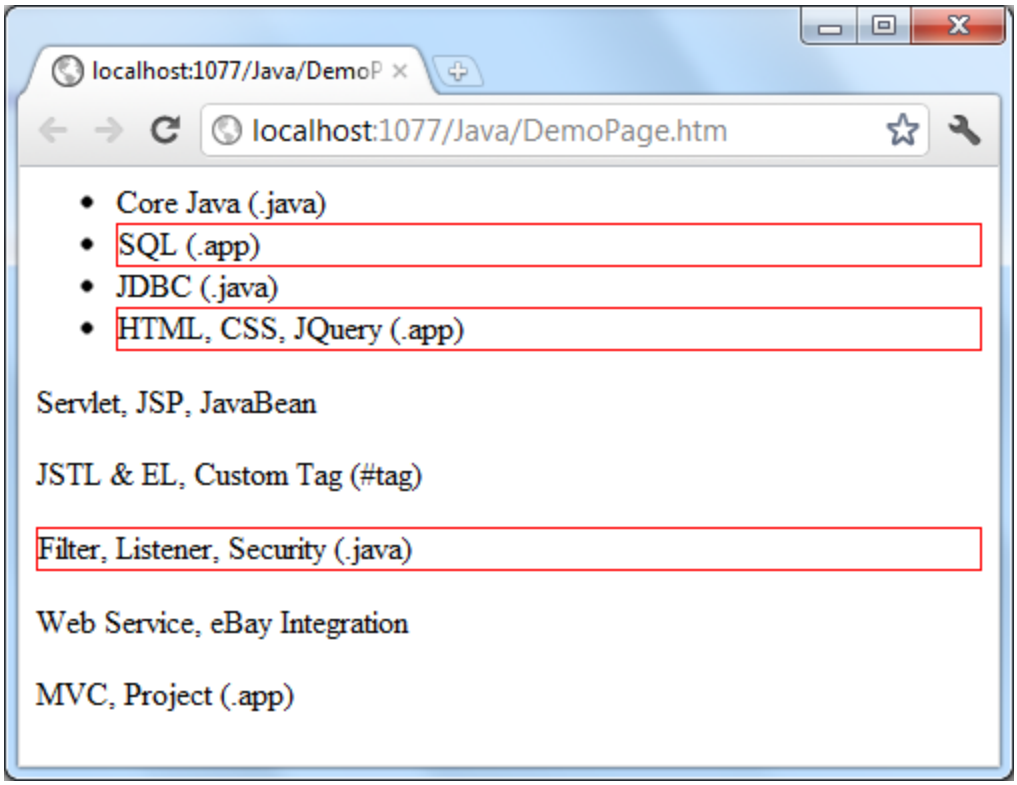
SELECTOR	VÍ DỤ	Ý NGHĨA
Selector1,selector2…	$(“#basic,.java”)	Chọn tất cả các thẻ được xác định bởi tất cả các bộ chọn
Parent > Child	$(“body>.app”)	Chọn tất cả các thẻ con trực tiếp của các thẻ thuộc bộ chọn <Parent>
Ancestor Descendant	$(“body .java”)	Chọn tất cả các thẻ con, cháu của các thẻ thuộc bộ chọn <Ancestor>
Prev + Next	$(“.java+p”)	Chọn một thẻ nằm kế tiếp của cả các thẻ thuộc bộ chọn <Prev>
Prev ~ Siblings	$(“p.java~p”)	Chọn tất cả các thẻ em của các thẻ thuộc bộ chọn <Prev>

Ví dụ: `$("body > *")` – xác định tất cả các thẻ nằm trong thẻ <body>



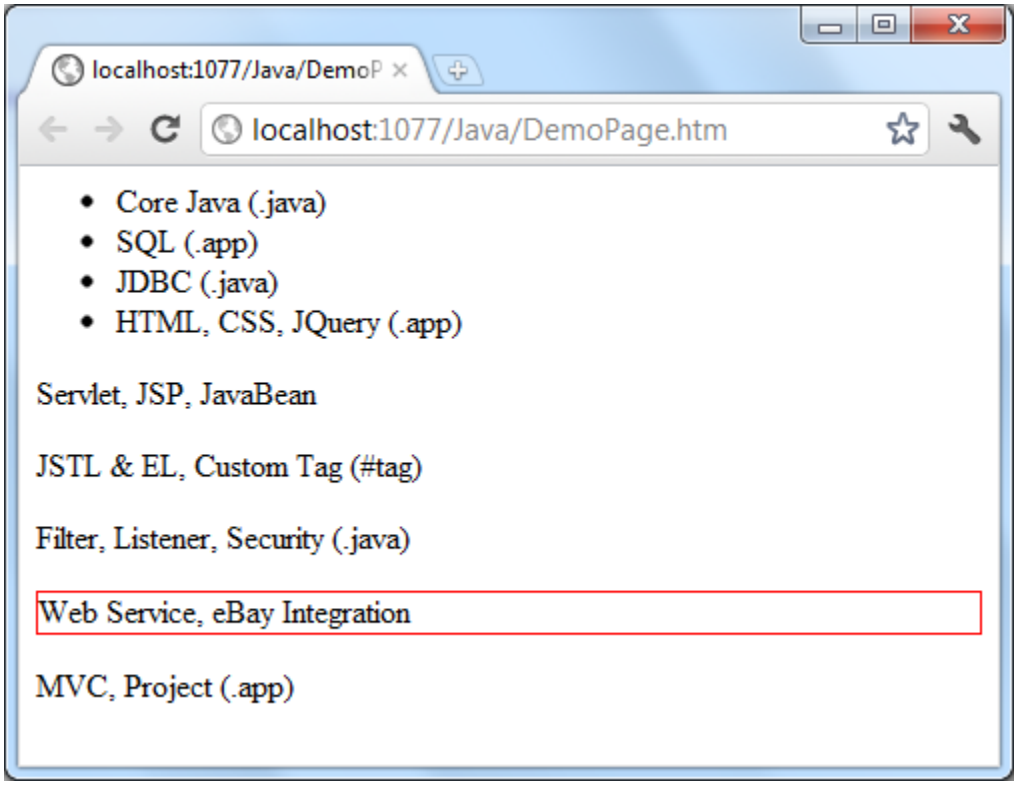
Hình 6.4: Chọn tất cả các con của <body>

Ví dụ: `$("body>.java, #basic .app")` – xác định tất cả các thẻ <p class=’java’> trên toàn bộ trang và các thẻ bất kỳ có class=’app’ nằm trong các thẻ có id=’#basic’



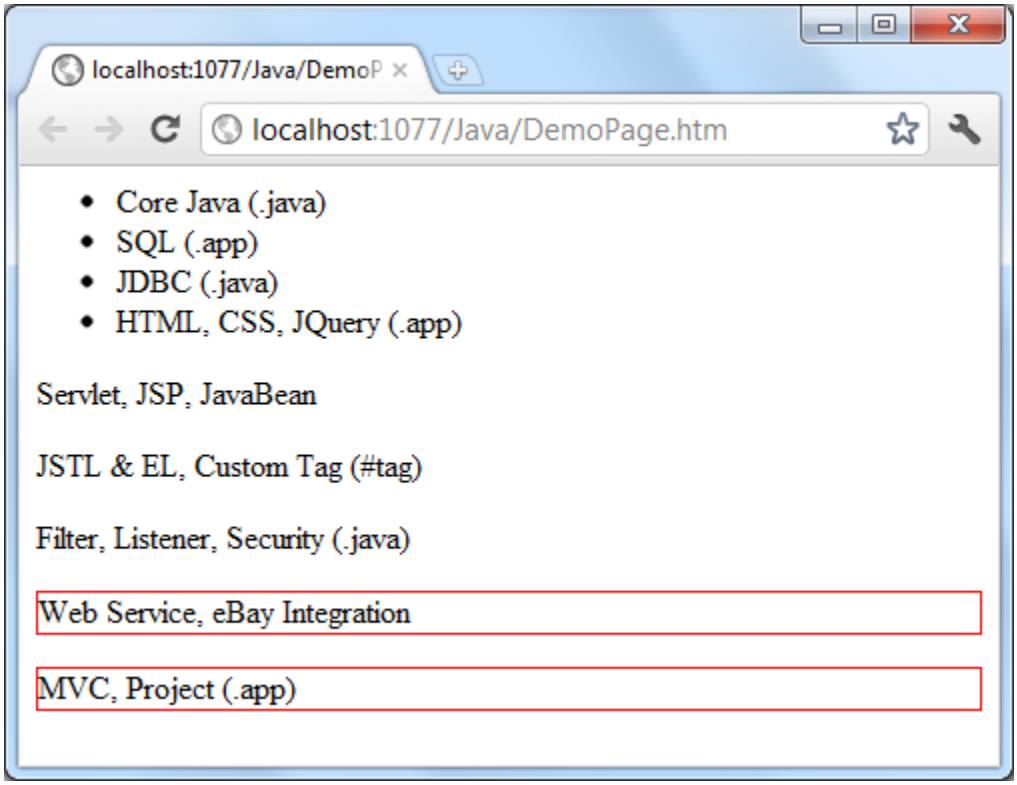
Hình 6.4: chọn các con của <body> có class=”java” và các thẻ có class=”app” đặt trong thẻ có id=”basic”

Ví dụ: `$("p.java+p")` – xác định các thẻ p nằm ngay sau thẻ <p class=’.java’>



Hình 6.5: chọn thẻ p nằm ngay sau thẻ <p class='.java'>

Ví dụ: `$("p.java~p")` – xác định các thẻ <p> nằm sau các thẻ <p class='java'>



Hình 6.6: chọn các thẻ p nằm sau thẻ <p class='.java'>

3.2.3Form Selector

SELECTOR	Ý NGHĨA
:input	Chọn tất cả thẻ input, textarea trên Form
:text	Chọn tất cả text field trên Form
:password	Chọn tất cả password field
:radio	Chọn tất cả radio button
:checkbox	Chọn tất cả checkbox
:submit	Chọn tất cả button submit
:reset	Chọn tất cả button reset
:image	Chọn tất cả image
:button	Chọn tất cả generalized button

:file	Chọn tất cả control upload file

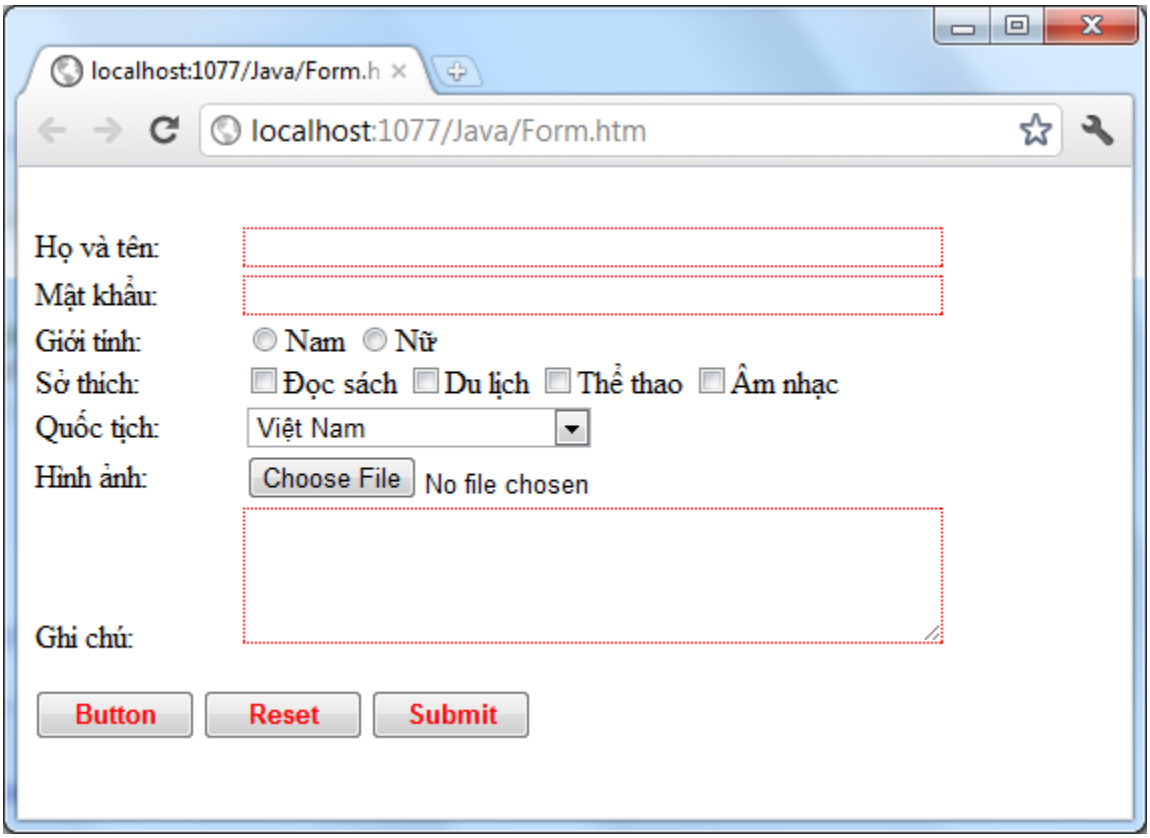
Ví dụ: đặt độ rộng và kẻ đường viền cho :text, :password và textarea đồng thời đặt kích thước, màu và độ đậm cho các nút.

```
<html>
<head>
<style type="text/css">
    li{display: inline-block;width: 100px;}
</style>
<script src="Scripts/jquery-1.4.1.min.js" type="text/javascript"></script>
<script type="text/javascript">
    /*sau khi trang web được tải thành công thì sự kiện ready sẽ xảy ra*/
    $(document).ready(function () {

        /*đặt độ rộng và kẻ đường viền cho :text, :password và textarea*/
        $(":text, :password, textarea").css(
                        { "width": "350px", "border": "1px dotted red" });
        $(":button, :reset, :submit").css({ "width": "80px", "height":
                        "25px", "color": "red", "font-weight": "bold" });

    });
</script>
</head>
<body>
<form         id="frmDangKy"         action="DangKy.do"         method="post"
enctype="multipart/form-data">
    <br/><li>Họ và tên:</li>
    <input id="txtHoTen" name="txtHoTen" type="text" />
    <br/><li>Mật khẩu:</li>
    <input id="txtMatKhau" name="txtMatKhau" type="password" />
    <br/><li>Giới tính:</li>
    <input id="NA" name="rdoGioiTinh" type="radio" value="1" />Nam
    <input id="NU" name="rdoGioiTinh" type="radio" value="0" />Nữ
    <br/><li>Sở thích:</li>
    <input id="DS" name="chkSoThich" type="checkbox" />Đọc sách
    <input id="DL" name="chkSoThich" type="checkbox" />Du lịch
    <input id="TT" name="chkSoThich" type="checkbox" />Thể thao
    <input id="AN" name="chkSoThich" type="checkbox" />Âm nhạc
    <br/><li>Quốc tịch:</li>
    <select id="cboQuocTich" name="cboQuocTich">
        <option value="VN">Việt Nam</option>
        <option value="US">United States of America</option>
    </select>
    <br/><li>Hình ảnh:</li>
    <input id="filHinh" name="filHinh" type="file" />
    <br/><li>Ghi chú:</li>
    <textarea id="txtGhiChu" name="txtGhiChu" rows="4"></textarea>
    <p>
```

```
    <input id="btnButton" name="btnButton" type="button" value="Button" />
    <input id="btnReset" name="btnReset" type="reset" value="Reset" />
    <input id="btnSubmit" name="btnSubmit" type="submit" value="Submit" />
</form>
</body>
</html>
```



Hình 6.7: Chọn các thẻ trên form

3.3 Bộ lọc (JQuery Filter)

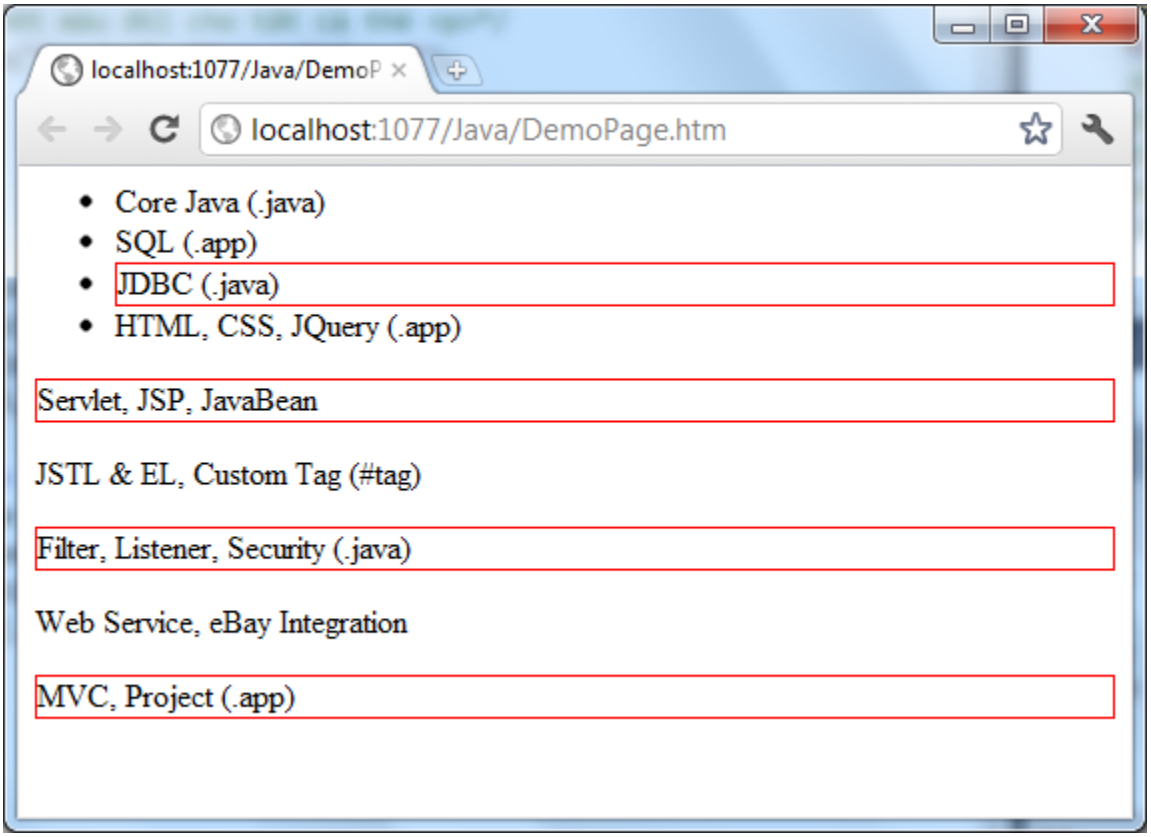
JQuery Selector thường trả về 1 tập đối tượng. JQuery Filter được dùng để lọc trên kết quả chọn của JQuery Selector. Jquery cung cấp các bộ lọc sau đây hỗ trợ xây dựng điều kiện tìm kiếm các thẻ chính xác và nhanh hơn. Có 6 loại bộ lọc:

- ✓ Basic: lọc phần tử ở vị trí đầu tiên, cuối cùng, chẵn, lẻ,
- ✓ Content: lọc dựa trên nội dung
- ✓ Visibility: lọc dựa trên trạng thái hiển thị của element
- ✓ Attribute: lọc dựa trên thuộc tính của element
- ✓ Child: lọc dựa trên mối quan hệ với element cha
- ✓ Form: lọc trên các thành phần khai báo trên Form

3.3.1Bộ lọc cơ bản

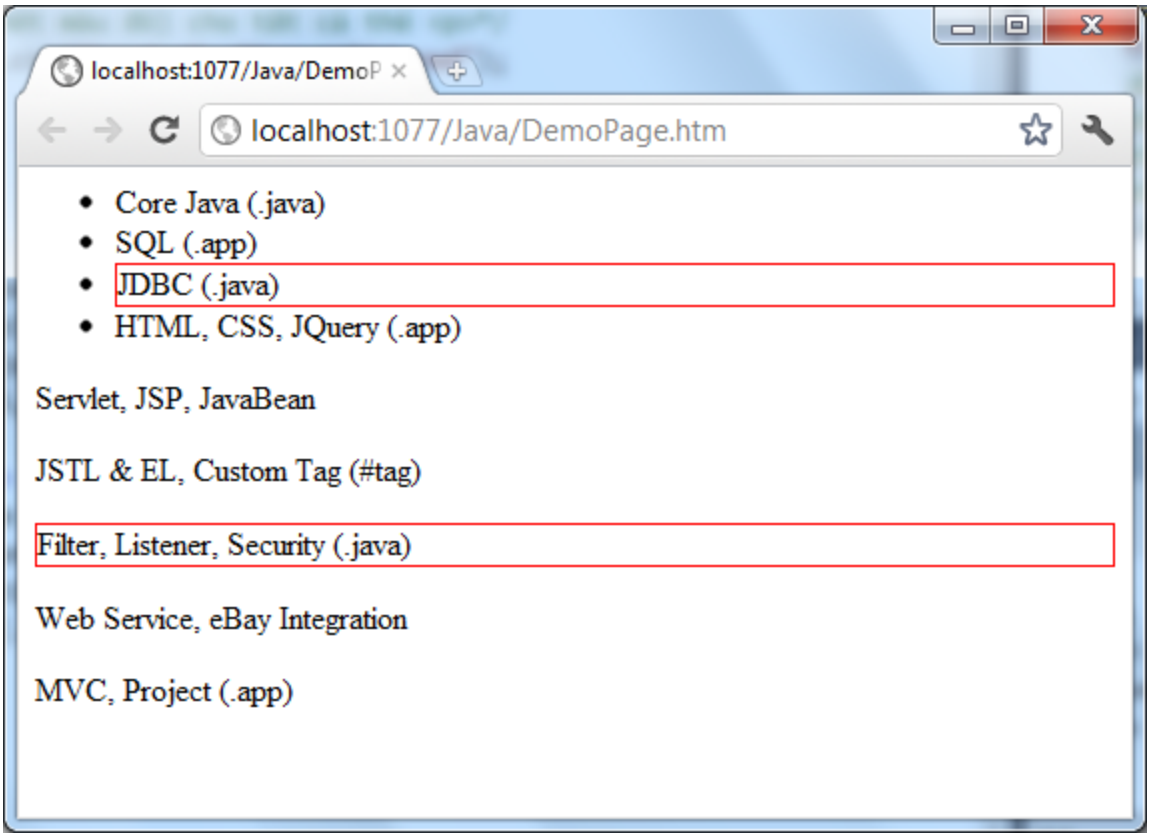
FILTER	Ý NGHĨA
:first	Chọn phần tử đầu tiên trong tập kết quả do Selector trả về
:last	Chọn phần tử cuối cùng trong tập kết quả do Selector trả về
:even	Chọn phần tử chẵn
:odd	Chọn phần tử lẻ
:eq (index)	Chọn phần tử tại vị trí index
:gt (index)	Chọn phần tử có vị trí > index
:lt (index)	Chọn phần tử có vị trí < index
:header	Chọn tất cả header element (H1, H2, .. H6)
:not (selector)	Chọn phần tử không thỏa selector

Ví dụ: `$("p:even, li:eq(2)")`- Xác định các thẻ <p> chẵn và <li> thứ 3



Hình 6.8: Chọn các thẻ <p> chẵn và <li> thứ 3

Ví dụ `$(".java:not(li:eq(0))")` – xác định các thẻ có class='java' nhưng loại thẻ <li class='java'> thứ nhất.



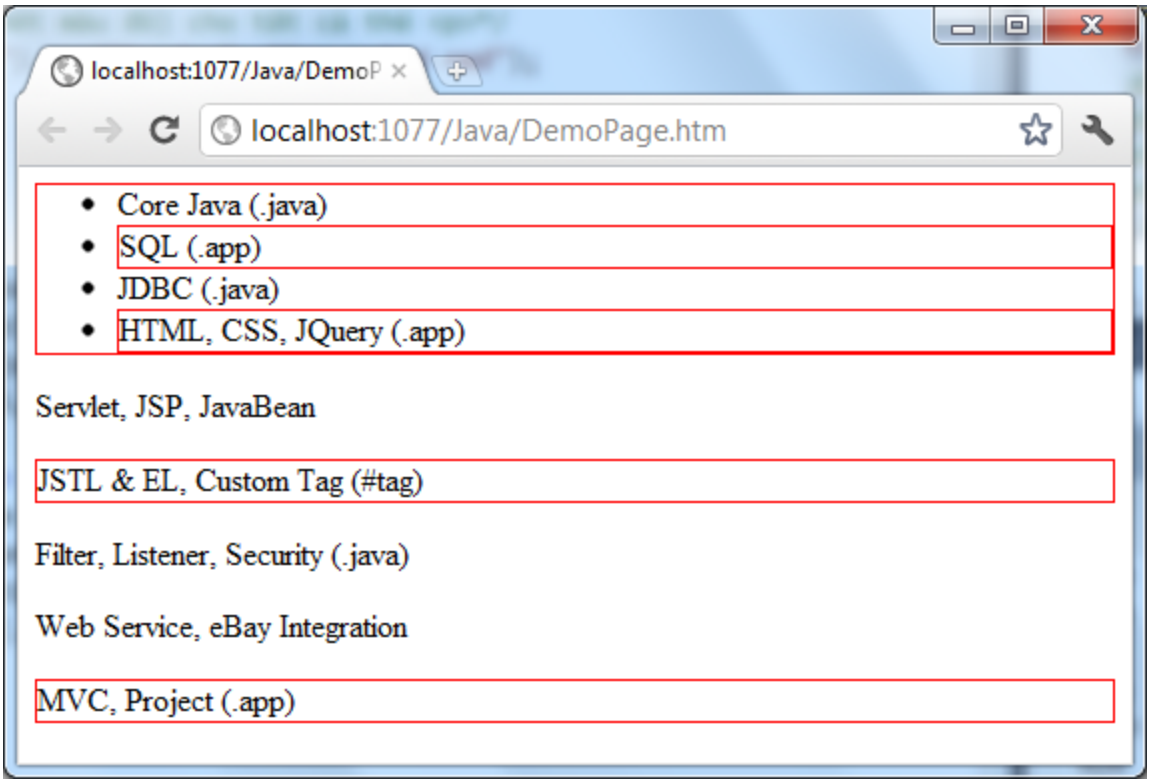
Hình 6.9: chọn các thẻ có class='java' nhưng loại thẻ <li class='java'>

3.3.2Bộ lọc thuộc tính

Lọc các thẻ dựa vào điều kiện liên quan đến thuộc tính

FILTER	Ý NGHĨA
[attribute]	Lọc các phần tử có khai báo attribute
[attribute=value]	Lọc các phần tử có attribute với giá trị = value
[attribute!=value]	Lọc các phần tử có attribute với giá trị != value
[attribute^=value]	Lọc các phần tử có attribute với giá trị bắt đầu là value
[attribute$=value]	Lọc các phần tử có attribute với giá trị kết thúc là value
[attribute*=value]	Lọc các phần tử có attribute chứa giá trị value
[attributeFilter1] [attributeFilter2]…	Lọc các phần tử thỏa tất cả các attribute filter.

Ví dụ: `$("*[id], *[class*='pp']")` – xác định tất cả các thẻ có chứa thuộc tính id hoặc thuộc tính class có chứa chuỗi "pp"



Hình 6.10: Chọn các thẻ có thuộc tính id hoặc thuộc tính class e chứa chuỗi "pp"

3.3.3Bộ lọc nội dung

Lọc các thẻ dựa vào điều kiện liên quan đến nội dung.

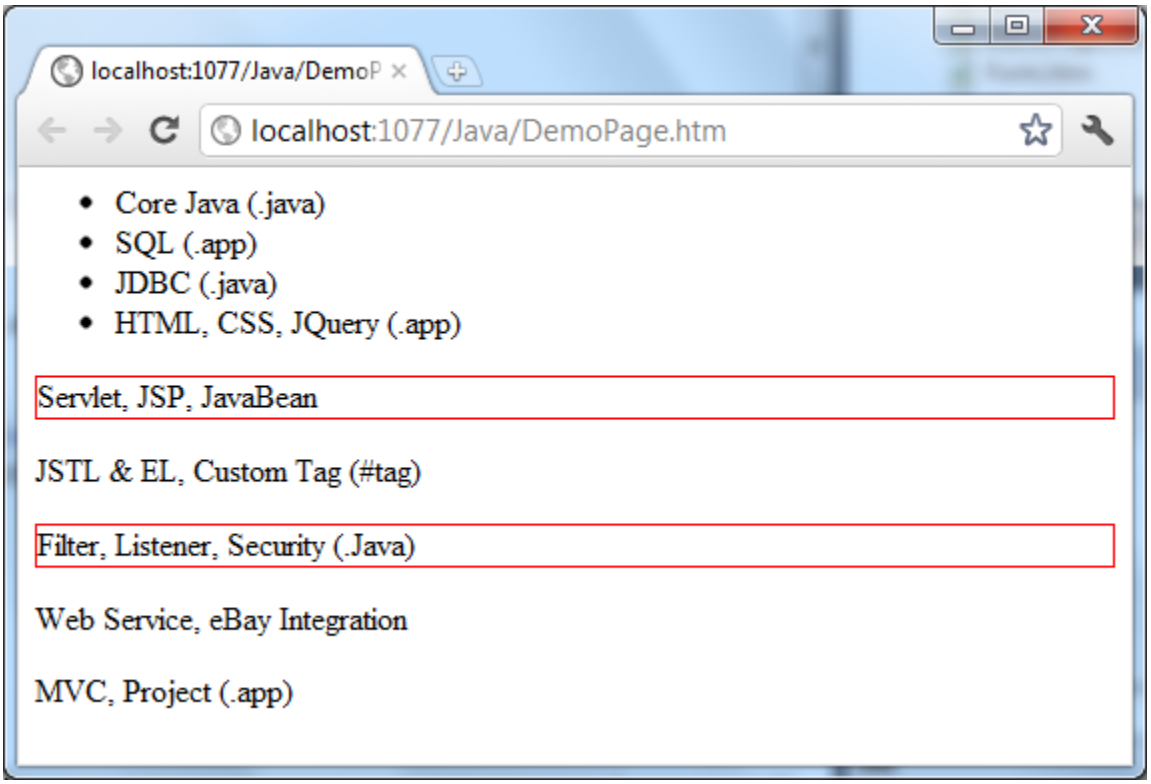
FILTER	Ý NGHĨA
:contains(text)	Lọc các phần tử có chứa chuỗi text
:empty	Lọc các phần tử rỗng
:has(selector)	Lọc các phần tử có chứa ít nhất 1 element thỏa selector
:parent	Lọc các phần tử có ít nhất 1 con hoặc text (Lấy các thẻ không rỗng)

3.3.4Bộ lọc tính khả kiến

Lọc các thẻ dựa vào điều kiện liên quan đến tính đang ẩn hay hiển thị.

FILTER	Ý NGHĨA
:visible	Lọc các phần tử có trạng thái là visible (đang hiển thị)
:hidden	Lọc các phần tử có trạng thái hidden (đang ẩn)

Ví dụ: `$(".app:empty, p:contains('Java')")` – xác định các thẻ rỗng có thuộc tính class='app' và các thẻ <p> có nội dung chứa chuỗi "Java"



Hình 6.11: Chọn các thẻ rỗng có thuộc tính class='app' và các thẻ <p> có nội dung chứa chuỗi "Java"

3.3.5Bộ lọc con cháu

FILTER	Ý NGHĨA
:nth-child(index) :nth-child(even) :nth-child(odd)	Lọc các phần tử theo vị trí so với cha của nó
:nth-child(equation)	Lọc phần tử theo vị trí (vị trí thỏa phương trình tham số) so với cha của nó
:first-child	Lấy phần tử đầu tiên so với cha của nó
:last-child	Lấy phần tử cuối cùng so với cha của nó
:only-child	Lấy phần tử nếu phần tử này là con duy nhất so với cha của nó

Ví dụ:

$("li:nth-child(even)") - xác định các thẻ <li> con tại vị trí chẵn

$("li:nth-child(even)") - xác định thẻ <li> là con duy nhất

$("li:nth-child(2)") - xác định các thẻ <li> con thứ 3

$("li:nth-child(3n+1)") – xác định các thẻ <li> thứ 1, 4, 7, 10…

3.3.6Xử lý tập kết quả

METHOD	Ý NGHĨA
size()	Lấy số phần tử trong tập kết quả của Selector
get()	Lấy tập DOM elements trong tập kết quả của Selector
get(index)	Lấy DOM element ở vị trí index
find(expression)	Lấy các element con cháu thỏa expression
each()	Gọi thực thi phương thức với từng element trong tập kết quả của Selector

Ví dụ:

- ✓ $("li.java").size() – cho số phần tử <li class='java'>
- ✓ $("li").get(2) – cho phần tử <li> thứ 3
- ✓ $("ul").find("li.app") – cho các <li class='app'> trong
- ✓ $(".java").each(function(){alert($(this).html())}) – thông báo nội dung các thẻ có class='java'

3.4 Thao tác dữ liệu trang web

Khi lập trình JavaScript nói chung và Jquery nói riêng, chúng ta thường thao tác các thẻ. Vì vậy dữ liệu mà chúng ta thường xuyên làm việc là:

- ✓ Nội dung thẻ (phần giữa thẻ) gồm HTML và thuần văn bản (text)
- ✓ Thuộc tính của thẻ
- ✓ Các thuộc tính CSS

3.4.1Thao tác nội dung

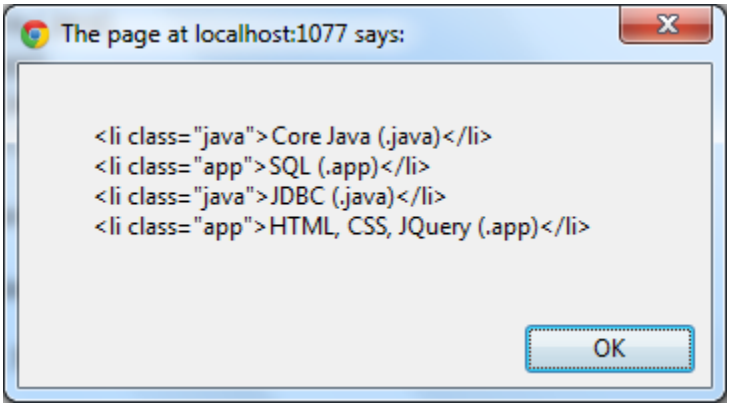
Jquery cung cấp 2 hàm cho phép thao tác nội dung là $(selector).html([HTML]) và $(selector).text([text]). Cả 2 hàm này cùng có 2 cách dùng. Nếu có tham số thì dùng để ghi (thay đổi nội dung) ngược lại thì lấy nội dụng. Mô tả chi tiết như sau.

METHOD	Ý NGHĨA	VÍ DỤ
html()	Lấy nội dung html bên trong element đầu tiên thỏa selector	HTML = $("#basic").html();
html(newContent)	Thay đổi nội dung html bên trong mọi element thỏa selector (tương tự innerHTML trong DOM)	$(".app").html("Hello");
text()	Lấy nội dung text bên trong element đầu tiên	Text = $("#basic").text();

text(newTextContent)	Thay đổi nội dung text bên trong mọi element thỏa selector (tương tự innerText)	$(“.app”).html(“Hello”);

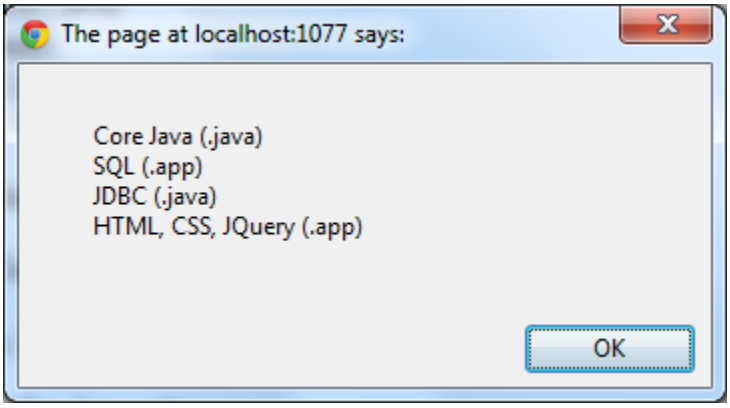
Ví dụ:

```
var html = $("#basic").html();
alert(html);
```



Hình 6.12: HTML

```
var text = $("#basic").text();
alert(text);
```



Hình 6.13: Văn bản

Các hàm bổ sung nội dung

METHOD	Ý NGHĨA
append(content)	Chèn content vào sau nội dung có sẵn của các element thỏa selector
appendTo(selector)	Chèn element thỏa selector vào sau nội dung có sẵn của các element thỏa selector tham số
prepend(content)	Chèn content vào trước nội dung có sẵn của các element thỏa selector
prependTo(selector)	Chèn element thỏa selector vào trước nội dung có sẵn của các element thỏa selector tham số

after(content)	Chèn content vào sau các element thỏa selector
before (content)	Chèn content vào trước các element thỏa selector

Ví dụ:

GỌI HÀM	TRƯỚC KHI CHÈN	SAU KHI CHÈN
$(“li”).append(“Subject”);	Java	Java Subject
$(“p:first”).appendTo(“li”);	Subject Java	Java Subject
$(“li”).prepend(“Subject: ”);	Java	Subject : Java
$(“p:first”).prependTo(“li”);	Subject Java	Subject Java
$(“li”).after(“Subject”)		Java Subject
$(“li”).before (“Subject”)		Subject Java

3.4.2Thao tác thuộc tính

METHOD	Ý NGHĨA	VÍ DỤ
attr(name)	Lấy attribute value của element đầu tiên thỏa selector	v = $(“#txtId”).attr(“value”)
attr(name, value)	Thiết lập attribute cho mọi element thỏa selector	$(“#txtId”).attr(“value”, “Hello”)
attr(name, function)	Thiết lập giá trị attribute dựa trên 1 function với mọi element thỏa selector.	$(“#txtId”).attr(“value”, function(){ return “Hello”; });
attr(properties)	Thiết lập tập attribute cho mọi element thỏa selector. Properties có dạng object-notation syntax.	$(“#txtId”).attr({“value”:”Hello”, “disabled”:”true”})
removeAttr(name)	Xóa attribute với mọi element	$(“#txtId”).removeAttr(“disabled”)
val([value])	Viết tắc của attr(“value”) để thao tác thuộc tính các phần tử form	$(“#txtId”).val(“Hello”); V=$(“#txtId”).val();

3.4.3 Thao tác CSS

Để làm việc với các thuộc tính CSS, Jquery cung cấp 3 nhóm hàm. Nhóm 1 chuyên thao tác từng thuộc tính CSS, nhóm 2 thao tác nguyên class và nhóm 3 là các hàm chuyên thao tác kích thước.

3.4.3.1 Thao tác thuộc tính CSS

METHOD	Ý NGHĨA	VÍ DỤ
css (name)	Lấy giá trị thuộc tính name của element đầu tiên thỏa selector	w = $(“img”).css(“width”);
css (properties)	Thiết lập tập thuộc tính css đối với mọi element thỏa selector	$(“div:first”).css({“text-align”:”center”, “background-image”:”url(bg.gif)”})
css (property,value)	Thiết lập giá trị 1 thuộc tính đối với mọi element thỏa selector	$(“tr:odd”).css(“background-color”:”red”})

3.4.3.2 Thao tác lớp

METHOD	Ý NGHĨA
addClass (class)	Thêm class vào các element thỏa selector
hasClass (class)	Kiểm tra class có tồn tại trong các element thỏa selector
removeClass(class)	Xóa class khỏi các element thỏa selector
toggleClass (class)	Thêm class vào các element thỏa selector nếu class chưa khai báo, ngược lại nếu đã tồn tại rồi, class sẽ bị xóa

3.4.3.3 Thao tác kích thước

METHOD	Ý NGHĨA	VÍ DỤ
height ()	Lấy chiều cao của element đầu tiên thỏa selector	H=$(“#MyImage”).height()
width ()	Lấy chiều rộng của element đầu tiên thỏa selector	W=$(“#MyDiv”).width()
height (val)	Thiết lập chiều cao của mọi element thỏa selector	$(“#MyImage”).height(300);
width (val)	Thiết lập chiều rộng của mọi element thỏa selector	$(“#MyImage”).width(400)

3.5 Xử lý sự kiện

Trong phần giới thiệu, chúng tôi đã giới thiệu sự kiện click để xử lý thao tác click lên các nút $(“:button”).click(function(){…}). Với cú pháp này chúng ta có 3 phần:

- ✓ **$(“:button”)**: selector (bộ chọn) đã được giới thiệu

- ✓ **.click()**: sự kiện, chúng tôi sẽ giới thiệu trong phần này
- ✓ **function(){…}**: handler, hàm xử lý sự kiện

Vì vậy toàn bộ dòng lệnh $(":button").click(function(){…}) được hiểu là đăng ký xử lý sự kiện click cho các thẻ <input type="button">. Như vậy một sự kiện cần được xử lý lúc xảy ra thì phải đăng ký trước.

3.5.1Các sự kiện thường gặp

Jquery cung cấp đầy đủ các hàm bẩy sự kiện trên web và tăng cường thêm. Sau đây là các sự kiện thường gặp trong lập trình Jquery.

EVENT	Ý NGHĨA	VÍ DỤ
click(func)	Nhấp chuột	$("a:first").click(function(){alert("Hello")})
dbclick(func)	Nhấp đúp chuột	$(":button").click(function(){alert("Hello")})
mouseover(func)	Chuột đi vào	$("img").mouseover(function(){$("div").toggleClass("highlight");})
mouseout(func)	Chuột đi ra	$("img").mouseout(function(){$("div").toggleClass("highlight");})
mousedown(func)	Đè chuột	$(":text").mousedown(function(){$(this).css("color", "red")})
mouseup(func)	Nhả chuột	$(":text").mouseup(function(){$(this).css("color", "black")})
mousemove(func)	Di chuyển chuột	$("#b"). mousemove (function(){alert("Hello")})
keydown(func)	Đè phím	$(":text"). keydown (function(){$(this).css("color", "red")})
keypress(func)	Gõ phím	$("#b").click(function(){alert("Hello")})
keyup(func)	Nhả phím	$(":text").mouseup(function(){$(this).css("color", "black")})
submit(func)	Submit form	$("#b"). submit (function(){alert("Good luck")})
blur(func)	Mất tích cực	$("#b"). blur (function(){alert("Hello")})
focus(func)	Tích cực	$("#b"). focus (function(){alert("Hello")})
hover(func1, func2)	Vào/ra	$("#b"). hover(function(){alert("Hi")}, function(){alert("Bye")})

3.5.2Đối tượng Event

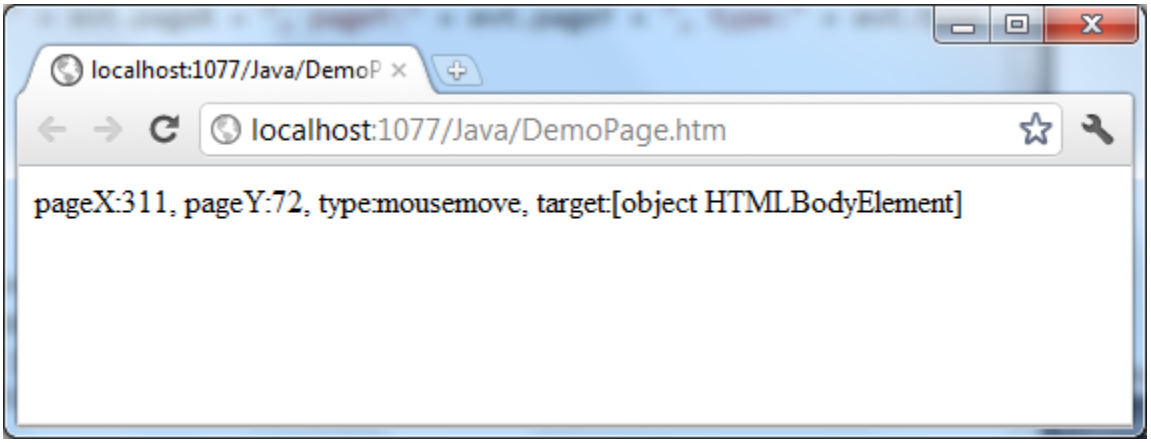
Khi sự kiện xảy ra, các thông tin liên quan đến sự kiện được gọi gọn trong đối tượng event và truyền vào hàm xử lý sự kiện. Các thông tin mà đối tượng event cung cấp bao gồm.

Thuộc tính /Phương thức	Ý nghĩa

Type	Loại event xảy ra, ví dụ: “click”
target	Element mà event xảy ra
data	Dữ liệu được truyền vào handler bởi phương thức bind
pageX, pageY	Tọa độ chuột khi event xảy ra
preventDefault ( )	Ngăn trình duyệt không thực thi xử lý mặc định, ví dụ khi click vào liên kết

Ví dụ sau sẽ ghi nhận thông tin sự kiện và hiển thị lên trang web khi chuột di chuyển trên trang web.

```
$("body").mousemove(function (e) {
    $(this).html("pageX:" + e.pageX + ", pageY:" + e.pageY
                 + ", type:" + e.type + ", target:" + e.target);
});
```



Hình 6.14: Xử lý đối tượng sự kiện

3.6 Hiệu ứng và hoạt ảnh

Hiệu ứng thông thường có 2 loại là xuất hiện và biến mất với một hình thức nào đó. Jquery cung cấp rất nhiều hiệu ứng, giúp chúng ta tạo ra trang web thật sự sinh động và hập dẫn. Trong phần này xin được giới thiệu các hiệu ứng thường gặp sau:

- ✓ Ẩn, hiện element
- ✓ Fade-in, fade-out
- ✓ Sliding
- ✓ Di chuyển element
- ✓ Hiệu ứng chuyển động tùy chỉnh

Hầu hết các phương thức hiệu ứng đều có 2 tham số tùy chọn là speed và callback. Speed qui định thời gian kéo dài của hiệu ứng giá trị của nó là ***“slow”, “normal”, “fast” hoặc milli giây***. Callback là hàm gọi ngược tức hàm được thực hiện sau khi hiệu ứng hoàn tất.

3.6.1Ẩn/hiện element

Có 3 phương thức điều khiển hiển thị là show() – hiện, hide()-ẩn và toggle()-ẩn/hiện.

METHOD	Ý NGHĨA	VÍ DỤ
show ()	Hiển thị các element thỏa selector nếu trước đó bị ẩn	
show(speed, callback)	Hiển thị các element thỏa selector nếu trước đó bị ẩn, speed xác định tốc độ hiển thị. Sau khi hiển thị xong, phương thức callback sẽ được thực thi.	$("#div1").show("normal", function(){alert("Hello")});
hide ()	Ẩn element nếu trước đó đang hiển thị.	
hide(speed,callback)	Ẩn element nếu trước đó đang hiển thị, tham số có ý nghĩa tương tự phương thức show.	$("#div1").hide("slow"); $("#div1").hide(4000);
toggle( )	Chuyển qua lại trạng thái ẩn/hiện các element.	
toggle(speed,callback)	Chuyển qua lại trạng thái ẩn/hiện các element, tham số có ý nghĩa tương tự phương thức show.	$("#div1").toggle("fast");

3.6.2Fade in/fade out

METHOD	Ý NGHĨA	VÍ DỤ
fadeIn(speed,callback)	Hiển thị element bằng cách tăng dần độ trong suốt.	$("#div1").fadeIn("normal");
fadeOut(speed,callback)	Ẩn element bằng cách giảm dần độ trong suốt về 0, sau đó thiết lập style display là none.	$("#div1").fadeOut("slow");
fadeTo(speed,opacity, callback)	Thay đổi độ trong suốt của element.	$("#div1").fadeTo("slow",0.3, function() {alert("finished");}); $("#div1").fadeTo("slow",1.0);

3.6.3Sliding

METHOD	Ý NGHĨA	VÍ DỤ
slideDown(speed, callback)	Hiển thị element bằng cách tăng chiều cao.	$("#div1").slideDown("slow");

slideUp(speed, callback)	Ẩn element bằng cách giảm chiều cao.	$("#div1").slideUp("normal");
slideToggle(speed, callback)	Chuyển đổi trạng thái ẩn/hiện element.	$("#div1").slideToggle(3000);

3.6.4Custom Animation

Sử dụng phương thức animate() để tạo hiệu ứng tùy chỉnh. Bạn có cơ hội tạo ra các hiệu ứng cho riêng mình bằng cách điều chỉnh các tham số truyền vào phương thức. Trong trường hợp bạn muốn dừng hiệu ứng thì bạn chỉ việc gọi phương thức stop.

$(“selector").animate(properties,[duration], [easing],[callback]);

$(“selector").stop();

THAM SỐ	Ý NGHĨA	VÍ DỤ
properties	Các thuộc tính xác trạng thái hiển thị sau khi animate.	{width:”100”, color:”red”}
duration	Animate kéo dài trong bao lâu (“slow”, “normal”, “fast”, milisecond)	1000
easing	Hiệu ứng xóa : swing, linear	swing
Callback	Hàm được gọi sau khi animate xong	function(){alert(“Hello”)}

Một số ví dụ về hiệu ứng tùy chỉnh

```
$("#button_growright").click(function () {
    $("#div1").animate({ width: "800" }, "normal");
});
$("#button_growleft").click(function () {
    $("#div1").animate({ width: "100" }, "fast");
});
$("#button_bigtext").click(function () {
    $("#div1").animate({ fontSize: "40" }, 2000);
});
$("#button_movediv").click(function () {
    $("#div1").animate({ left: "500", fontSize: "50" }, 1000, "linear");
});
```

3.7 Form Validation

Kiểm duyệt dữ liệu form nhập trước khi chuyển dữ liệu đến server để xử lý là nhiệm vụ cực kỳ quan trọng. Rất nhiều tiêu chí được đặt ra để kiểm duyệt tùy thuộc vào yêu cầu cụ thể của form nhập liệu. Dù sao vẫn có thể liệt kê một số tiêu chí kiểm duyệt chung chung như sau:

- ✓ Không cho để trống ô nhập…
- ✓ Dữ liệu nhập vào phải theo một khuôn dạng nhất định nào đó: email, creditcard, url…
- ✓ Dữ liệu phải nhập vào phải đúng kiểu: số nguyên, số thực, ngày giờ…
- ✓ Dữ liệu nhập vào phải có giá trị tối thiểu, tối đa, trong phạm vi…
- ✓ Dữ liệu nhập phải đúng theo một kết quả tính toán riêng của bạn…

Bây giờ chúng ta hãy khám phá khả năng kiểm duyệt dữ liệu đầu vào của Jquery.

3.7.1Khởi động nhanh với Jquery Validation

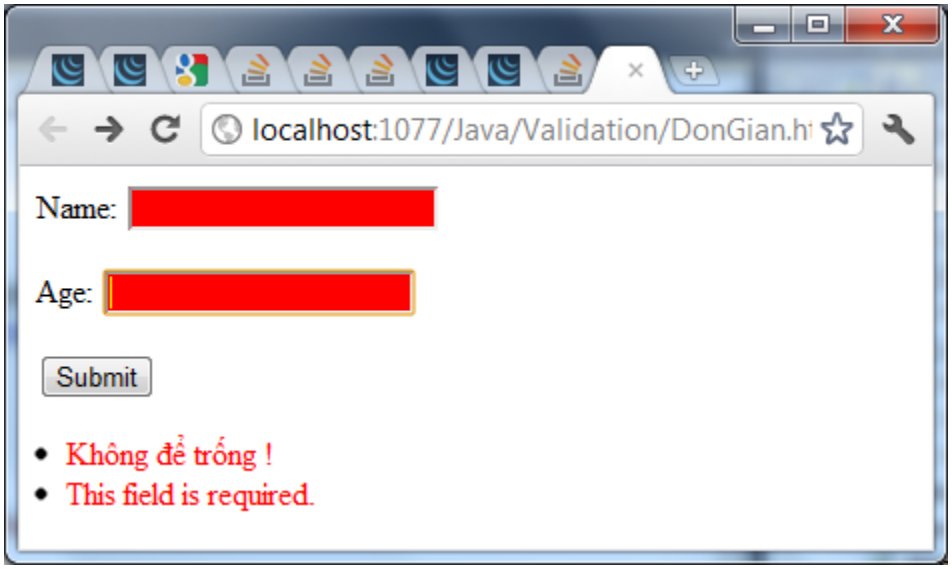
3.7.1.1Ví dụ

Ví dụ sau đây kiểm tra ô nhập “Name” phải nhập ít nhất 3 ký tự và ô nhập “Age” phải nhập số từ 25 đến 65.

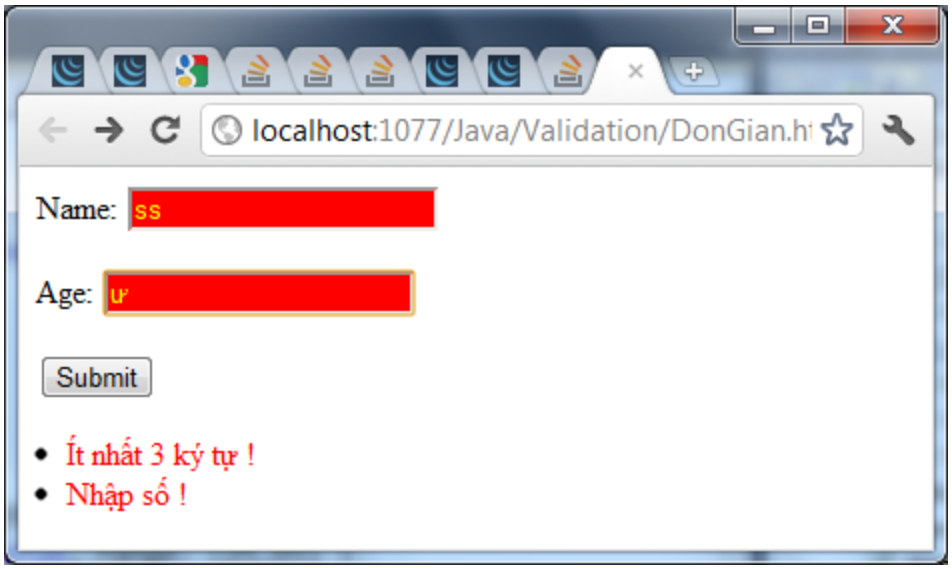
```
<html>
<head>
<script src="../Scripts/jquery-1.4.1.js" type="text/javascript"></script>
<script src="jquery.validate.js" type="text/javascript"></script>
<script type="text/javascript">
$(document).ready(function () {
    $("#form1").validate(
      {
          rules:
            {
                txtName: { required: true, minlength: 3 },
                txtAge: { required: true, digits: true, range: [25,65] }
            },
          messages:
            {
                txtAge: { digits: "Nhập số !" },
                txtName: { required: "Không để trống !", minlength: "Ít nhất
3 ký tự !" }
            },
          errorLabelContainer: "#myError",
          wrapper: "li",
          submitHandler: function (form) {
              if (confirm("Dữ liệu form đã hợp lệ. Bạn có muốn submit không
?")) {
                  form.submit();
              }
          }
```

```
      });
});
</script>
<style type="text/css">
    label.error{color:Red;}
    input.error { background-color:Red; color:yellow;}
</style>
</head>
<body>
<form id="form1" name="form1" method="post" action="">
  <p>
    Name:
    <input name="txtName" type="text" id="txtName" />
 </p>
  <p>
    Age:
    <input name="txtAge" type="text" id="txtAge" />
  </p>
  <p>
    <input type="submit" name="Submit" value="Submit" />
  </p>
  <div id="myError" />
</form>
</body>
</html>
```

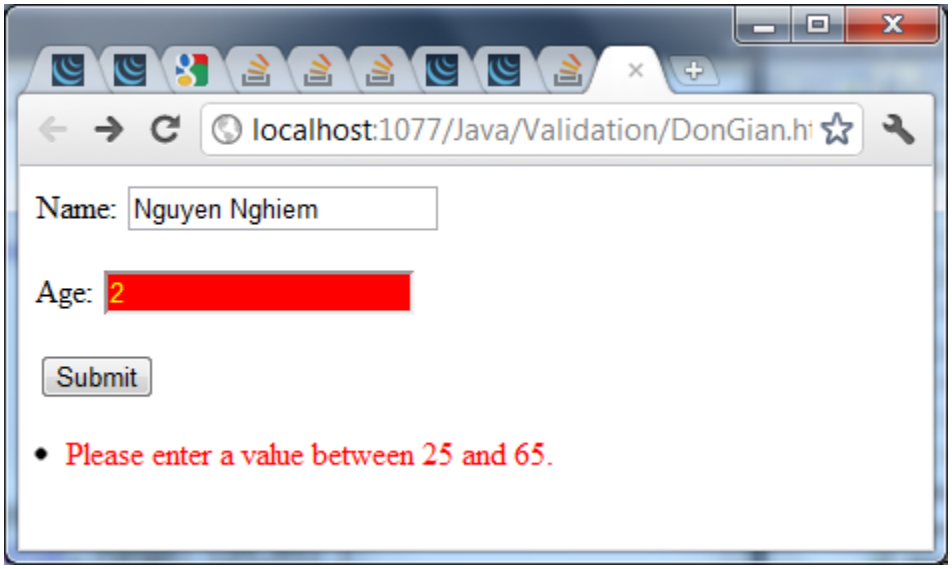
Chạy ứng dụng với các tình huống sau



Hình 6.15: không nhập gì cả



Hình 6.16: nhập "Name" ít hơn 3 ký tự, "Age" không phải số



Hình 6.17: nhập "Name" hợp lệ và "Age" số không thuộc (25, 65)

3.7.1.2 Phân tích

Thư viện cần thiết cho việc bẩy lỗi:

```
<script src="jquery.validate.js" type="text/javascript"></script>
```

Cấu trúc cơ bản của phương thức validate() dùng để cài đặt các tùy chọn bẩy lỗi.

```
$("#form1").validate(
{
          rules: {<khai báo luật bẩy lỗi cho các trường>},
          messages: {<định nghĩa các thông báo lỗi>},
          errorLabelContainer: "<khai báo thẻ chứa lỗi>",
          wrapper: "<khai báo thẻ bọc lỗi>",
```

```
        submitHandler: <hàm xử lý submit>
});
```

Trong bài này:

✪ **Khai báo luật bẩy lỗi cho các trường**

```
rules:
{
    txtName: { required: true, minlength: 3 },
    txtAge: { required: true, digits: true, range: [25,65] }
}
```

- ✓ txtName: không được để trống, phải có ít nhất 3 ký tự
- ✓ txtAge: không được để trống, phải là số nguyên và thuộc khoản (25, 65)

✪ **Định nghĩa các thông báo lỗi**

```
messages:
{
  txtAge: { digits: "Nhập số !" },
  txtName: { required: "Không để trống !", minlength: "Ít nhất 3 ký tự !" }
}
```

- ✓ txtName: thông báo khi để trống và thông báo khi nhập ít hơn 3 ký tự
- ✓ txtAge: thông báo khi không nghập đúng số nguyên
- ❖ Chú ý: trong trường hợp không định nghĩa thông báo lỗi, hệ thông sẽ đưa ra thông báo lỗi mặc định (“this field is required”, “please enter value between 25 and 65”…)

✪ **Khai báo thẻ chứa lỗi**

```
errorLabelContainer: "#myError",
```

- ✓ Chỉ định vị trí đặt lỗi tại thẻ có id=”myError”. Trong trường hợp nay là .
- ❖ Nếu chúng ta không chỉ định vị trí đặt lỗi thì vị trí mặt định của nó là ngay sau phần tử phạm lỗi.

✪ **Khai báo thẻ bọc lỗi**

```
wrapper: "li"
```

- ✓ Mỗi thông báo lỗi được bọc trong thẻ <li>. Nếu chúng ta không chỉ định thì lỗi sẽ xếp liên tiếp mà không xuống hàng.

✪ **Hàm xử lý submit**

```
submitHandler: function (form) {…}
```

- ✓ Tùy chọn này cho phép chúng ta viết mã JavaScript để điều khiển việc submit form. Nếu không có ttùy chọn này thì form sẽ tự động được submit sau mọi lỗi đã được kiểm tra hợp lệ.

3.7.2Luật kiểm tra (validation rules)

Sau đây là bảng định nghĩa các qui luật kiểm lỗi đã được định nghĩa trước bởi Jquery.

RULE	VALUE	Ý NGHĨA	VÍ DỤ
required	boolean	Bắt buộc nhập	required:true

required	selector	Bắt buộc nhập nếu tập kết quả của selector rỗng	required:"#abc:blank"
required	boolean callback function	Bắt buộc nhập nếu kết quả trả về có giá trị false.	required: function(){return true;}
remote	boolean remote check url	Hợp lệ khi kết quả kiểm tra từ xa là false.	remote: "check.jsp"
minlength	length	Số ký tự tối thiểu	minlength:10
maxlength	length	Số ký tự tối đa	maxlength:100
rangelength	[minlength, maxlength]	Số ký tự từ min đến max	rangelength:[10, 100]
min	value	Giá trị tối thiểu	min:10
max	value	Giá trị tối thiểu	max:100
range	[minvalue, maxvalue]	Giá trị từ min đến max	range:[10,100]
email	boolean	Định dạng email	email:true
url	boolean	Đạnh dạng url	url:true
date	boolean	Đạnh dạng ngày javascript	date:true
dateISO	boolean	Định dạng ngày ISO	dateISO:true
number	boolean	Số thực	number:true
digits	boolean	Số nguyên	digits:true
creditcard	boolean	Định dạng creditcard	creditcard:true
accept	list of file extensions	Kiểu mở rộng file	accept:"doc,xsl,pdf"
equalTo	Selector	So sánh giá trị của phần tử và giá trị của selector	equalTo:"#RetypeSelector"

Chú ý: bạn có 2 cách để khai báo luật bẩy lỗi

1. Khai báo trong tùy chọn **rules** như trong ví dụ trên trên
2. Khai báo ngay trong thẻ bạn muốn bẩy lỗi

Ví dụ để kiểm lỗi cho ô nhập txtAge của ví dụ trên, bạn có thể khai báo ngay trên thẻ <input> như sau:

```
<input class="required digits" min=”25” max=”65” id="txtAge" />
```

3.7.3Luật do người dùng định nghĩa (user-defined rules)

Trên đây chỉ là danh sách các luật phổ thông hàng ngày. Bạn có thể có những qui luật riêng của mình mà chỉ có bạn mới có thể hiểu và định nghĩa được. Vì vậy Jquery cung cấp cho bạn một cách định nghĩa các luật mới của riêng mình. Hãy xem và phân tích ví dụ sau để hiểu rõ cách để định nghĩa một luật mới.

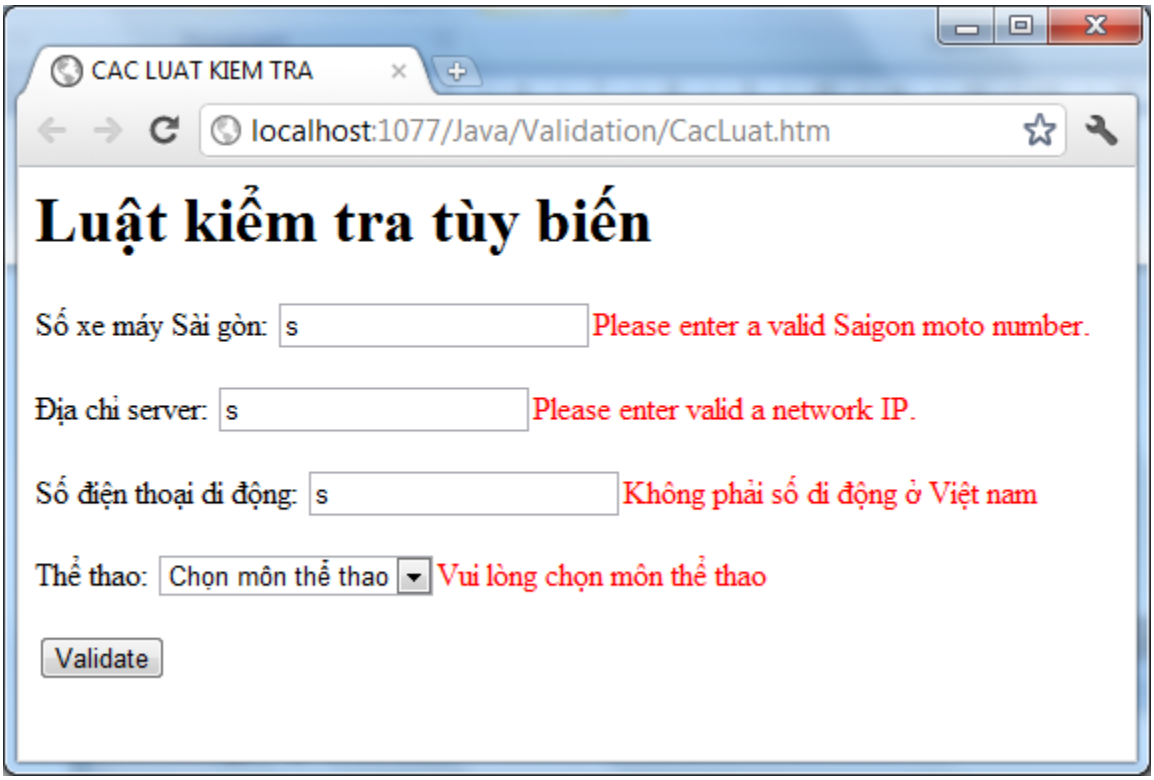
```
<html>
<head>
<script src="../JQuery/js/jquery-1.4.1.min.js"></script>
<script src="../JQuery/js/jquery.validate.js"></script>
<script type="text/javascript">

/*--Định nghĩa hàm kiểm tra số di động việt nam--*/
function fnValidateMobile(value, element) {
    var regex = /^0[0-9]{9,10}$/g;
    return this.optional(element) || regex.test(value);
}
/*--Định nghĩa hàm kiểm tra số xe gắn máy sài gòn--*/
function fnValidateSaigonMoto(value, element) {
    var regex = /^5\d-[A-Z]\d-\d{4}$/g;
    return this.optional(element) || regex.test(value);
}
/*--Định nghĩa hàm kiểm tra IP mạng máy tính--*/
function fnValidateNetuworkIP(value, element) {
    var regex = /^\d{3}\.\d{3}\.\d{3}\.\d{3}$/g;
    if (this.optional(element) || regex.test(value)) {
        var nums = value.split(".");
        for (var i = 0; i < nums.length; i++) {
            if (parseInt(nums[0]) > 255) {
                return false;
            }
        }
    }
    else {
        return false;
    }
    return true;
}
/*--Định nghĩa hàm kiểm tra mục chọn của combo box--*/
function fnValidateSelectOne(value, element) {
    return (element.value != "none");
}
```

```
/*--Định nghĩa luật kiểm tra kết hợp với hàm và một thông
    báo lỗi nếu kết quả trả về của hàm có giá trị false--*/
$.validator.addMethod("selectone", fnValidateSelectOne,
                                    "Please select an item.");
$.validator.addMethod("vinaphone", fnValidateMobile,
                                    "Please enter a valid VinaPhone number.");
$.validator.addMethod("saigonmoto", fnValidateSaigonMoto,
                                    "Please enter a valid Saigon moto number.");
$.validator.addMethod("networkip", fnValidateNetuworkIP,
                                    "Please enter valid a network IP.");
</script>

<script type="text/javascript">
$(document).ready(function () {
    $("#form1").validate(
      {
            rules:
            {
                  sport: { selectone: true },
                  mobile: { vinaphone: true }
            },
            messages:
            {
                  sport: { selectone: "Vui lòng chọn môn thể thao" },
                  mobile: { vinaphone: "Không phải số di động ở Việt nam" }
            }
      });
});
</script>
<style type="text/css">
label.error
{
    color: Red;
}
</style>
</head>
<body>
    <h1>Luật kiểm tra tùy biến</h1>
    <form id="form1">
        <p>Số xe máy Sài gòn:
        <input type="text" id="moto" name="moto" class="required saigonmoto">
        <p>Địa chỉ server:
        <input type="text" id="ip" name="ip" class="networkip">
```

```
        <p>Số điện thoại di động:
        <input type="text" id="mobile" name="mobile">
        <p>
        Thể thao:
        <select name="sport" id="sport">
            <option value="none">Chọn môn thể thao</option>
            <option value="baseball">Bóng chày</option>
            <option value="basketball">Bóng rỗ</option>
            <option value="volleyball">Bóng chuyền</option>
            <option value="football">Bóng đã</option>
        </select>
        <p><input class="submit" type="submit" value="Validate">
    </form>
</body>
</html>
```



Hình 6.18: Kiểm lỗi form nhập

Phân tích ví dụ:

Trong bài trên chúng ta định nghĩa 4 luật kiểm tra mới là vinaphone, saigonmoto, networkip và selectone. Và sau đó áp dụng để kiểm tra dữ liệu cho các thành phần giao diện trên form. Để hiểu được cơ chế định nghĩa và sử dụng chúng ta cần thực hiện các bước sau.

- ✪ **Bước 1: Định nghĩa**. cần 2 bước là viết hàm kiểm tra và khai báo luật kiểm với Jquery
- ❖ **Viết hàm kiểm tra:**

```
/*--Định nghĩa hàm kiểm tra số di động việt nam--*/
function fnValidateMobile(value, element) {
    var regex = /^0[0-9]{9,10}$/g;
    return this.optional(element) || regex.test(value);
}
```

Cú pháp của hàm này phải nhận 2 tham số vào là value (giá trị nhập vào) và element (phần tử gây lỗi). Hàm này phải trả về kết quả true (đã hợp lệ) hoặc false (không hợp lệ). Bạn có thể phân tích giá trị để thực hiện kiểm tra nhờ vào tham số value và thay đổi css hay giá trị của phần tử này thông qua tham số element.

❖ **Khai báo luật kiểm với Jquery**

Sau khi đã định nghĩa hàm kiểm tra, bước tiếp theo là định nghĩa luật kiểm tương ứng với hàm trên và tất nhiên cung cấp thông báo lỗi.

```
/*--Định nghĩa luật kiểm tra kết hợp với hàm và một thông
    báo lỗi nếu kết quả trả về của hàm có giá trị false--*/
$.validator.addMethod("vinaphone", fnValidateMobile,
                                    "Please enter a valid VinaPhone number.");
```

Sử dụng phương thực $.validator.addMethod(rule, method, message) để khai báo luật kiểm. Tham số rule (“vinaphone”) là tên luật mới, tham số method (“`fnValidateMobile`”) là tên phương thức kết hợp với luật mới và message (“`Please enter a valid VinaPhone number.`”) là thông báo lỗi.

✪ **Bước 2: sử dụng**

Bạn sử dụng các luật mới như các luật đã định nghĩa sẵn trong Jquery. Cụ thể là bạn có thể chỉ định trong tùy chọn rules (`rules: {mobile: { vinaphone: true }}`) của phương thức validate hoặc chỉ ra trên thẻ cần kiểm tra “`<input type="text" id="moto" name="moto" class="required saigonmoto">`”.

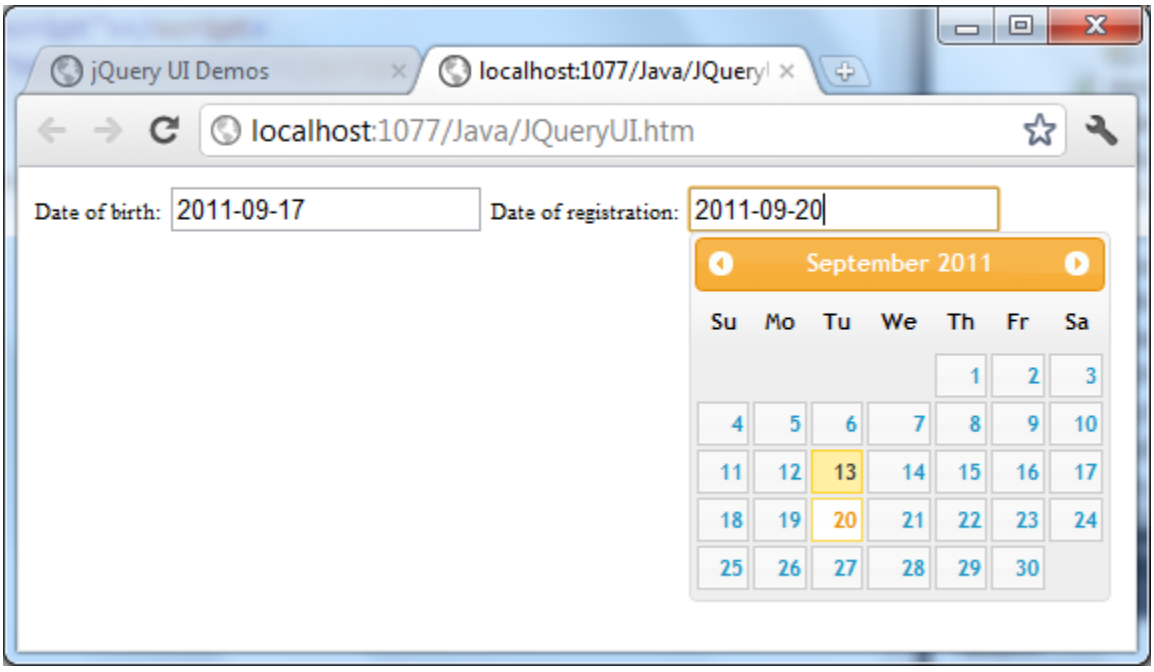
3.8 Các thành phần giao diện

Giao diện đóng vai trò quan trọng trong việc làm web. Gioa diện tốt thể hiện ở chỗ đẹp, hợp logic, tương thích với mọi trình duyệt và tốc độ tải nhanh. Bộ thư viện jquery-ui gần như đáp ứng hoàn toàn những yêu cầu đề ra. Để làm việc với thành phần giao diện bạn cần thư viện css và javascript được cung cấp bởi Jquery sau.

```
// phần thư viện lõi thường dùng trước đây
<script src="jquery-1.4.4.min.js"></script>
// css cần thiết cho các thành phần giao diện
<link href="jquery-ui-1.8.7.custom.css" rel="stylesheet"/>
// thư viện xử lý các thành phần giao diện
<script src="jquery-ui-1.8.7.custom.min.js"></script>
```

3.8.1Datepicker

Datepicker là thành phần giao diện cần thiết để nhập ngày tháng nhanh chóng và tránh sai sót. Nó cũng cho phép chúng ta định dạng tùy thích.



Hình 6.19: Datepicker

```
<html>
<head>
    <link href="jquery-ui-1.8.7.custom.css" rel="stylesheet"/>
    <script src="jquery-1.4.4.min.js"></script>
    <script src="jquery-ui-1.8.7.custom.min.js"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            $("#birthday, #joined").datepicker({ dateFormat: 'yy-mm-dd' });
        });
    </script>
    <style type="text/css">
        body{font-size: 11px;}
    </style>
</head>
<body>
    Date of birth: <input id="birthday" />
    Date of registration: <input id="joined" />
</body>
</html>
```

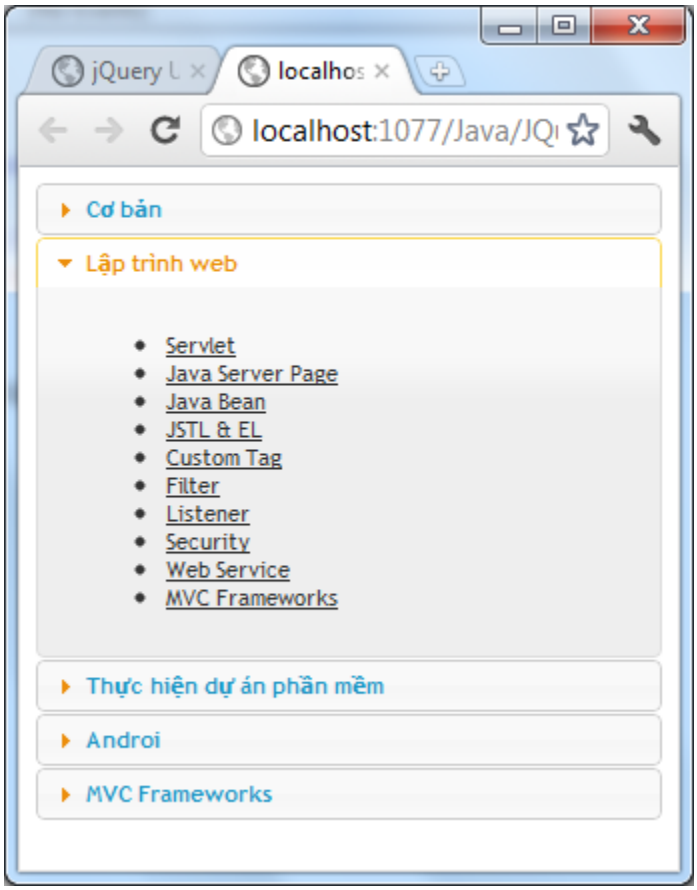
Phân tích ví dụ: ngoài các dòng chỉ định thư viện cần dùng, bạn cần lưu ý dòng lệnh quan trọng nhất ở đây là:

```
$("#birthday, #joined").datepicker({ dateFormat: 'yy-mm-dd' });
```

Cho biết 2 ô nhập #birthday và #joined áp dụng datepicker. Phần tùy chọn cho biết định dạng ngày là năm-tháng-ngày (mặc định là mm/dd/yyyy)

3.8.2Accordion

Accordion là thành phần giao diện chia làm nhiều phần nhưng tại một thời điểm chỉ cho phép hiển thị một vùng. Bạn cũng có thể qui định sự kiện click, mouseover…để thực hiện việc mở vùng.



Hình 6.20: sử dụng Accordion

```
<html>
<head>

<link href="jquery-ui-1.8.7.custom.css" rel="stylesheet" type="text/css" />
<script src=" jquery-1.4.4.min.js" type="text/javascript"></script>
<script src=" jquery-ui-1.8.7.custom.min.js" type="text/javascript"></script>
<script>
      $(function () {
          $("#accordion").accordion({autoHeight: false, event: "mouseover"});
      });
</script>
<style type="text/css">
    #accordion{font-size: 11px;}
</style>
```

```
</head>
<body>

<div id="accordion">
      <h3><a href="#">Cơ bản</a></h3>
      <div>...</div>

    <h3><a href="#">Lập trình web</a></h3>
      <div>
        <ul>
            <li><a href="#1">Servlet</a></li>
            <li><a href="#2">Java Server Page</a></li>
            <li><a href="#3">Java Bean</a></li>
            <li><a href="#4">JSTL & EL</a></li>
            <li><a href="#5">Custom Tag</a></li>
            <li><a href="#6">Filter</a></li>
            <li><a href="#7">Listener</a></li>
            <li><a href="#8">Security</a></li>
            <li><a href="#8">Web Service</a></li>
            <li><a href="#8">MVC Frameworks</a></li>
        </ul>
      </div>

    <h3><a href="#">Thực hiện dự án phần mềm</a></h3>
      <div>...</div>

    <h3><a href="#">Androi</a></h3>
      <div>....</div>

    <h3><a href="#">MVC Frameworks</a></h3>
      <div>....</div>
</div>

</body>
</html>
```

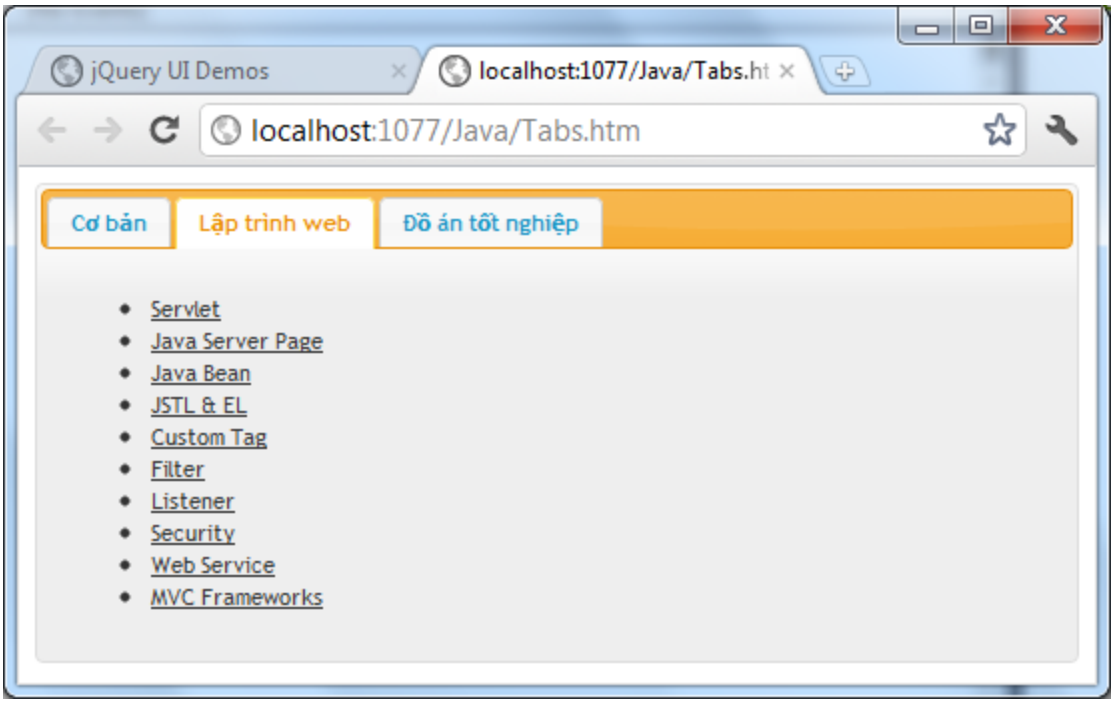
Phân tích ví dụ:

Phần HTML: mỗi vùng gồm tiêu đề (mặc định dùng thẻ <h3>) và nội dung (mặc định dùng thẻ ngay kế sau). Các vùng này đặt trong một thẻ container (bài này dùng thẻ).

Phần Jquery: **$("#accordion").accordion({autoHeight: false, event: "mouseover"});** khởi tạo vùng accordion. **autoHeight** là tùy chọn qui định chiều cao của mỗi vùng là không cố định còn tùy chọn **event** chỉ ra hành vi thực hiện mở vùng.

3.8.3Tab

Hình ảnh của tab quen thuộc với các bạn hơn Accordion. Tuy nhiên về bản chất, tab và Accordion lại giống nhau. Tại một thời điểm chỉ có một vùng được mở ra.



Hình 6.21: Sử dụng tab

```
<html>
<head>
    <link href="jquery-ui-1.8.7.custom.css" rel="stylesheet"/>
    <script src=" jquery-1.4.4.min.js"></script>
    <script src=" jquery-ui-1.8.7.custom.min.js"></script>
      <script>
          $(function () {
              $("#tabs").tabs({ event: "mouseover"});
          });
      </script>

    <style type="text/css">
        #tabs{font-size: 11px;}
    </style>
</head>
<body>

<div id="tabs">
      <ul>
            <li><a href="#tabs-1">Cơ bản</a></li>
            <li><a href="#tabs-2">Lập trình web</a></li>
            <li><a href="#tabs-3">Đồ án tốt nghiệp</a></li>
      </ul>
      <div id="tabs-1">...</div>
      <div id="tabs-2">
       <ul>
            <li><a href="#1">Servlet</a></li>
```

```
            <li><a href="#2">Java Server Page</a></li>
            <li><a href="#3">Java Bean</a></li>
            <li><a href="#4">JSTL & EL</a></li>
            <li><a href="#5">Custom Tag</a></li>
            <li><a href="#6">Filter</a></li>
            <li><a href="#7">Listener</a></li>
            <li><a href="#8">Security</a></li>
            <li><a href="#8">Web Service</a></li>
            <li><a href="#8">MVC Frameworks</a></li>
        </ul>
      </div>
      <div id="tabs-3">...</div>
</div>
</body>
</html>
```

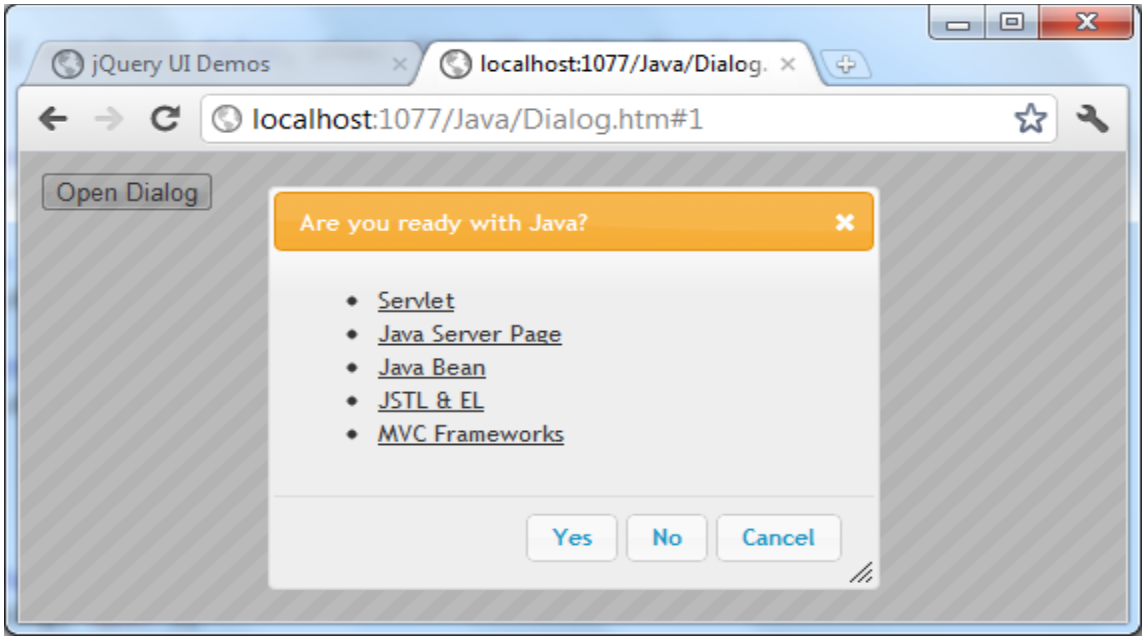
Phân tích ví dụ:

Phần HTML: một thẻ được dụng làm container. Một thẻ <ul> chứa các tiêu đề tab, các liên kết của mỗi tiêu đề chỉ đến thuộc tính id của các thẻ chứa nội dung tab.

Phần Jquery: $("#tabs").tabs({ event: "mouseover"}); khởi động vùng tabs và chỉ định sự kiện thực hiện hành vi chọn tab.

3.8.4Dialog

Hộp thoại cũng là thành phần cần thiết để thực hiện các công việc thông báo, tiếp nhận dữ liệu của người dùng một cách mềm dẻo.



Hình 6.22: Sử dụng Dialog

```
<html>
<head>
```

```
<link rel="stylesheet" type="text/css" href="jquery-ui-1.8.7.custom.css" />
<script src=" jquery-1.4.4.min.js" type="text/javascript"></script>
<script src=" jquery-ui-1.8.7.custom.min.js" type="text/javascript"></script>
<script type="text/javascript">
$(function () {
      $("#dialog").dialog({ autoOpen: false, show: "blind", hide: "explode",
          modal: true,
          buttons: {
              Yes: function () {
                  $(this).dialog("close");
              },
              No: function () {
                  $(this).dialog("close");
              },
            Cancel: function () {
                  $(this).dialog("close");
              }
          }
    });

      $("#opener").click(function () {
          $("#dialog").dialog("open");
          return false;
      });
});
</script>
<style type="text/css">
    body{font-size: 11px;}
</style>
</head>
<body>

<div id="dialog" title="Are you ready with Java?">
      <ul>
        <li><a href="#1">Servlet</a></li>
        <li><a href="#2">Java Server Page</a></li>
        <li><a href="#3">Java Bean</a></li>
        <li><a href="#4">JSTL & EL</a></li>
        <li><a href="#8">MVC Frameworks</a></li>
    </ul>
</div>

<button id="opener">Open Dialog</button>

</body>
</html>
```

Phân tích ví dụ:

Phần HTML: đóng vai trò như một container của hộp thoại. Nút <button id=”opener”> đóng vai trò là nút kích hoạt mở hộp thoại.

Phần Jquery:

- ✓ $("#dialog").dialog({tùy chọn}); khởi động hộp thoại với các tùy chọn như sau:
  - autoOpen: false là không cho mở hộp thoại lúc khởi động
  - show: "blind", hide: "explode" điều khiển hiệu ứng mở và đóng
  - modal: true là để không cho phép tương tác cửa sổ mẹ
  - buttons{} là khai báo các nút và hàm xử lý click nút
- ✓ $("#opener").click(); lệnh này là để mở hộp thoại khi click #opener

3.8.5Jquery plugins thông dụng

Trên thế giới còn rất nhiều jquery pluggin viết về giao diện và thự sự tuyệt vời, bạn có thể tìm kiếm, chia sẽ cùng bạn bè về thế giới jquery khắp nơi. Sau đây là một số từ khóa google để bạn có thể tìm thấy các thành phần giao diện khác hay hơn, hấp dẫn hơn:

- ✓ Jquery Scrollable
- ✓ Jquery TreeView
- ✓ Jquery Menu
- ✓ Jquery Round Conner

3.9 Ajax

3.9.1Giới thiệu

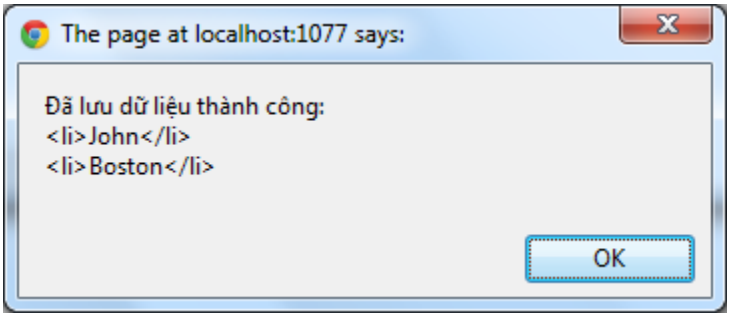
Đặc điểm nổi bậc nhất của Ajax là kỹ thuật tương tác với server theo cơ chế bất đồng bộ để tạo cảm giác như người dùng đang làm việc trên ứng dụng desktop mà không phải trên web. Jquery cung cấp một số hàm để thực hiện điều này. Trong bài này chúng tôi giới thiệu hàm ajax chung nhất của Jquery và chỉ thảo luận một số tùy chọn thường xuyên sử dụng. Xin mời các bạn đến với jquery.com để xem chi tiết hơn.

Cú pháp hàm $ajax:

```
$.ajax( <tùy chọn> )
```

<tùy chọn>: chứa các thông số để hàm thực hiện.

Ví dụ sau là kết quả nhận được khi chạy trang web Ajax.htm



Hình 6.23: phản hồi từ server

Trang Ajax.htm

```
<html>
<head>
    <script src="JQuery/js/jquery-1.4.1.min.js"></script>
    <script>
        $.ajax({
            url: "Ajax.jsp",
            data: "name=John&location=Boston",
            success: function (responseText) {
                alert("Đã lưu dữ liệu thành công: " + responseText);
            }
        });
    </script>
</head>
<body>
</body>
</html>
```

Trang Ajax.jsp

```
<li>${param.name}</li>
<li>${param.location}</li>
```

Phân tích kết quả:

Thư viện lõi là đủ để thực hiện ajax

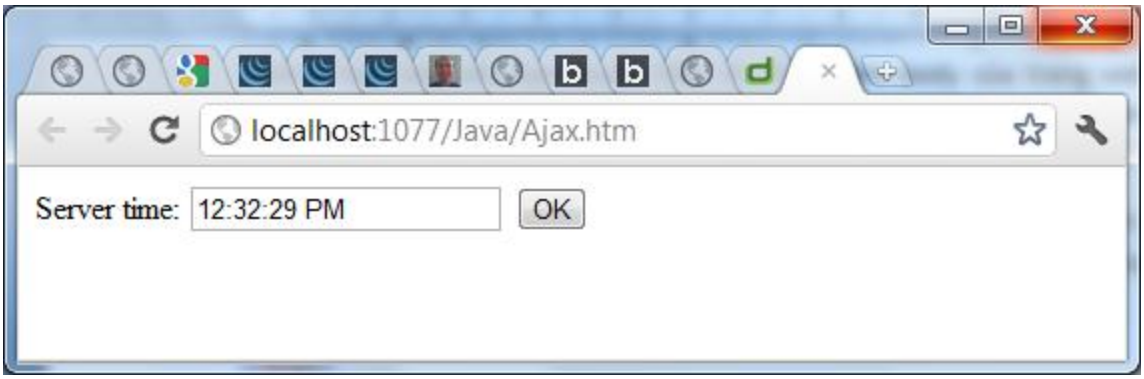
```
<script src="JQuery/js/jquery-1.4.1.min.js"></script>
```

Hàm $.ajax({tùy chọn}) gồm:

- ✓ url: chỉ ra trang web phía server cần tương tác (“Ajax.jsp”)
- ✓ data: dữ liệu (“name=John&location=Boston”) cần truyền đến server đẻ trang trong url (“Ajax.jsp”) xử lý
- ✓ success: chỉ hàm điều khiển kết quả trả về từ server. Trong bài này chỉ thông báo thông qua hộp thoại alert(). Tham số của hàm (responseText) là kết quả trả về từ server.

Thông thường kết quả trả về thường được hiển thị lên trang web thông qua một số thẻ nào đó. Trong trường hợp này nếu bạn muốn hiển thị lên phầm body của trang web thì có thể thay dòng lệnh `alert("Đã lưu dữ liệu thành công: " + responseText);` bằng dòng `$(“body”).html("Đã lưu dữ liệu thành công: " + responseText).`

Tải và thực hiện JavaScript: đôi khi chúng ta cần tải và thực hiện một đoạn JavaScript được sinh động từ phía server. Để làm điều đó chúng ta chỉ cần chỉ định tùy chọn dataType là script. Ví dụ sau khi bạn click vào nút OK thì thời gian trên ô nhập sẽ được cập nhật theo thời gian server.



Hình 6.24: Dùng ajax để lấy giờ server

Trang Ajax.htm

```
<html>
<head>
    <script src="jquery-1.4.1.min.js" type="text/javascript"></script>
    <script>
        $(document).ready(function () {
            $("button").click(function () {
                $.ajax({
                    type: "GET",
                    url: "ServerTime.jsp",
                    cache: false,
                    dataType: "script"
                });
            });
        });
    </script>
</head>
<body>
    Server time: <input id="name" />
    <button>OK</button>
</body>
</html>
```

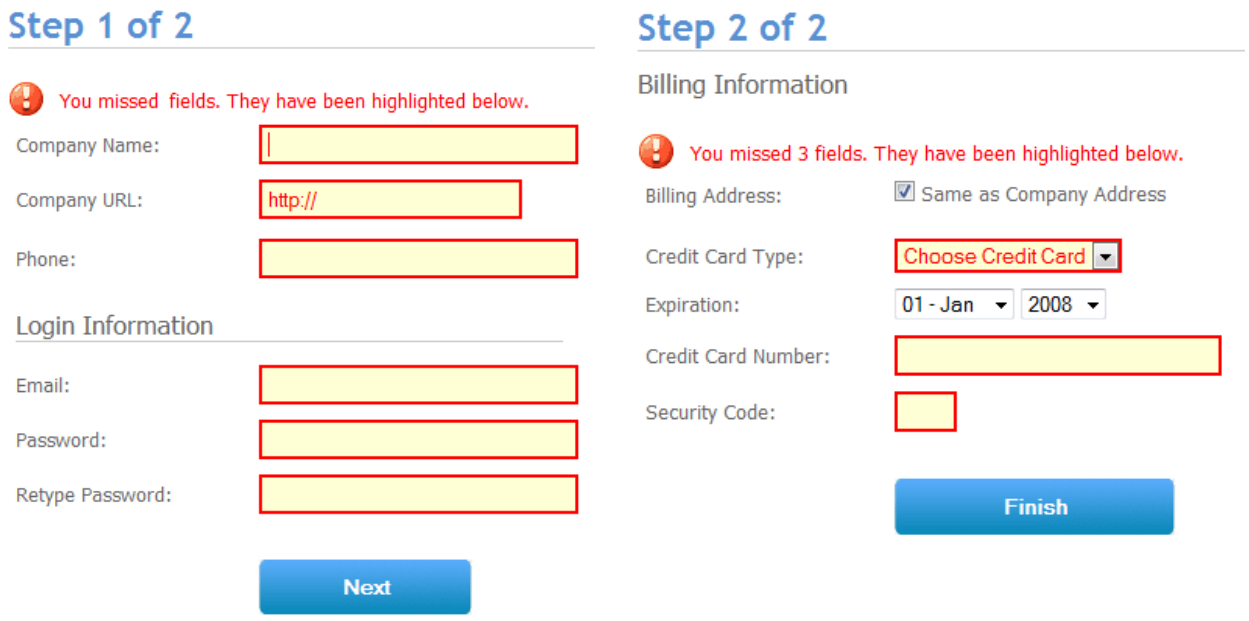
Trang ServerTime.jsp

```
<%@ taglib prefix="f" uri="http://java.sun.com/jsp/jstl/fmt" %>
<%request.setAttribute("time", new java.util.Date());%>
<f:formatDate value="${time}" pattern="HH:mm:ss"/>
```

3.10 Bài tập

Viết các bộ lọc thực hiện theo yêu cầu sau

1. Tìm tất cả các thẻ <input type='checkbox'>
2. Tìm tất cả các thẻ <option> của thẻ <select id="country">
3. Tìm các thẻ <img> đặt trong
4. Tìm tất cả các thẻ <li> có nội dung
5. Tìm tất cả các thẻ <input type='checkbox'> chưa được đánh dấu
6. Tìm tất cả các thẻ <input type='text'> chưa nhập dữ liệu
7. Đọc/ghi dữ liệu form
   - ✓ TextBox
   - ✓ Radio
   - ✓ CheckBox
   - ✓ ComboBox
   - ✓ ListBox
   - ✓ CheckBoxList
   - ✓ TextArea
8. Thay đổi kiểu dáng của các thành phần web
   - ✓ Màu nền các hàng của bản (thường và hover)
   - ✓ Ảnh nền và đường kẻ cho các vùng
   - ✓ Nội dung các thông báo hay tùy biến
9. Dựa vào focus và blur để tạo hiệu ứng Stringfall
10. Dựa vào sự kiện hover để tạo tín hiệu tích cực cho các thành phần trên web
11. Ẩn hiện các block
12. Ẩn hiện form đăng nhập và thông tin tài khoản dưới dạng menu
13. Bỏ hàng vào giỏ
14. Định nghĩa class error cho thẻ input
15. Kiểm tra form đăng nhập, đổi mật khẩu
16. Tạo một form nhập 2 bước và kiểm lỗi các ô nhập trên form như sau.



Hình 6.25: Kiểm lỗi form đăng ký

17. Sử dụng Dialog cho các trường hợp

- ✓ Thông báo lỗi, xác nhận lỗi
- ✓ Đăng nhập nhanh
- ✓ Gửi mail cho bạn bè

18. Sử dụng Accordion để làm menu đứng
19. Sử dụng tab để xây dựng CRUD 2 giao diện. Tab 1 dành cho edition (thêm, sử, xóa) và Tab 2 dành cho seacrh (tìm kiếm, hiển thị).
20. Sử dụng Datepicker cho tất cả các ô nhập ngày
21. Tab ajax
22. Tải các liên kết bằng ajax
23. Kiểm tra tính hợp lệ thông qua ajax
    - ✓ Login
    - ✓ Captcha
24. Tìm kiếm nhanh dựa vào sự kiện keyup