

Đặt vấn đề



1. Bạn có biết hết tất các thẻ HTML hay không?
2. Bạn có thể điều khiển được tất các thẻ HTML hay không?

Các thẻ HTML thông dụng (p1)



STT	Thẻ HTML	Miêu tả	Kiểu Tag
1	<html>	Thẻ mở đầu của trang HTML	none
2	<head> </head>	Thẻ chứa các thẻ trong phần đầu của trang HTML	none
3	<title></title>	Tiêu đề trang web	none
4	<meta>	Mô tả tổng quát về nội dung trang web	none
5	<link>	Dùng để nhúng một tập tin nào đó vào trang web.	none
6	<script></script>	Dùng để nhúng các tập tin javascript	none
7	<style></style>	Dùng để bao bọc một nội dung về css	none
8	<body></body>	Thẻ chứa nội dung chính của website	block level
9	<h1> </h1>	Thẻ để thể hiện tiêu đề của một vấn đề nào đó	block level
10	<div></div>	Thẻ này là thẻ thường dùng để chứa nội dung	block level

Các thẻ HTML thông dụng (p2)



STT	Thẻ HTML	Miêu tả	Kiểu Tag
11		Thẻ chứa nội dung	inline
12	<p></p>	Thẻ chứa nội dung (đoạn văn)	block level
13	<center></center>	Thẻ canh giữa các đối tượng nằm bên trong	block level
14	<a>	Thẻ tạo link	inline
15		Kết hợp với thẻ để mô tả liệt kê theo dạng danh sách	block level
16		Thẻ dùng để hiển thị một hình ảnh nào đó	inline
17	<form> <input>	Thẻ hiển thị những phần tử trong form nhập liệu	block level
18	 	Thẻ xuống hàng	block level
19	<hr>	Thẻ tạo đường kẻ ngang	block level
20	<table></table>	Tạo bảng	block level

jQuery Master

zebra

Các thẻ HTML thông dụng (p3)



STT	Thẻ HTML	Miêu tả	Kiểu Tag
21	<iframe></iframe>	Tạo frame	block level
22		Tạo chữ đậm	inline
23	<i></i>	Tạo chữ nghiêng	inline
24	<u></u>	Tạo chữ gạch dưới	inline
25	<s></s>	Tạo chữ gạch cắt ngang	inline
26	 	Tạo kiểu chữ	inline
27	<blockquote> </blockquote>	Mô tả một phần của trích dẫn	block level
28	<tt></tt> <code></code>	Tạo kiểu chữ cho phần mô tả mã nguồn	block level
29	<pre></pre>	Định dạng nội dung	block level

Phân loại thẻ HTML

- None: Khối này không hiển thị nội dung bên trong
 - Block level: Khối này hiển thị nội dung bên trong và chiếm tràn hết trình duyệt
 - Inline: Khối này hiển thị nội dung bên trong và chiều ngang tùy thuộc độ dài của các đối tượng bên trong khối. Và nó sẽ nằm trên một dòng
- ➔ Các thẻ HTML trong cặp thẻ `<body></body>` thường là kiểu block và inline.

Phân nhóm định dạng

1. Type group: định dạng cho văn bản
2. Background group: định dạng hình ảnh nền cho một đối tượng nào đó
3. Block group: định dạng cho văn bản
4. Border group: định dạng đường viền cho một đối tượng nào đó
5. Box group: định dạng kích thước, vị trí cho khối
6. List group: định dạng cho các danh sách
7. Position group: định tọa độ của một phần tử HTML nào đó

01 – Type group

STT	Nhóm thuộc tính	Miêu tả
1	font-family	Nhóm font được sử dụng cho một đối tượng HTML
2	font-size	Kích thước của văn bản
3	font-style	Định kiểu cho font chữ nghiêng hay thẳng
4	font-variant	Định kiểu cho font chữ thường hoặc chữ hoa
5	font-weight	Kiểu của chữ
6	line-height	Chiều cao giữa các dòng của văn bản
7	text-transform	Kiểu hiển thị của font chữ trong văn bản
8	text-decoration	Kiểu hiển thị của font chữ trong văn bản
9	color	Màu sắc của văn bản

02 – Background group

STT	Nhóm thuộc tính	Miêu tả
1	background-color	Màu nền của đối tượng HTML
2	background-image	Sử dụng nền là một hình ảnh
3	background-repeat	Kiểu hiển thị hình nền nếu sử dụng ảnh làm nền cho đối tượng
4	background-position	Vị trí bắt đầu hiển thị của hình nền
5	background-attachment	Chế độ cố định hình nền

03 – Block group

STT	Nhóm thuộc tính	Miêu tả
1	letter-spacing	Khoảng cách giữ các ký tự
2	word-spacing	Khoảng cách giữa các từ trong đoạn văn bản
3	text-align	Vị trí của của đoạn văn bản
4	text-indent	Khoảng cách thụt vào đầu dòng của một đoạn văn bản
5	white-space	Định dạng cho khoảng trắng trong đoạn văn bản
6	vertical-align	Vị trí của một phần tử
7	display	Các kiểu hiện thị theo kiểu block, inline...

04 – Border group

STT	Nhóm thuộc tính	Miêu tả
1	border-width border-top-width, border-right-width border-bottom-width, border-left-width	Độ rộng của đường viền
2	border-style border-top-style, border-right-style border-bottom-style, border-left-style	Kiểu của đường viền
3	border-color border-top-color, border-right-color border-bottom-color, border-left-color	Màu sắc của đường viền

05 – Box group

STT	Nhóm thuộc tính	Miêu tả
1	width min-width, max-width	Chiều rộng của đối tượng
2	height min-height, max-height	Chiều cao của đối tượng
3	margin margin-top, margin-right, margin-bottom, margin-left	Khoảng cách đối với phần tử bên ngoài
4	padding padding-top, padding-right, padding-bottom, padding-left	Khoảng cách đối với phần tử bên trong
5	float	Lệch khối về bên trái hoặc phải
6	clear	Xóa các thuộc tính float ở các phần tử phía trên

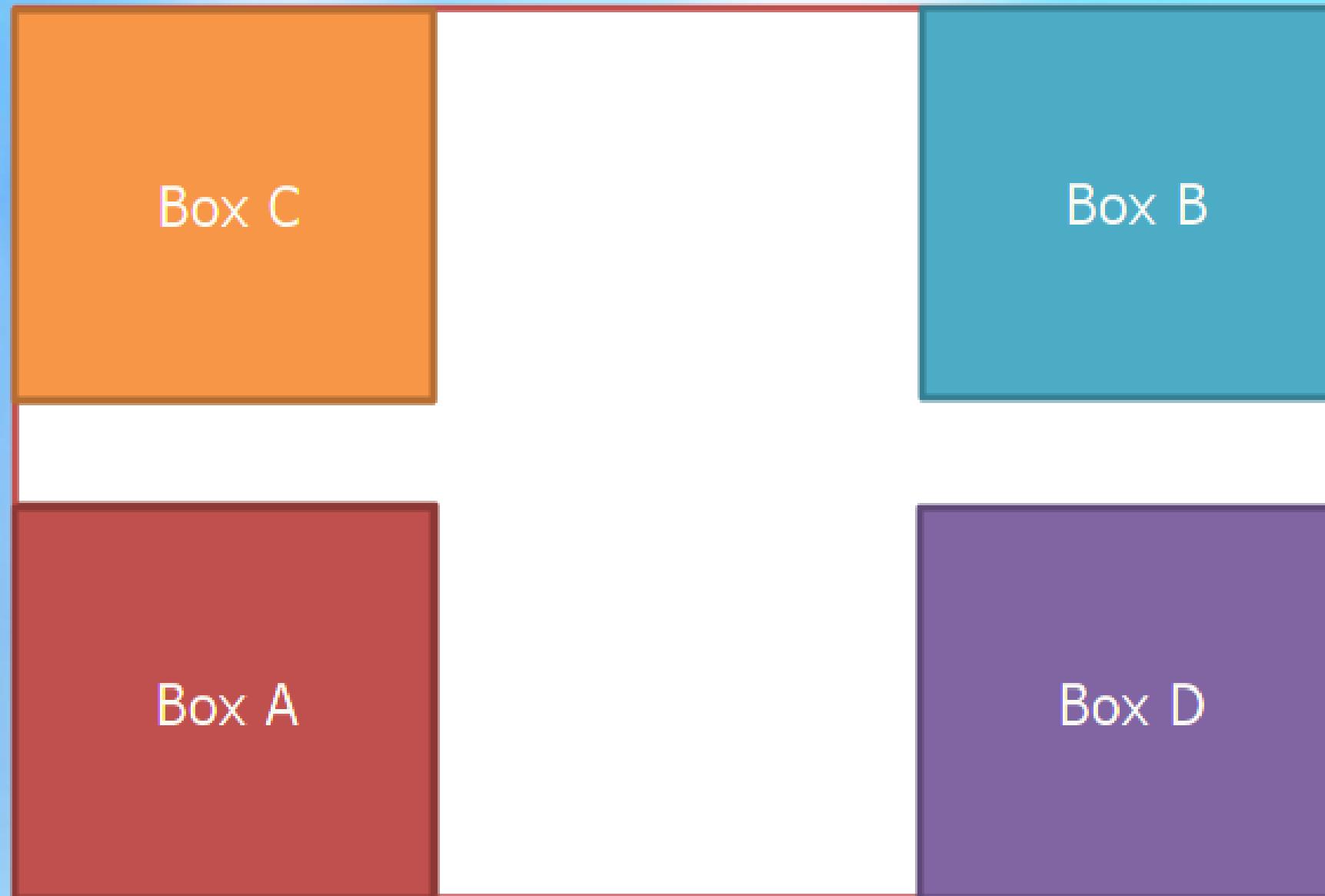
06 – List group

STT	Nhóm thuộc tính	Miêu tả
1	list-style-position	Vị trí của icon . Giá trị mặc định là outside
2	list-style-type	Kiểu icon của
3	list-style-image	Hình ảnh icon của của

07 – Position group

STT	Nhóm thuộc tính	Miêu tả
1	position	Kiểu hiển thị của một đối tượng
2	top	Khoảng cách từ đối tượng đến vị trí top
3	right	Khoảng cách từ đối tượng đến vị trí right
4	bottom	Khoảng cách từ đối tượng đến vị trí bottom
5	left	Khoảng cách từ đối tượng đến vị trí left
6	z-index	Vị trí của đối tượng
7	overflow overflow-x, overflow-y	Chế độ hiển thị thanh cuộn

Ví dụ 1: Sử dụng giá trị relative, absolute của thuộc tính position



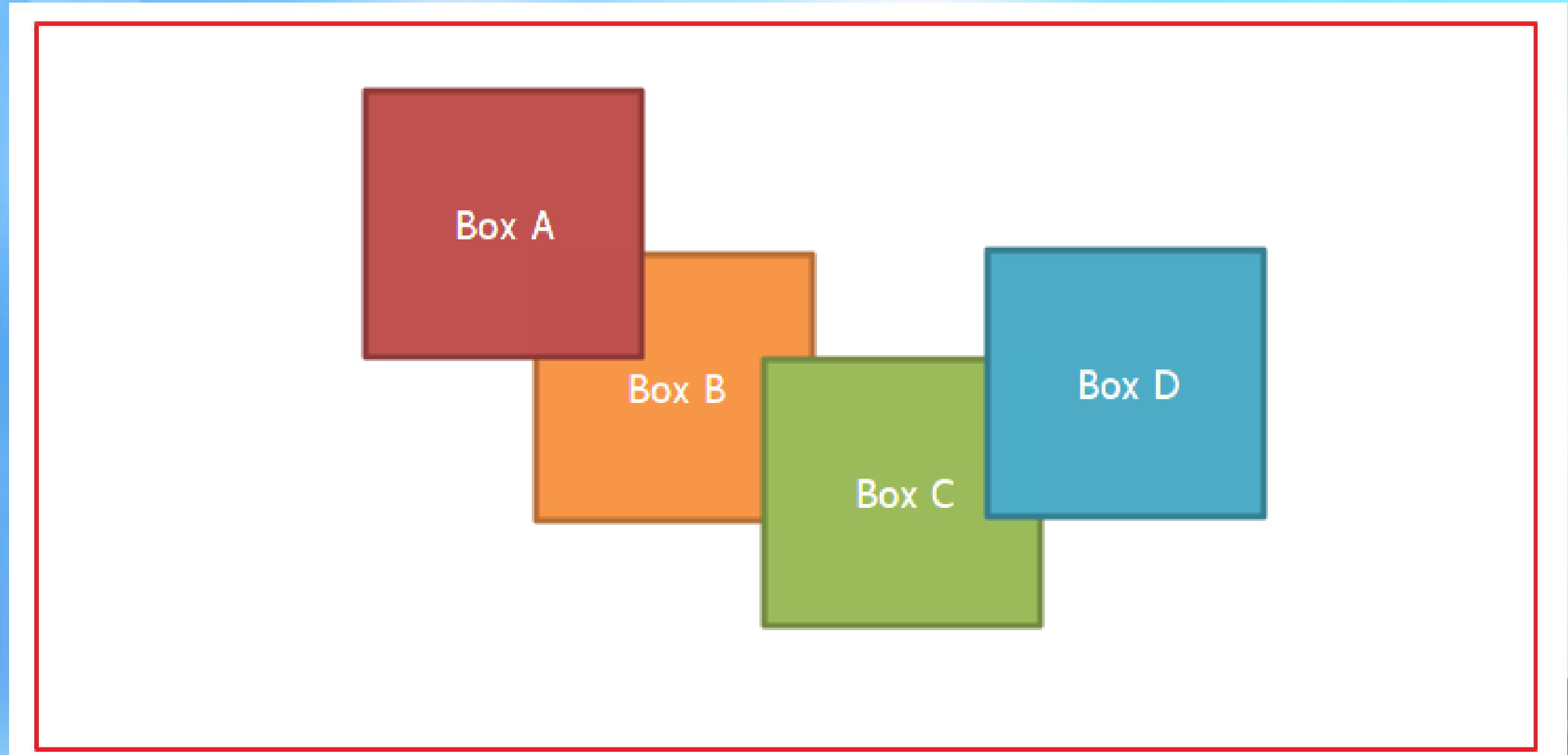
Ví dụ 2: Sử dụng giá trị relative, fix của thuộc tính position



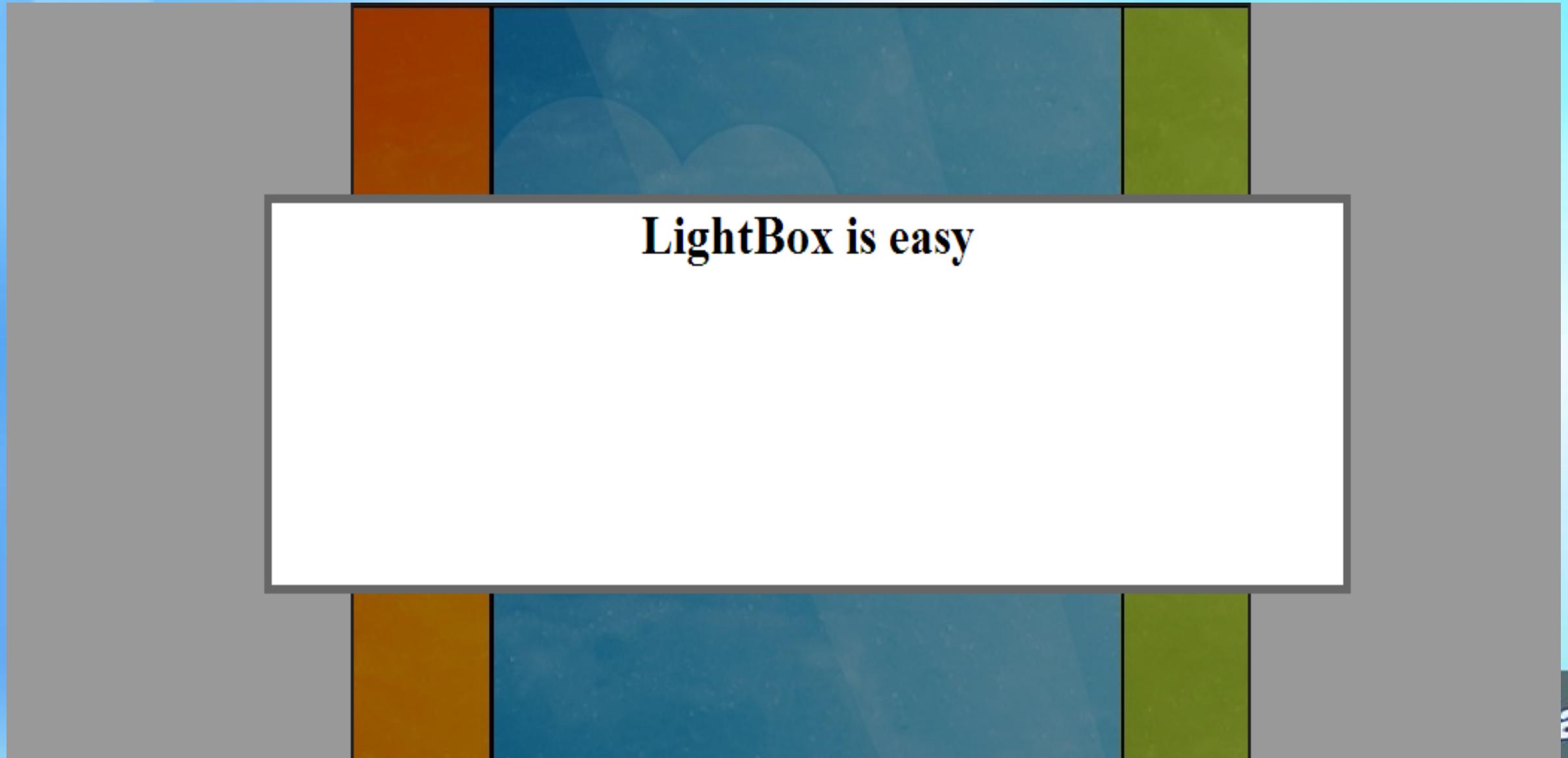
Ví dụ 3: Sử dụng giá trị fix của thuộc tính position để tạo menu dọc trình duyệt



Ví dụ 4: Sử dụng giá trị relative, absolute của thuộc tính position và giá trị của z-index



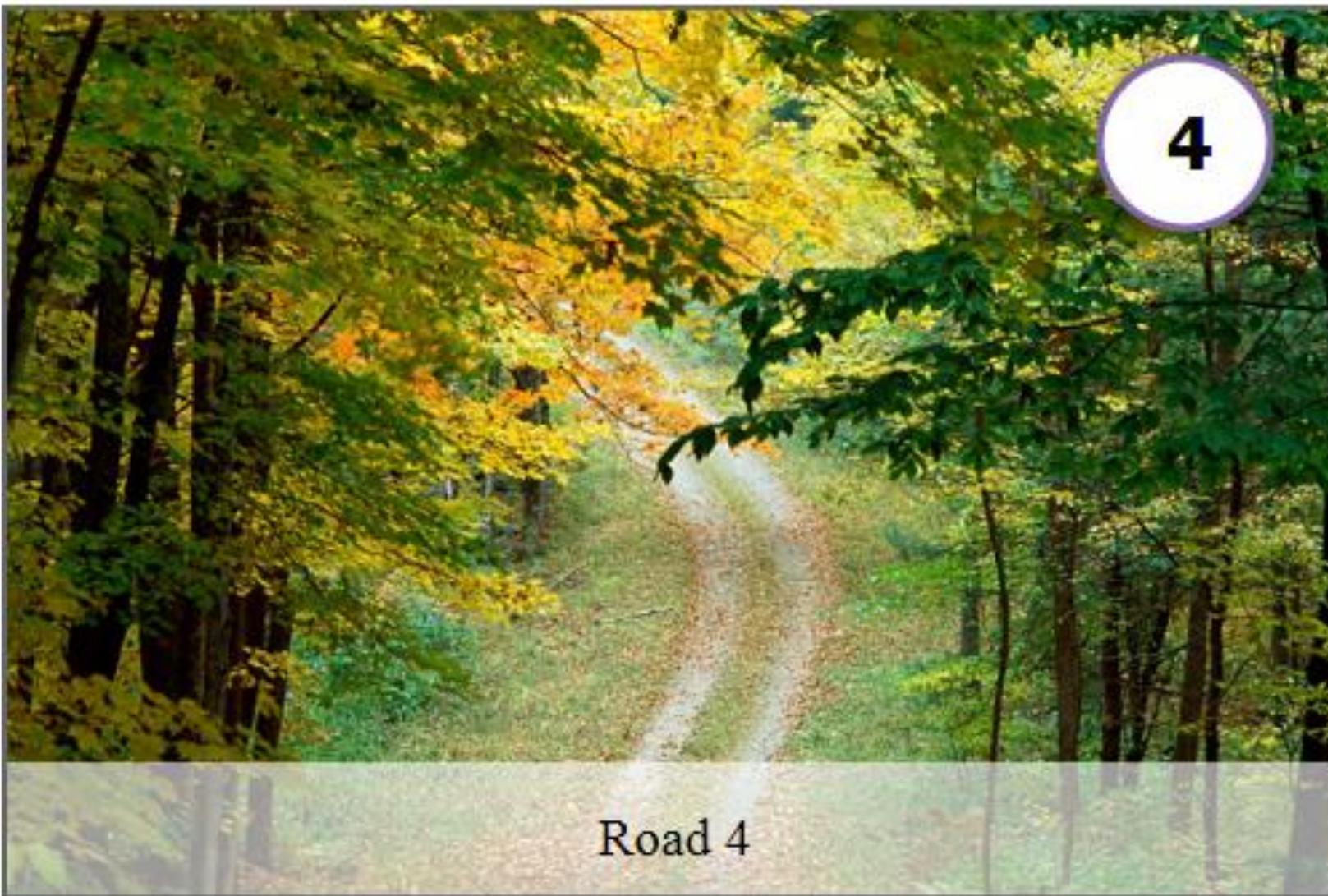
Ví dụ 5: LightBox is easy



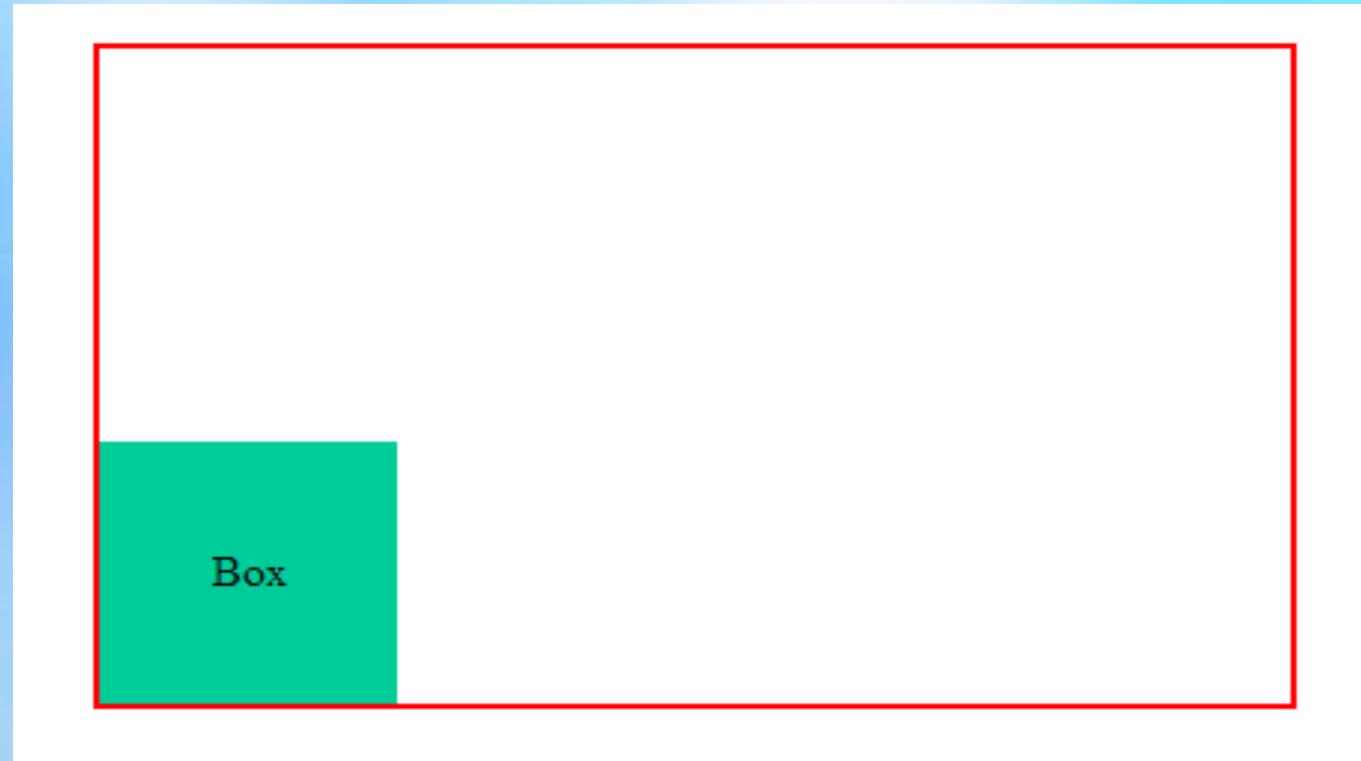
Ví dụ 6: Slide



Ví dụ 6: Slide ảnh (cơ bản)



Sự phức tạp của các giá trị position



Trong hình ảnh trên, có một phần tử HTML (tạm gọi là khối HTML) chứa một hình vuông có tên box. Bây giờ chúng ta sẽ tìm hiểu sự tương tác giữa giá trị của khối HTML và ô vuông có tên box.

TH1: Khối HTML có giá trị position = static

	static	relative	absolute	fixed
top	None	Active	Active (b)	Active (b)
right	None	Active (r1)	Active (b)	Active (b)
bottom	None	Active (r2)	Active (b)	Active (b)
left	None	Active	Active (b)	Active (b)

- r1: Lệch về bên phải so với vị trí gốc
- r2: Lệch lên phía trên so với vị trí gốc
- b: so với trình duyệt (browser)

TH2: Khối HTML có giá trị position = relative

	static	relative	absolute	fixed
top	None	Active	Active (o)	Active (b)
right	None	Active (r1)	Active (o)	Active (b)
bottom	None	Active (r2)	Active (o)	Active (b)
left	None	Active	Active (o)	Active (b)

- r1: Lệch về bên phải so với vị trí gốc
- r2: Lệch lên phía trên so với vị trí gốc
- b: so với trình duyệt (browser)
- o: so với khối HTML

TH3: Khối HTML có giá trị position = absolute

	static	relative	absolute	fixed
top	None	Active	Active (o)	Active (b)
right	None	Active (r1)	Active (o)	Active (b)
bottom	None	Active (r2)	Active (o)	Active (b)
left	None	Active	Active (o)	Active (b)

- r1: Lệch về bên phải so với vị trí gốc
- r2: Lệch lên phía trên so với vị trí gốc
- b: so với trình duyệt (browser)
- o: so với khối HTML

TH4: Khối HTML có giá trị position = fixed

	static	relative	absolute	fixed
top	None	Active	Active (o)	Active (b)
right	None	Active (r1)	Active (o)	Active (b)
bottom	None	Active (r2)	Active (o)	Active (b)
left	None	Active	Active (o)	Active (b)

- r1: Lệch về bên phải so với vị trí gốc
- r2: Lệch lên phía trên so với vị trí gốc
- b: so với trình duyệt (browser)
- o: so với khối HTML

1. Khối HTML có giá trị position = static

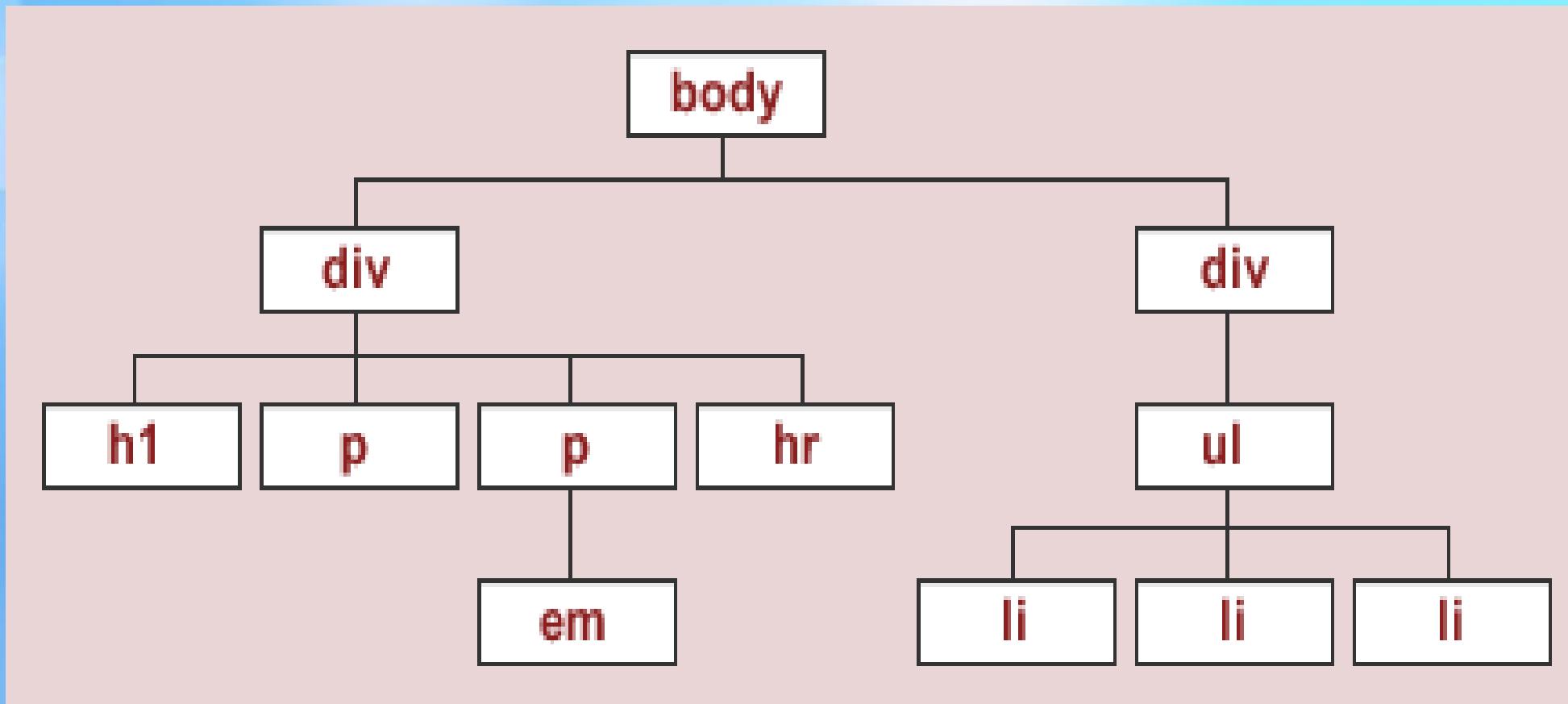
	static	relative	absolute	fixed
top	None	Active	Active (b)	Active (b)
right	None	Active (r1)	Active (b)	Active (b)
bottom	None	Active (r2)	Active (b)	Active (b)
left	None	Active	Active (b)	Active (b)

2. Khối HTML có giá trị position = fixed, relative, absolute

	static	relative	absolute	fixed
top	None	Active	Active (o)	Active (b)
right	None	Active (r1)	Active (o)	Active (b)
bottom	None	Active (r2)	Active (o)	Active (b)
left	None	Active	Active (o)	Active (b)

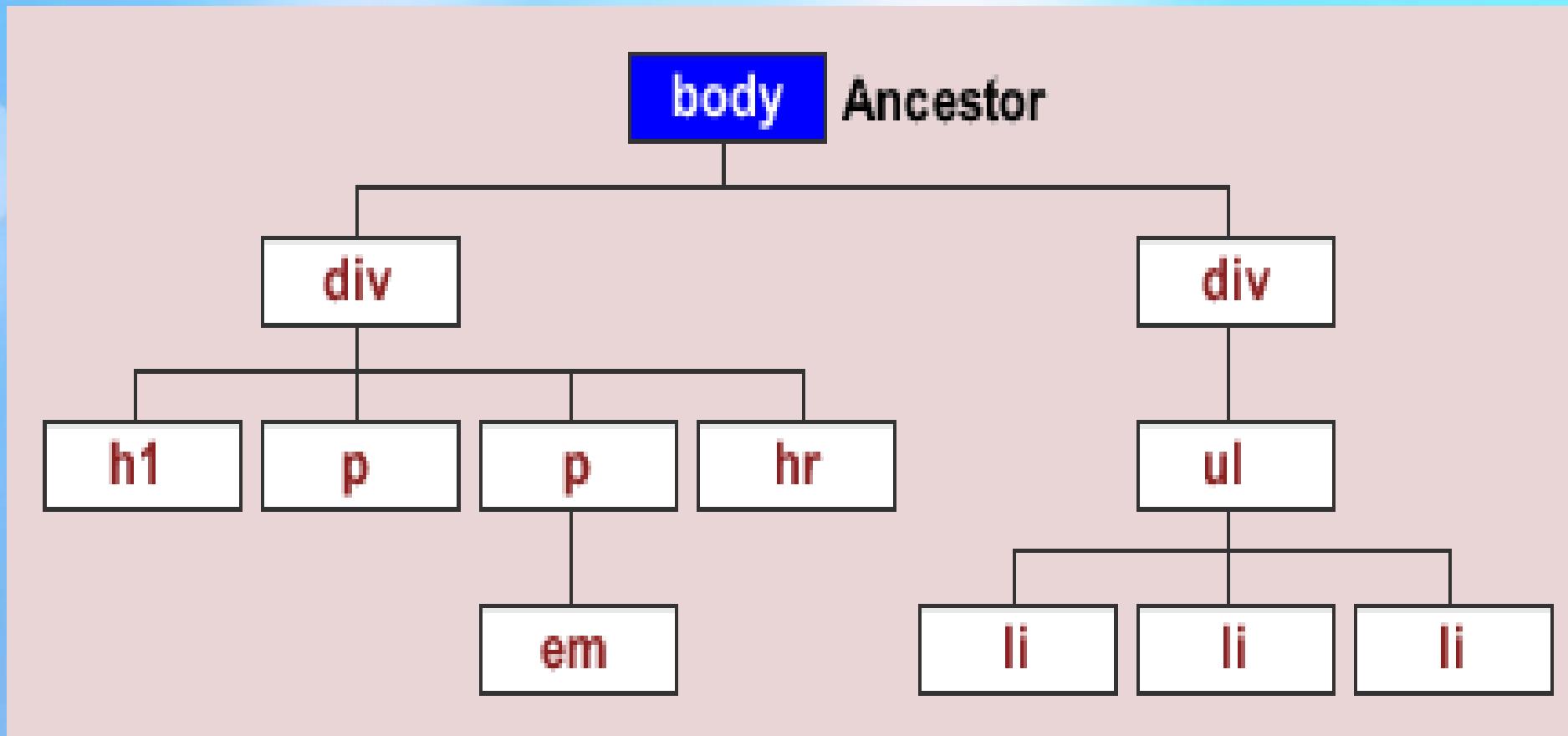
Các khái niệm về Selector

Giới thiệu



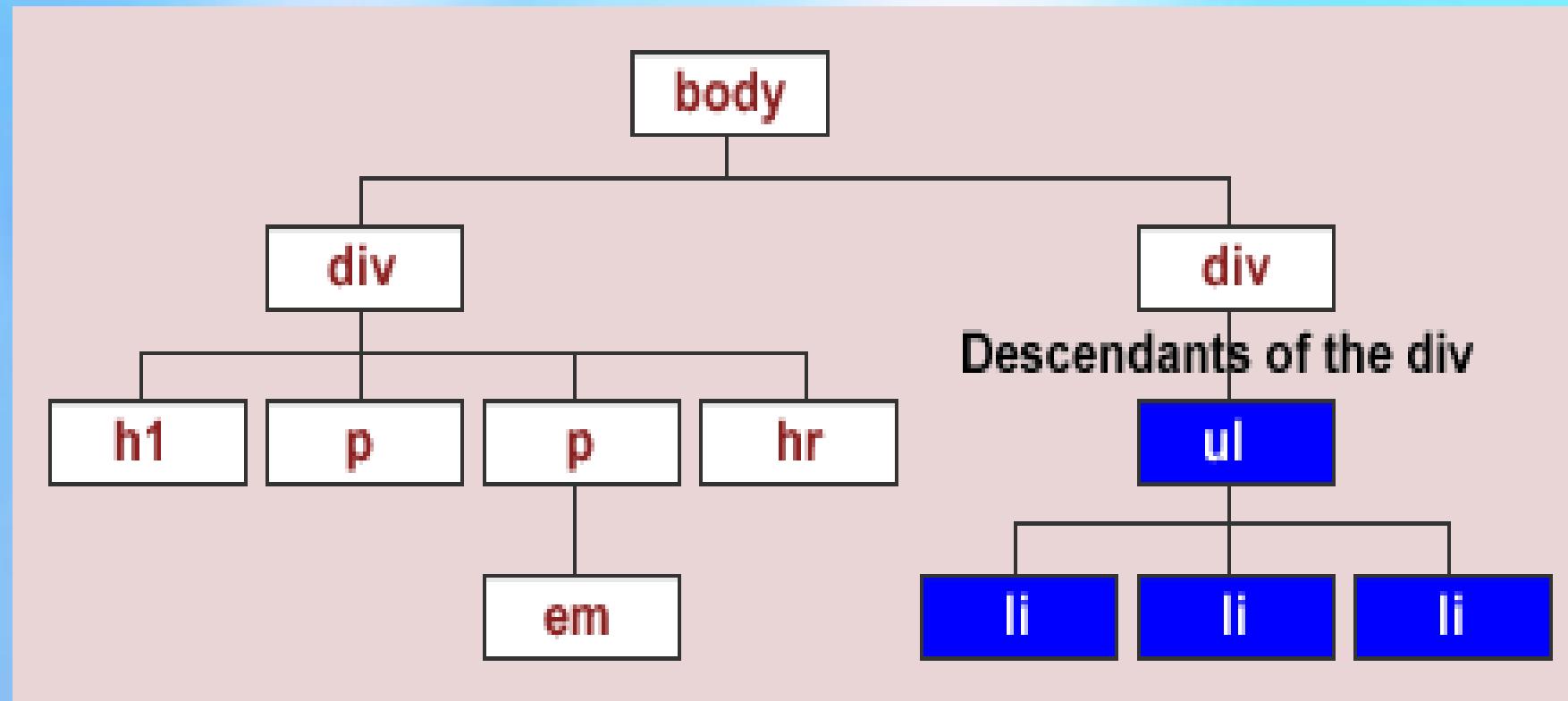
- Tài liệu HTML là một dạng tài liệu bao gồm nhiều thẻ và nó được sắp xếp theo dạng cây

Ancestor (tổ tiên – nút gốc)



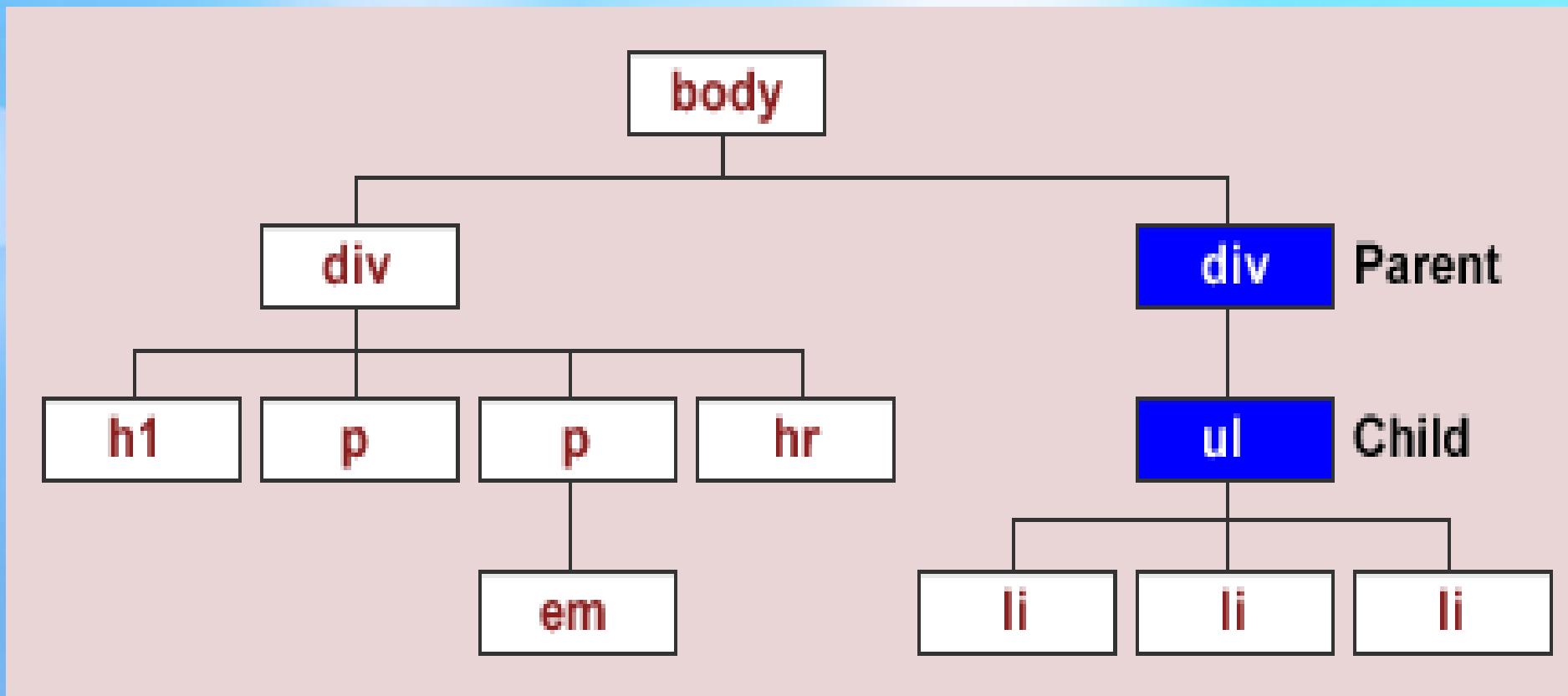
- Trong tài liệu HTML thẻ <body> được gọi là Ancestor.

Descendant (con cháu)



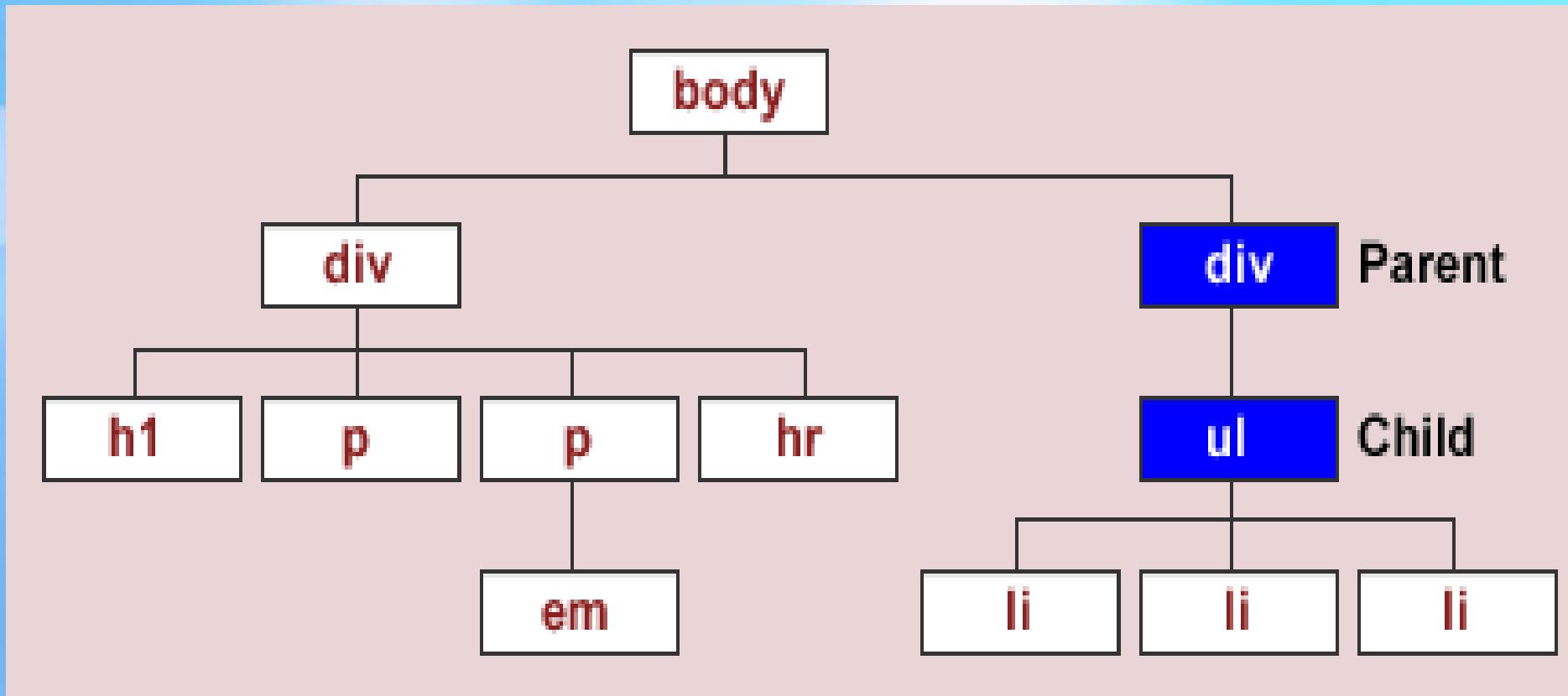
- Các thẻ HTML nằm trong thẻ <body> được gọi là các thẻ con cháu
- Các thẻ <div>, <h1>, <p>... được gọi là cá descendants của thẻ <body>
- Các thẻ , ... được gọi là các descendants của thẻ <div>

Parent (Cha)



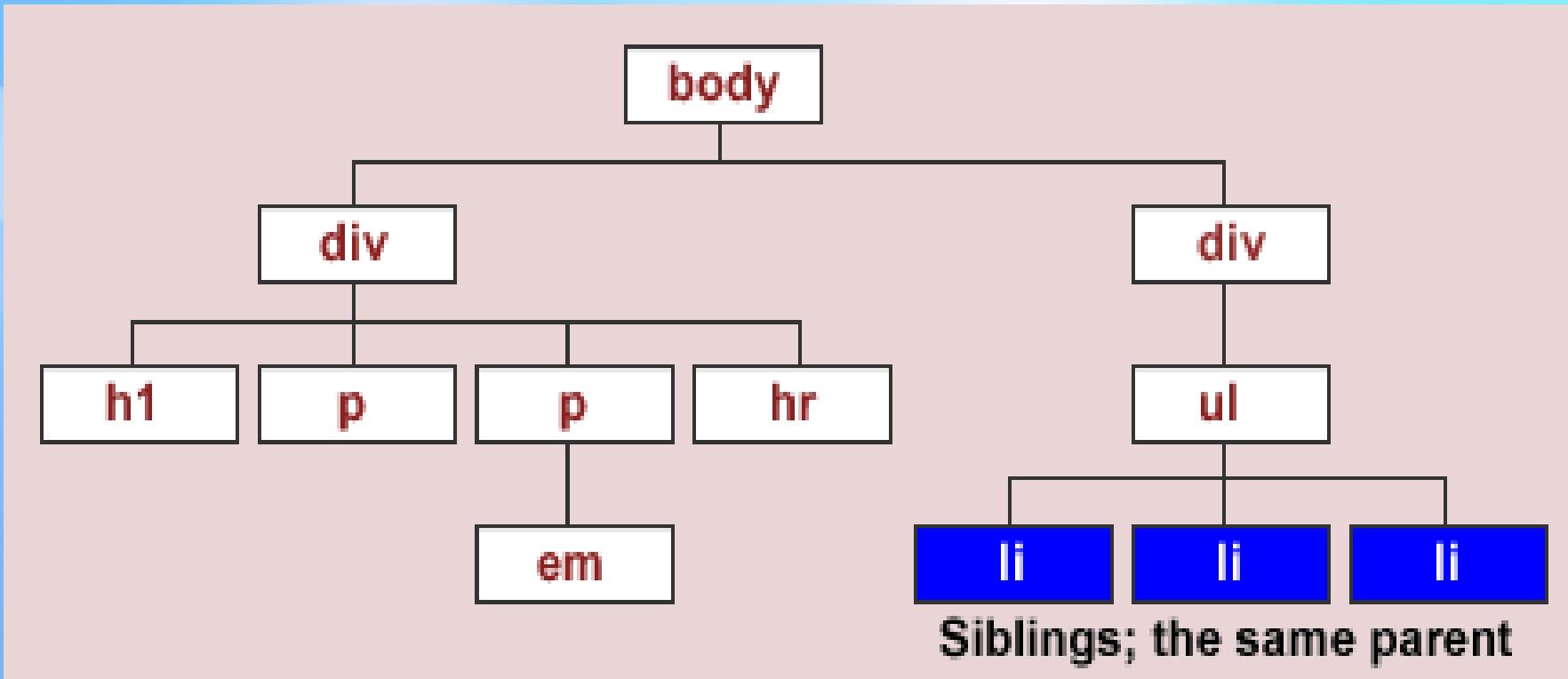
- Thẻ `<div>` được gọi là thẻ cha của thẻ ``
- Thẻ `<div>` được gọi là thẻ cha của thẻ `<p>`
- ...

Child (Con)



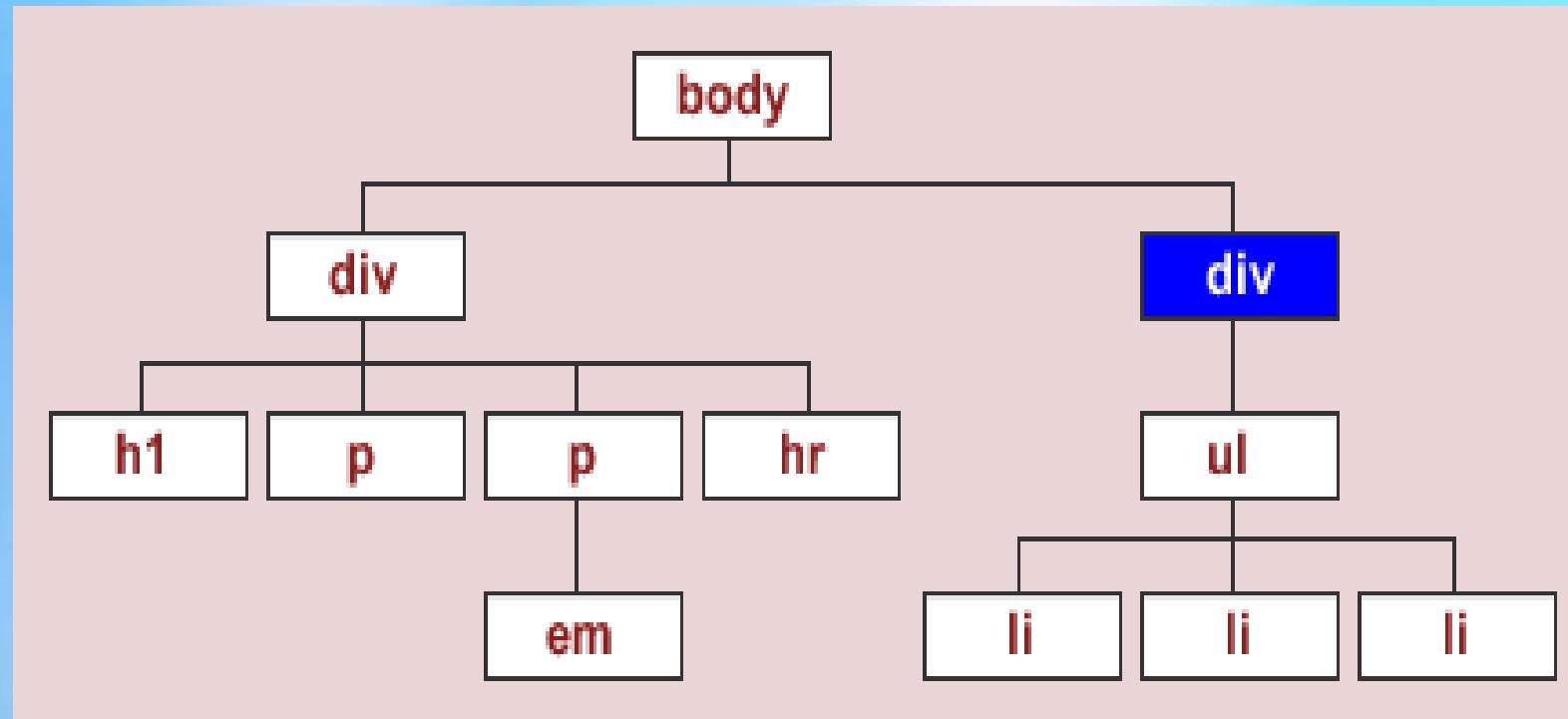
- Thẻ được gọi là thẻ con của thẻ <div>
- Thẻ <p> được gọi là thẻ con của thẻ <div>
- ...

Sibling (anh em)



- Các thẻ có cùng cha được gọi là anh em → Các thẻ **** là thẻ anh em của nhau, ...

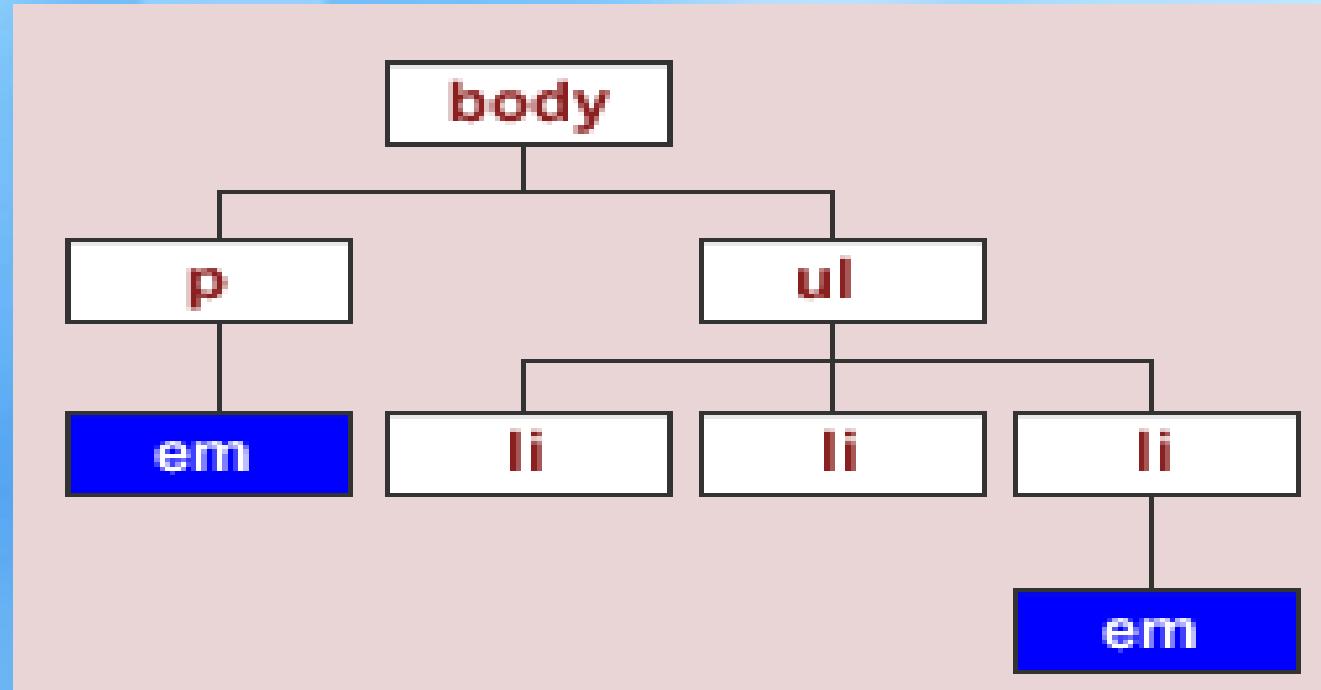
Vị trí của một phần tử HTML



- Vị trí của thẻ div: con của phần tử `<body>`; cha của phần tử ``; ancestor của ``, ``; anh em của phần tử `<div>` bên tay trái

Kết hợp Selector trong CSS

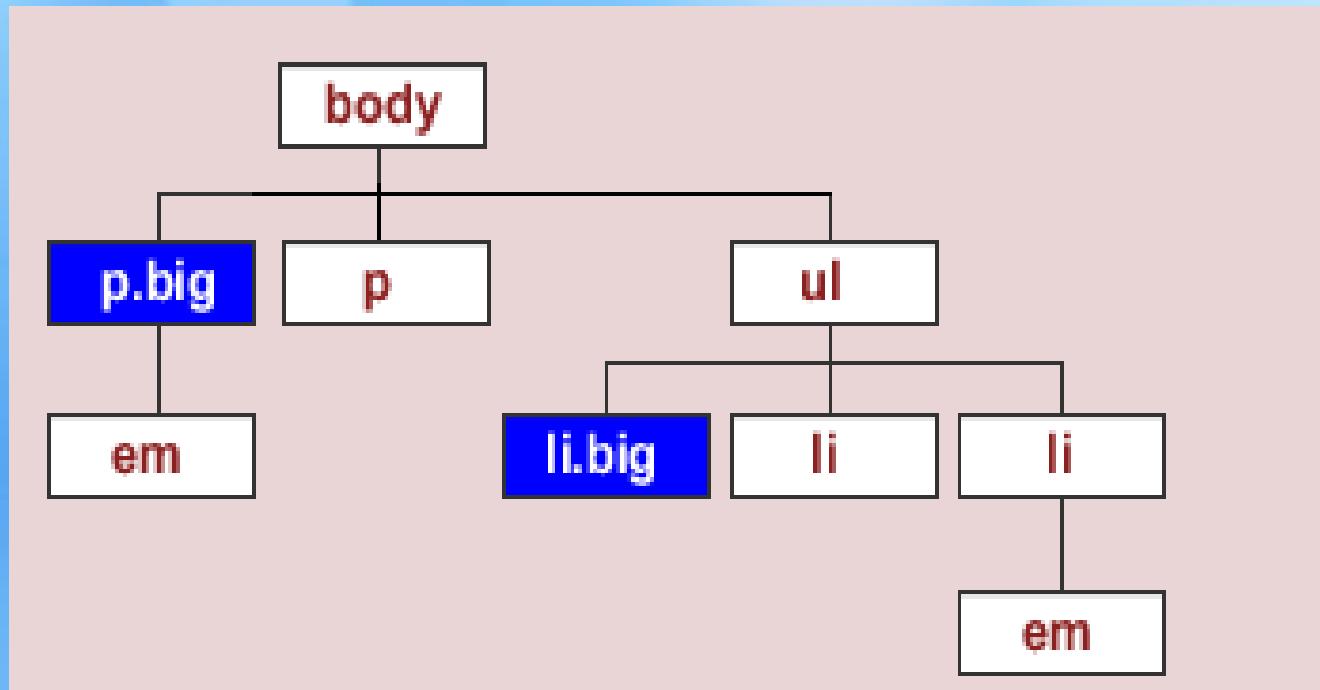
Type selectors



- Để định dạng cho các phần tử (thẻ) bất kỳ của HTML

```
em {color: blue;}
```

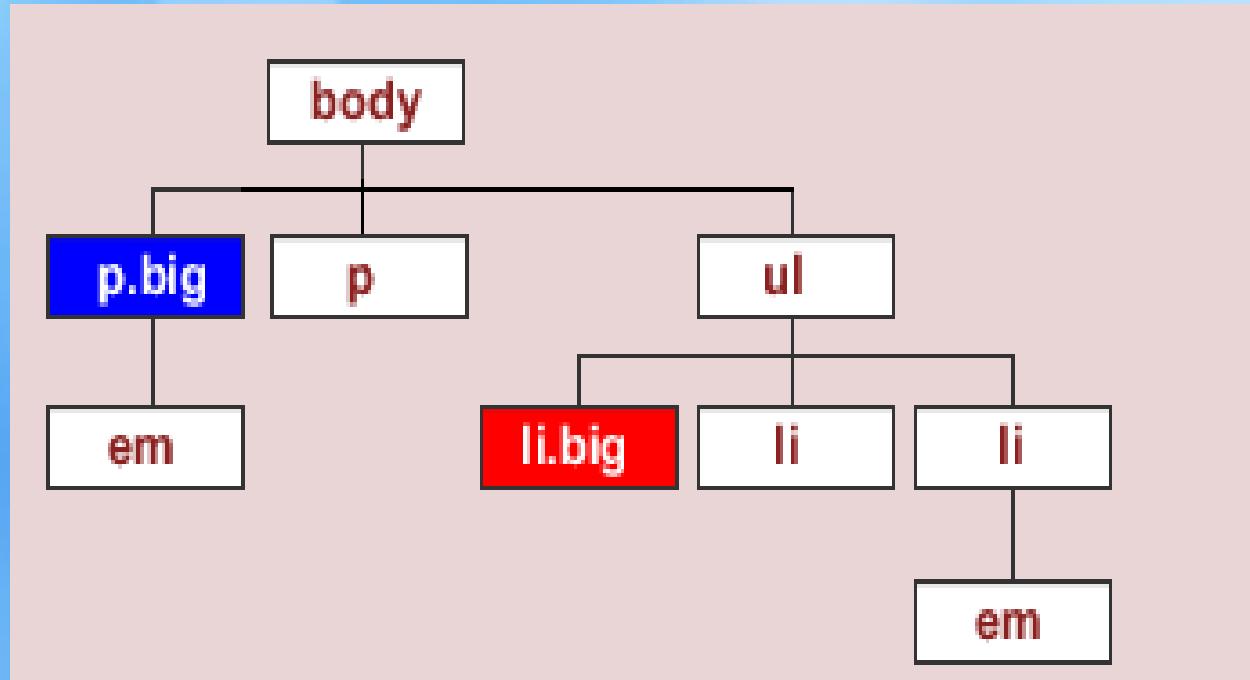
Class selectors



- Định dạng cho các phần tử (thẻ) bất kỳ bằng thuộc tính “class” của selector

```
.big { font-size: 110%; font-weight: bold; }
```

Kết hợp “class” và “type”



- Cú pháp: <tag_name>.<class_name>

```
.big { color: red; } // affects <p> and <li>  
p.big { color: blue; } // affects <p> only
```

Kết hợp nhiều “class”

- HTML

```
<p class="big indent">
```

- CSS

```
.big      { font-weight: bold; }  
.indent  { padding-left: 2em; }
```

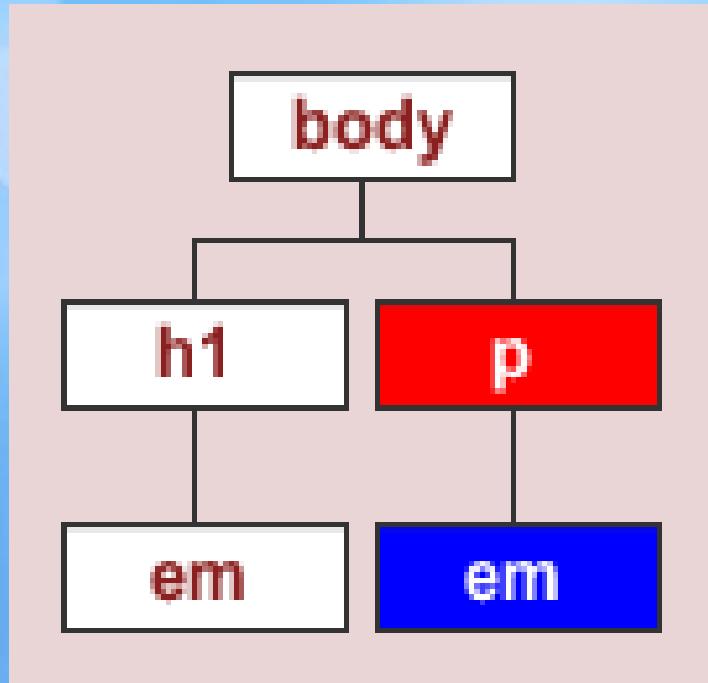
- Chú ý thứ tự khai báo các lớp trong file ?

ID selectors

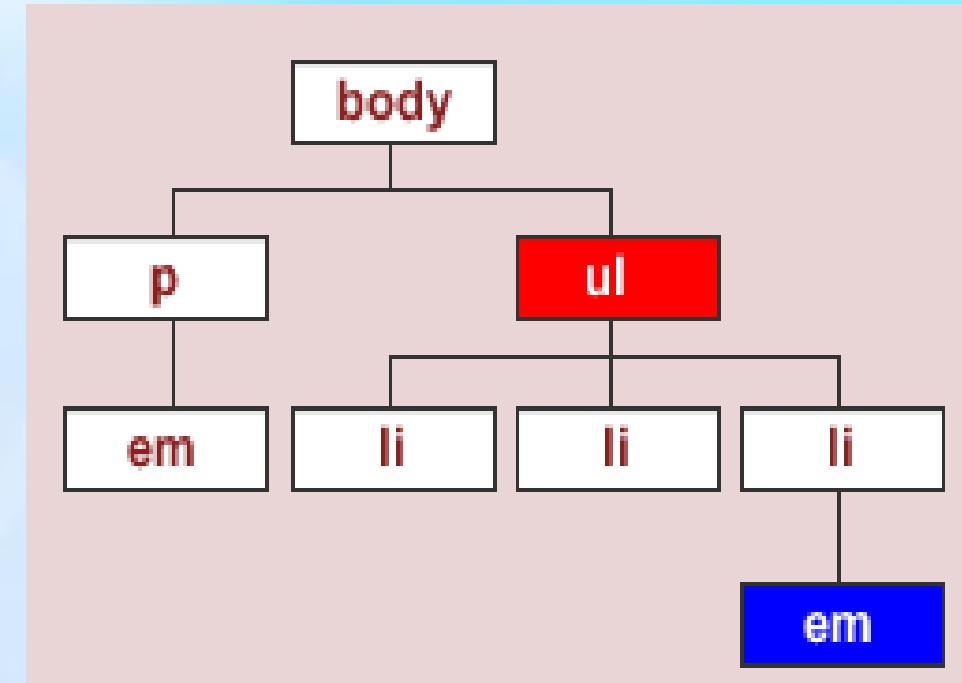
- Để khai báo cho ID chúng ta sử dụng kí hiệu # ở trong phần css.
- Tên của ID chỉ được sử dụng một lần trên mỗi webpage. Tên class có thể sử dụng nhiều lần trên một webpage

```
#big { font-size: 110%; font-weight: bold; }
```

Descendant selectors (selector phía trong)

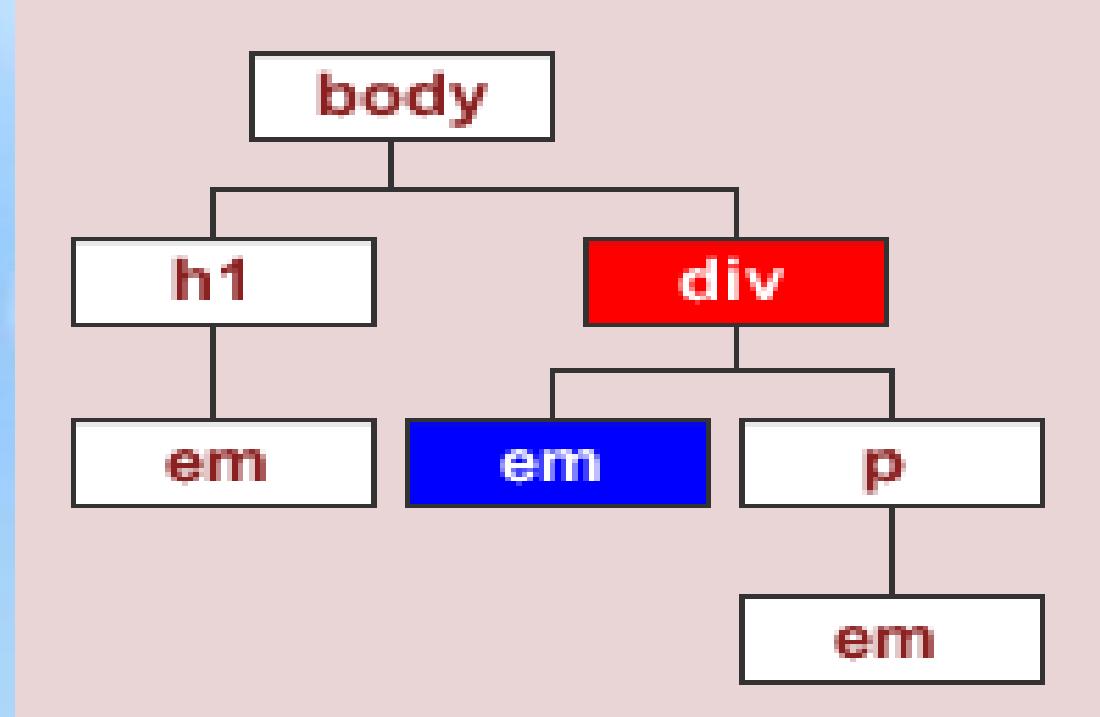


```
em {color: blue; }  
p em {color: blue; }
```



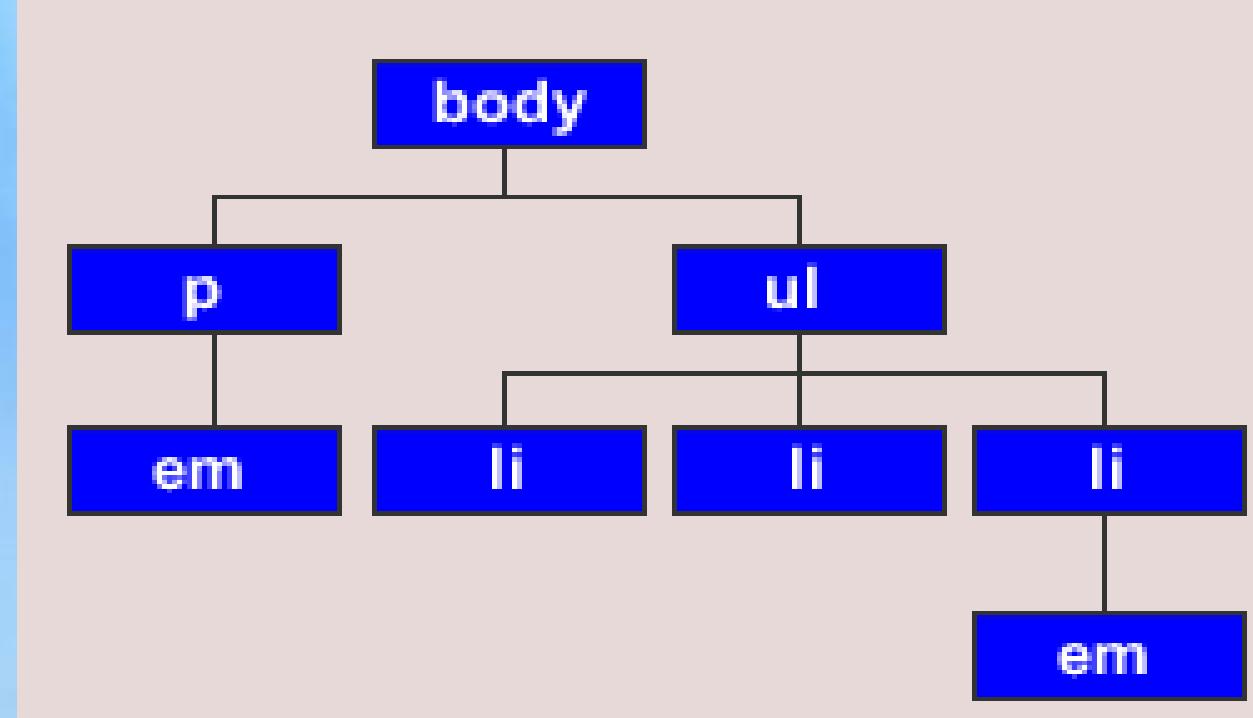
```
ul em {color: blue; }
```

Child selectors (Selector con)



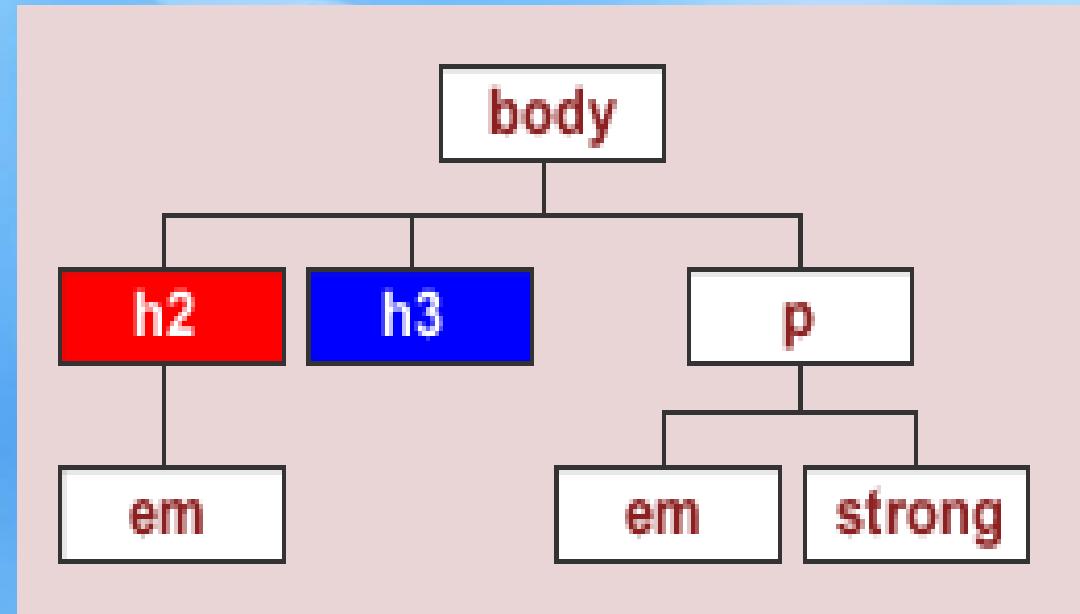
```
div > em { color: blue; }  
div>em { color: blue; }
```

Universal selectors (Toàn bộ selector)

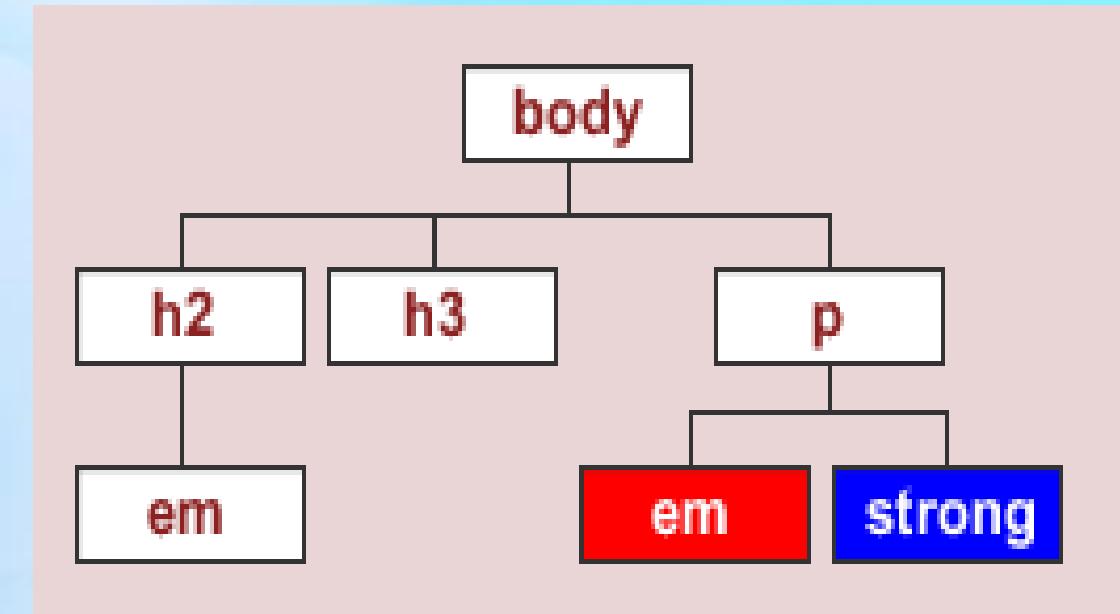


```
* {color: blue; }
```

Adjacent sibling selectors (Những selector cùng cấp kế bên)



```
h2 + h3 {color: blue;}
```



```
em + strong {color: blue;}
```

Attribute selectors

- Dạng 1: theo tên thuộc tính

```
[title] { border: 3px solid red; }  
img[width] { border: 3px solid red; }
```

- Dạng 2: theo tên thuộc tính và giá trị của thuộc tính đó

```
img[src="small.gif"] { border: 3px solid red; }  
img[title~="small"] { border: 3px solid red; }  
img[title|= "small"] { border: 3px solid red; }  
img[title*= "small"] { border: 3px solid red; }  
img[title^= "small"] { border: 3px solid red; }  
img[title$= "small"] { border: 3px solid red; }  
img[src="small.gif"][title~="small"] { border: 3px solid red; }
```

Pseudo Classes/Element

STT	Selector	Miêu tả
1	:link, :visited, :active, :hover	Thao tác với các liên kết
2	:focus	Thao tác với các đối tượng khi nó ở “focus”
3	:lang(language)	Tất cả các đối tượng với thuộc tính lang
4	:first-letter	Ký tự đầu tiên của một đối tượng nào đó
5	:first-line	Dòng đầu tiên của một đối tượng nào đó
6	:first-child	Con đầu tiên của một đối tượng nào đó
7	:before	Chèn nội dung vào trước đối tượng nào đó
8	:after	Chèn nội dung vào sau đối tượng nào đó

Xây dựng giao diện mẫu cho button – Dạng 1

Hello World

Xây dựng giao diện mẫu cho button – Dạng 2



Xây dựng giao diện mẫu cho button – Dạng 3



Hello World

Xây dựng giao diện mẫu cho box

Box chứa nội dung – Dạng 01

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Xây dựng giao diện mẫu cho box

Box chứa nội dung – Dạng 02

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. **Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.** Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Xây dựng giao diện mẫu cho box

Box chứa nội dung – Dạng 03

Xây dựng giao diện mẫu cho box

Box chứa nội dung (có tiêu đề) – Dạng 01

Title

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Xây dựng giao diện mẫu cho box

Box chứa nội dung (có tiêu đề) – Dạng 2

■ *Title*

 Lorem ipsum dolor sit amet, consectetur
 adipisicing elit, sed do eiusmod tempor incididunt
 ut labore et dolore magna aliqua. Ut enim ad minim
 veniam, quis nostrud exercitation ullamco laboris
 nisi ut aliquip ex ea commodo consequat. Duis
 aute irure dolor in reprehenderit in voluptate velit
 esse cillum dolore eu fugiat nulla pariatur.

Xây dựng giao diện mẫu cho box

Box chứa nội dung (có tiêu đề) – Dạng 3

■ **Title**

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Xây dựng giao diện mẫu cho box

Box chứa nội dung (có tiêu đề) – Dạng 4

■ **Title**

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Xây dựng giao diện mẫu cho box

Box chứa nội dung (có tiêu đề) – Dạng 5

■ Title

Consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. **Consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.** Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Xây dựng giao diện mẫu cho box

Box chứa nội dung (có tiêu đề) – Dạng 6

■ *Title*

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. **L**orem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Xây dựng giao diện mẫu cho box

Box bo tròn chứa nội dung – Dạng 1

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

ter

Xây dựng giao diện mẫu cho box

Box bo tròn chứa nội dung – Dạng 2

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

ter

Xây dựng giao diện mẫu cho box

Box bo tròn chứa nội dung và tiêu đề – Dạng 1

Title

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

■ Title

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Xây dựng giao diện mẫu cho box

Box bo tròn chứa nội dung và tiêu đề – Dạng 2

■ **Title**

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

■ **Title**

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Xây dựng giao diện mẫu cho box

Box bo tròn chứa nội dung và tiêu đề – Dạng 3

■ **Title**

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

■ **Title**

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Xây dựng giao diện mẫu cho tab

Dạng 1

PHP Script

ASP Script

JSP Script

PHP script. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Xây dựng giao diện mẫu cho tab

Dạng 2

PHP Script

ASP Script

JSP Script

ASP script. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Xây dựng giao diện mẫu cho tab

Dạng 3

PHP Script

ASP Script

JSP Script

ASP script. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Xây dựng giao diện mẫu cho tab

Dạng 4

 PHP script

 ASP script

ASP script. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

 JSP script

Xây dựng giao diện mẫu cho menu

Dạng 1

 <u>Home</u>		
 <u>About</u>		
 <u>News</u>	 <u>Tin trong nuoc</u>	
 <u>Products</u>	 <u>Tin quoc te</u>	 <u>Cong nghe</u>
 <u>Contact</u>	 <u>Tin trong nha</u>	 <u>Tai chinh</u>
	 <u>Tin noi bo</u>	 <u>Ngoai giao</u>

Xây dựng giao diện mẫu cho menu

Dạng 2

Home	About	News	Products	Contact
		Tin trong nuoc		
		Tin quoc te	Cong nghe	
		Tin trong nha	Tai chinh	
		Tin noi bo	Ngoai giao	

Xây dựng giao diện mẫu cho menu

Dạng 3



Javascript là gì ?

- Ngôn ngữ thông dịch, mã nguồn của nó được nhúng hoặc tích hợp trực tiếp vào tập tin HTML. Khi trang web được tải xong, trình duyệt sẽ thông dịch và thực hiện các mã lệnh này.
- Được cung cấp hoàn toàn miễn phí

Javascript có thể làm gì?

- Làm cho trang HTML trở nên sinh động hơn.
- Phản ứng lại với một sự kiện nào đó từ phía người dùng.
- Đọc hoặc thay đổi nội dung của các phần tử trong trang HTML
- Kiểm tra dữ liệu
- Phát hiện các loại trình duyệt khác nhau
- Tạo các tập tin cookie lưu trữ và truy xuất thông tin trên máy tính của người truy cập website
- ...

Sử dụng Javascript như thế nào ?

- Để sử dụng Javascript rất đơn giản, chúng ta chỉ cần đưa các câu lệnh của nó vào trong thẻ <script> của HTML

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Example 01</title>
    <script type="text/javascript">
        document.write("Hello World!");
    </script>
</head>
<body>
</body>
</html>
```

Javascript thực thi lệnh khi nào ?

- TH1: Thực hiện lệnh ngay khi trang web được tải về trình duyệt của người sử dụng.
- TH2: Thực hiện lệnh khi nhận được một tác động nào đó như nhấn nút, di chuyển chuột, ...

Vị trí Javascript trong trang HTML

- Đặt trong cặp thẻ <head> của trang web
- Đặt trong cặp thẻ <body> của trang web
- Đặt trong tập tin .js sau đó nhúng tập tin này vào trang web

Mã lệnh Javascript

- Mã lệnh javascript là một chuỗi các câu lệnh.
- Các câu lệnh này kết thúc bằng dấu chấm phẩy “;”
- Phân biệt chữ hoa và chữ thường
- Ký tự khoảng trắng không ảnh hưởng đến kết quả thực thi của mã lệnh.

Biến trong Javascript

x = 5

y = 6

z = x + y = 5 + 6 =11

- Biến dùng để lưu trữ một giá trị nào đó có thể là một chuỗi, một đối tượng, một con số, một mảng, một phép toán ...
- Khai báo một biến trong JavaScript: `var <tên_var>;`

Quy tắc đặt tên biến

- Tên biến phải bắt đầu bằng ký tự hoặc dấu gạch dưới (_) và không có khoảng trắng.
- Tên biến phân biệt chữ hoa và chữ thường

Kiểu dữ liệu trong JavaScript

Kiểu	Ví dụ
String	<pre>var answer = "It's alright"; var answer = "He is called 'Johnny'";</pre>
Number	<pre>var x1 = 34.00; var x2 = 34;</pre>
Boolean	<pre>var x = true; var y = false;</pre>
Array	<pre>var cars=new Array(); cars[0]="Saab"; cars[1]="Volvo"; cars[2]="BMW";</pre>
Object	<pre>var person={firstname:"John", lastname:"Doe", id:5566};</pre>

Kiểm tra kiểu dữ liệu trong JavaScript

- Javascript là ngôn ngữ không ràng buộc về kiểu dữ liệu: không cần khai báo kiểu dữ liệu khi khai báo biến, một biến đang thuộc kiểu dữ liệu này có thể bị gán bởi một giá trị thuộc kiểu dữ liệu khác.
- Xác định kiểu dữ liệu của một biến ta sử dụng câu lệnh: `typeof <tên biến>;`

Lưu ý về kiểu dữ liệu của biến khi khai báo

```
<script type="text/javascript">
    var pi      = 3.14;
    var name    = "John Doe";
    var answer  = 'Yes I am!';
</script>
```

- Giá trị của một biến là một chuỗi khi và chỉ khi nó nằm trong cặp dấu ngoặc kép ("...") hoặc cặp dấu ngoặc đơn ('...')
- Giá trị của một biến là một số khi và chỉ khi nó không nằm trong cặp dấu ngoặc kép ("...") và không nằm trong cặp dấu ngoặc đơn ('...')

Sử dụng hàm trong JavaScript

Tại sao cần sử dụng hàm

- Xuất ra trình duyệt lời chào đối với mỗi thành viên trong diễn đàn ?

Sử dụng hàm trong JavaScript

Khai báo hàm

```
function function_name (var1, var2, ... , varN) {  
    // code goes here  
}
```

- var1, var2 ... varN được gọi là các tham số của hàm. Hàm có thể có nhiều tham số hoặc không có tham số nào cả
- Cách đặt tên hàm tương tự như cách đặt tên biến. Hoặc chúng ta dùng dấu gạch dưới (_) nếu tên hàm là một cụm từ.

Phân biệt biến cục bộ và biến toàn cục

Biến cục bộ (Local Variables)

- Phạm vi ảnh hưởng chỉ trong hàm mà nó được khai báo
- Vòng đời bắt đầu khi biến được khởi tạo
- Vòng đời kết thúc khi hàm thực hiện xong.

Biến toàn cục (Global Variables)

- Phạm vi ảnh hưởng đến toàn trang
- Vòng đời bắt đầu khi biến được khởi tạo
- Vòng đời kết thúc khi trang được đóng lại.

Toán tử trong JavaScript

Toán tử số học

Cho $y = 9$

Toán tử	Miêu tả	Ví dụ	Kết quả
+	Cộng	$x = y + 2$	$x = 11$
-	Trừ	$x = y - 2$	$x = 7$
*	Nhân	$x = y * 2$	$x = 18$
/	Chia	$x = y / 2$	$x = 4.5$
%	Lấy giá trị lẻ	$x = y \% 2$	$x = 1$
++	Tăng	$x = ++y = y + 1$	$x = 10$
--	Giảm	$x = --y = y - 1$	$x = 8$

Toán tử trong JavaScript

Toán tử gán

Cho $x = 10$ và $y = 5$

Toán tử	Ví dụ	Hình thức khác	Kết quả
=	$x = y$		$x = 5$
+=	$x += y$	$x = x + y$	$x = 15$
-=	$x -= y$	$x = x - y$	$x = 5$
*=	$x *= y$	$x = x * y$	$x = 50$
/=	$x /= y$	$x = x / y$	$x = 2$
%=	$x \% y$	$x = x \% y$	

Toán tử trong JavaScript

Toán tử so sánh

Cho $x = 5$

Toán tử	Mô tả	Ví dụ	Kết quả
$==$	So sánh bằng	$x == 8$	false
$====$	So sánh tuyệt đối	$x === "5"$ $x === 5$	false true
$!=$	So sánh không bằng	$x != 8$	true
$>$	So sánh lớn hơn	$x > 8$	false
$<$	So sánh nhỏ hơn	$x < 8$	true
$>=$	So sánh lớn hơn hoặc bằng	$x >= 8$	false
$<=$	So sánh nhỏ hơn hoặc bằng	$x <= 8$	true

Toán tử trong JavaScript

Toán tử logic

Cho $x = 6$ và $y = 3$

Toán tử	Mô tả	Ví dụ	Kết quả
$&&$	And	$(x < 10 \ \&\& \ y > 1)$ $(x < 10 \ \&\& \ y > 4)$	true false
$\ $	Or	$(x < 10 \ \ \ y > 1)$ $(x < 10 \ \ \ y > 4)$	true true
!	Not	$! (x == y)$ $! (x != y)$	true false

Toán tử trong JavaScript

Toán tử điều kiện

Cú pháp: `variablename = (condition) ? value1 : value2;`

```
<script type="text/javascript">
    var age = 5;
    var result = ( age > 5 ) ? "Vào cấp 1" : "Không được vào cấp 1";
    document.write(result);
</script>
```

Câu điều kiện trong Javascript

- Câu điều kiện là câu lệnh mà chúng ta thường xuyên sử dụng khi viết mã cho bất kỳ ngôn ngữ lập trình nào. Để thực hiện những hành động khác nhau trong những điều kiện khác nhau.
- Hai câu lệnh điều kiện thường được sử dụng trong JavaScript:
 - Câu điều kiện IF ... ELSE
 - Câu điều kiện SWITCH

Câu điều kiện IF ... ELSE

Câu lệnh IF

- Sử dụng câu lệnh này để thực hiện một số mã lệnh nếu đúng điều kiện định ra.

```
<script type="text/javascript">
    if(condition) {
        // code goes here
    }
</script>
```

Câu điều kiện IF ... ELSE

Câu lệnh IF ... ELSE

- Sử dụng câu lệnh điều kiện này để thực hiện một số mã lệnh nếu đúng điều kiện đã định và nếu không đúng điều kiện đã định thì thực hiện mã lệnh khác

```
<script type="text/javascript">
    if(condition) {
        // code goes here
    }else {
        // code goes here
    }
</script>
```

Câu điều kiện IF ... ELSE

Câu lệnh IF ... ELSE IF ... ELSE

- Sử dụng câu lệnh điều kiện này trong trường hợp có nhiều điều kiện đặt ra và khi thỏa mỗi điều kiện sẽ thực hiện một số mã lệnh khác nhau.

```
<script type="text/javascript">
    if(condition1) {
        // code goes here
    }else if(condition2) {
        // code goes here
    }else {
        // code goes here
    }
</script>
```

Câu điều kiện SWITCH

- Câu điều kiện Switch có một điều kiện mặc định, nghĩa là khi giá trị đưa vào không thỏa một điều kiện nào thì nó sẽ lấy các câu lệnh trong phần điều kiện mặc định để thực hiện.

```
<script type="text/javascript">
    switch (expression) {
        case constant1:
            // execute code block 1
            break;
        case constant1:
            // execute code block 2
            break;
        default:
            // code to be executed if n is different from case 1 and 2
    }
</script>
```

Vòng lặp trong JavaScript

- Vòng lặp được dùng để thực thi một số việc nào đó cho đến khi đúng điều kiện thì thoát khỏi vòng lặp và thi hành lệnh tiếp theo
- Các vòng lặp thường được sử dụng trong JavaScript:
 - Vòng lặp FOR
 - Vòng lặp WHILE
 - Vòng lặp DO ... WHILE
 - Vòng lặp FOR ... IN

Vòng lặp FOR

```
<script type="text/javascript">
    for (startValue; endValue; varIncrement) {
        //Code goes here
        /*
            startValue:      giá trị bắt đầu
            endValue:       giá trị cuối cùng
            varIncrement:   giá trị tăng của vòng lặp
        */
    }
</script>
```

Vòng lặp WHILE

- Vòng lặp while thực hiện một khối lệnh khi điều kiện thỏa và dừng lại ngay khi điều kiện không thỏa

```
<script type="text/javascript">
    while(expression) {
        //Code goes here
    }
</script>
```

Vòng lặp DO ... WHILE

- Vòng lặp này sẽ thực hiện khối lệnh ít nhất một lần, rồi sau đó mới kiểm tra điều kiện. Khối lệnh vẫn sẽ được thực hiện khi biểu thức điều kiện vẫn còn đúng.

```
<script type="text/javascript">
  do {
    //Code goes here
  }while(expression);
</script>
```

Sử dụng break và continue trong vòng lặp

- Câu lệnh break có chức năng thoát khỏi một vòng lệnh. Nó có thể được sử dụng để nhảy ra khỏi một vòng lặp.
- Câu lệnh continue có chức năng dừng vòng lặp tại giá trị đó và nhảy sang giá trị khác trong vòng lặp

Ôn tập

0 1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
2 3 4 5 6 7 8 9
3 4 5 6 7 8 9
4 5 6 7 8 9
5 6 7 8 9
6 7 8 9
7 8 9
8 9
9

0
1 0 1
2 1 0 1 2
3 2 1 0 1 2 3
4 3 2 1 0 1 2 3 4
5 4 3 2 1 0 1 2 3 4 5
6 5 4 3 2 1 0 1 2 3 4 5 6
7 6 5 4 3 2 1 0 1 2 3 4 5 6 7
8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8
9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9

JavaScript Errors - Throw and Try to Catch

- Các nguyên nhân gây ra lỗi khi thực thi JavaScript: lỗi cú pháp, lỗi đầu vào bị sai và các nguyên nhân khó xác định khác
- Trong lập trình có những ngoại lệ mà chúng ta ít để ý tới:
 - Phép chia giữa 2 số a và b, khi b bằng 0
 - Đọc và ghi file nhưng file chưa được tạo hay không có sẵn
 - Chưa điền dữ liệu vào text box nhưng vẫn đưa ra xử lý

JavaScript Errors - Throw and Try to Catch

- Cú pháp:

```
<script type="text/javascript">

    try{
        //Run some code here
    }catch(err){
        //Handle errors here
    }

</script>
```

JavaScript Object

- Đối tượng là một khái niệm bao gồm hai thành phần: thuộc tính (đặc điểm) và phương thức (hành động)
- Ví dụ đối với đối tượng chiếc xe, chúng ta có:
 - Thuộc tính: màu sơn, cân nặng, loại xe, ...
 - Phương thức: chạy thẳng, chạy lùi, dừng, đỗ, ...
- Các đối tượng có sẵn trong JavaScript: number, string, boolean, array, date, math, screen, location, ...

JavaScript Object

- Khởi tạo đối tượng

```
var objectName = new Object();
```

- Truy cập thuộc tính của đối tượng

```
objectName.propertyName
```

- Truy cập phương thức của đối tượng

```
objectName.methodName()
```

JavaScript Number Object

- JavaScript Number Object bao gồm tất cả các số nguyên, số thập phân, ...
- Tất cả các số trong JavaScript được lưu trữ dưới dạng 64-bit (8-byte)

```
<script type="text/javascript">

    var pi  = 3.14159;      // Số thập phân
    var x   = 12;           // Số nguyên
    var y   = 123e5;         // 12300000 = 123 x 10^5
    var z   = 123e-5;        // 0.00123 = 123 x 10^(-5)

</script>
```

Number Object Properties

Thuộc tính	Miêu tả
constructor	Trả về hàm khởi tạo đối tượng
MAX_VALUE (*)	Trả về số lớn nhất có thể có trong JavaScript
MIN_VALUE (*)	Trả về số nhỏ nhất có thể có trong JavaScript
NEGATIVE_INFINITY (*)	Trả về giá trị nhỏ hơn MIN_VALUE (âm vô cùng trong toán học)
NaN (*)	Kết quả được trả về khi các phép tính số học không hợp lệ
POSITIVE_INFINITY (*)	Trả về giá trị lớn hơn MAX_VALUE (dương vô cùng trong toán học)
prototype (*)	Thêm các thuộc tính hoặc các phương thức mới cho đối tượng

Number Object Methods

Phương thức	Miêu tả
toExponential(x)	Chuyển một số về dạng số e, với x là số mũ
toFixed(x)	Làm tròn số với x chữ số thập phân
toPrecision(x)	Chuyển một số thành một chuỗi với chiều dài là x
toString()	Chuyển một số thành một chuỗi
valueOf()	Trả về giá trị ban đầu của một đối tượng Number.

Number Object – Ôn tập

Viết các phương thức mở rộng cho Number Object thực hiện các yêu cầu sau:

1. Tìm chỉ số lớn nhất của số tự nhiên n bất kỳ
2. Đảo ngược các chỉ số của số tự nhiên n bất kỳ
3. Tìm ước chung lớn nhất và bội chung nhỏ nhất của hai số

JavaScript Array Object

- Mảng là một đối tượng được sử dụng để lưu trữ nhiều giá trị trong một biến duy nhất

```
<script type="text/javascript">
    var array = new Array();
    var array = new Array("HTML", "CSS", "JavaScript");
    var array = ["HTML", "CSS", "JavaScript"];
</script>
```

Array Object Properties

Thuộc tính	Miêu tả
constructor	Trả về hàm khởi tạo đối tượng
length	Trả về hoặc thiết lập số phần tử của chuỗi
prototype	Thêm các thuộc tính hoặc các phương thức mới cho đối tượng

Array Object Methods (1)

Phương thức	Miêu tả
concat()	Nối các mảng lại với nhau và trả về mảng mới
indexOf()	Trả về vị trí đầu tiên của phần tử được tìm kiếm trong mảng
lastIndexOf()	Trả về vị trí cuối cùng của phần tử được tìm kiếm trong mảng
join()	Nối các phần tử trong mảng thành một chuỗi duy nhất
pop()	Xóa phần tử cuối cùng trong mảng và trả về phần tử bị xóa đó
shift()	Xóa phần tử đầu tiên trong mảng và trả về phần tử bị xóa đó
push()	Thêm mới phần tử vào cuối mảng và trả về chiều dài mới của mảng
unshift()	Thêm mới phần tử vào đầu mảng và trả về chiều dài mới của mảng

Array Object Methods (2)

Phương thức	Miêu tả
slice(start, end)	Tạo một mảng mới bằng cách trích xuất các phần tử của mảng ban đầu, từ vị trí s đến vị trí e (không tính phần tử e)
sort()	Sắp xếp một mảng theo giá trị của các phần tử trong mảng đó
splice()	Thêm hoặc xóa phần tử trong mảng (thêm tương tự push(), xóa tương tự slice())
reverse()	Đảo ngược các phần tử trong mảng
toString()	Chuyển một mảng thành một chuỗi
valueOf()	Trả về giá trị ban đầu của một đối tượng Array.

JavaScript String Object

- JavaScript String Object được sử dụng để thao tác với tất cả các định dạng chuỗi.
- Khai báo đối tượng String

```
<script type="text/javascript">  
  
    var str = new String("Javascript is very very easy");  
  
</script>
```

String Object Properties

Thuộc tính	Miêu tả
constructor	Trả về hàm khởi tạo đối tượng
length	Trả về chiều dài của chuỗi
prototype	Thêm các thuộc tính hoặc các phương thức mới cho đối tượng

String Object Methods (1)

Phương thức	Miêu tả
charAt(n)	Trả về giá trị của phần tử nằm ở vị trí thứ n trong một chuỗi nào đó
charCodeAt(n)	Trả về mã Unicode của phần tử nằm ở vị trí thứ n trong một chuỗi nào đó
fromCharCode(n) (*)	Chuyển đổi mã Unicode của giá trị n thành giá trị chuỗi
concat()	Nối các chuỗi lại với nhau thành một chuỗi duy nhất
indexOf()	Trả về vị trí xuất hiện đầu tiên (chỉ số nhỏ nhất) của một giá trị nào đó trong một chuỗi bất kỳ
lastIndexOf()	Trả về vị trí xuất hiện cuối cùng (chỉ số lớn nhất) của một giá trị nào đó trong một chuỗi bất kỳ

String Object Methods (2)

Phương thức	Miêu tả
search()	Tìm kiếm và trả về vị trí xuất hiện đầu tiên của một giá trị nào đó được tìm kiếm trong chuỗi
match()	Tìm kiếm và trả về giá trị nào đó được tìm kiếm trong một chuỗi.
replace(old, new)	Tìm kiếm và thay thế giá trị old bằng giá trị new của một chuỗi nào đó.
slice(start,end)	Trích xuất chuỗi từ vị trí thứ s đến vị trí thứ e (không bao gồm ký tự e)
substring(start,end)	Trích xuất chuỗi từ vị trí thứ s đến vị trí thứ e (không bao gồm ký tự e)
substr(start, length)	Trích xuất chuỗi từ vị trí thứ s đến vị trí thứ s+l (không bao gồm ký tự thứ s+l)

String Object Methods (3)

Phương thức	Miêu tả
split()	Tách một chuỗi thành một mảng các chuỗi con tùy theo điều kiện đưa vào
toLowerCase()	Chuyển một chuỗi thành chữ thường
toUpperCase()	Chuyển một chuỗi thành chữ in hoa
valueOf()	Trả về giá trị ban đầu của một đối tượng String.

HTML Wrapper Methods (1)

Phương thức	Miêu tả
anchor()	Tạo ra thẻ a của tài liệu HTML (không có thuộc tính href)
link()	Tạo ra thẻ a của tài liệu HTML (có thuộc tính href)
big()	Hiển thị chuỗi với kích thước font chữ lớn hơn bình thường
small()	Hiển thị chuỗi với kích thước font chữ nhỏ hơn bình thường
bold()	Hiển thị chuỗi với font chữ đậm
italics()	Hiển thị chuỗi với font chữ nghiêng
fontsize()	Hiển thị chuỗi với kích thước quy định (giá trị truyền vào từ 1 đến 7)
fontcolor()	Định dạng màu sắc cho chuỗi văn bản (red; #FFF; 255,0,0)

HTML Wrapper Methods (2)

Phương thức	Miêu tả
blink()	Hiển thị chuỗi với định dạng blink (nhấp nháy)
strike()	Hiển thị chuỗi với đường gạch ngang
sub()	Hiển thị văn bản nằm ở vị trí dưới cùng của dòng
sup()	Hiển thị văn bản nằm ở vị trí trên cùng của dòng
fixed()	Hiển thị chuỗi với font chữ monospace (font-family: monospace)

Bài tập ôn tập

- Viết 1 phương thức mở rộng của đối tượng String để tách họ tên của một người thành 3 phần: họ, họ lót và tên.

Họ: Lưu

Họ lót: Trường

Tên: Hải

JavaScript Math Object

- JavaScript Math Object có thể thực hiện các công việc liên quan đến tính toán trong số học.
- JavaScript Math Object không có thuộc tính constructor. Do đó tất cả các thuộc tính và phương thức đều được gọi trực tiếp bởi Math.

Math Object Properties

Thuộc tính	Miêu tả	Cách gọi	Kết quả
E	Trả về hằng số Euler	Math.E	2,718
LN2	Trả lại logarit của 2	Math.LN2	0,693
LN10	Trả lại logarit của 10	Math. LN10	2,302
LOG2E	Trả lại logarit cơ số 2 của E	Math. LOG2E	1,442
LOG10E	Trả lại logarit cơ số 10 của E	Math. LOG10E	0,434
PI	Trả về số pi	Math. PI	3,14
SQRT1_2	Trả về căn bậc 2 của 1/2	Math.SQRT1_2	0,707
SQRT2	Trả về căn bậc 2 của 2	Math.SQRT2	1,414

Math Object Methods (1)

1 radian = (180/pi) độ, đặt x = (pi/180) * 60

Phương thức	Miêu tả	Gọi hàm	Kết quả
cos(x)	Trả về giá trị cosine của x (radian)	Math.cos(x)	0.5
sin(x)	Trả về giá trị sine của x	Math.sin(x)	0.866
tan(x)	Trả về giá trị tangent của x	Math.tan(x)	1.732
acos(x)	Trả về giá trị arccosine của một số	Math.acos(0.8)	0.644
asin(x)	Trả về giá trị arcsine của một số	Math.asin(0.8)	0.927
atan(x)	Trả về giá trị arctangent của một số	Math.atan(0.8)	0.675
atan2(x)	Trả về giá trị số nằm giữa $-\pi/2$ và $\pi/2$	Math.atan2(0.8)	1.1071

Math Object Methods (2)

Phương thức	Miêu tả	Gọi hàm	Kết quả
max(x,y,z,...,n)	Trả về giá trị lớn nhất của dãy số	Math.max(3, 3.12, -3.13, 1, 7)	7
min(x,y,z,...,n)	Trả về giá trị nhỏ nhất của dãy số	Math.min(3, 3.12, -3.13, 1, 7)	-3.13
random()	Trả về số ngẫu nhiên nằm giữa 0 và 1	Math.random()	0,5
round(x)	Làm tròn 1 số đến số nguyên gần x nhất	Math.round(3.2)	3
ceil(x)	Làm tròn 1 số đến số nguyên lớn nhất và gần x nhất	Math.ceil(3.2)	4
floor(x)	Làm tròn 1 số đến số nguyên nhỏ nhất và gần x nhất	Math.floor(3.8)	

Math Object Methods (3)

Phương thức	Miêu tả	Gọi hàm	Kết quả
abs(x)	Trả về giá trị tuyệt đối của x	Math.abs(-7.12)	7.12
pow(x,y)	Trả về x mũ y	Math.pow(2,3)	8
exp(x)	Trả về giá trị E mũ x	Math.exp (2)	2.71
log(x)	Trả về logarit tự nhiên (cơ sở E) của x	Math.log(2)	0.693
sqrt(x)	Trả về giá trị căn bậc 2 của x	Math.sqrt(9)	3

Bài tập: Hình ảnh ngẫu nhiên

[Random image](#)



JavaScript Boolean Object

- JavaScript Boolean Object được sử dụng để chuyển một giá trị nào đó (không phải giá trị Boolean) thành các giá trị Boolean
- Các đối tượng Boolean đại diện cho hai giá trị: true hoặc false.

Boolean Object Properties

Thuộc tính	Miêu tả
constructor	Trả về hàm khởi tạo đối tượng
prototype	Thêm các thuộc tính hoặc các phương thức mới cho đối tượng

Boolean Object Methods

Phương thức	Miêu tả
toString()	Chuyển một giá trị boolean thành một chuỗi
valueOf()	Trả về giá trị ban đầu của một đối tượng Array.

Các giá trị được chuyển đổi sang Boolean Object

Giá trị	Giá trị của Boolean Object
Giá trị số khác 0	True
0	False
Chuỗi rỗng ("")	False
null	False
NaN	False
Chuỗi khác rỗng	True

JavaScript Date Object

- JavaScript Date Object được sử dụng để xử lý và thao tác với các dữ liệu thời gian: ngày, tháng, năm, giờ, phút, giây, ...

```
<script type="text/javascript">
    var d = new Date();
    var d = new Date(milliseconds);
    var d = new Date(dateString);
    var d = new Date(year, month, day, hours, minutes, seconds, milliseconds);
</script>
```

Date Object Properties

Thuộc tính	Miêu tả
constructor	Trả về hàm khởi tạo đối tượng
prototype	Thêm các thuộc tính hoặc các phương thức mới cho đối tượng

Date Object Methods (1)

Phương thức	Miêu tả
getDate()	Trả về số thứ tự của ngày tại thời điểm hiện tại trong tháng (1 → 31)
getDay()	Trả về số thứ tự của ngày tại thời điểm hiện tại trong tuần (0 → 6)
getFullYear()	Trả về giá trị năm tại thời điểm hiện tại (4 chữ số: 2013, ...)
getHours()	Trả về giá trị giờ tại thời điểm hiện tại (0 → 23)
getMilliseconds()	Trả về giá trị milli second tại thời điểm hiện tại (0 → 999)
getMinutes()	Trả về giá trị phút tại thời điểm hiện tại (0 → 59)
getMonth()	Trả về giá trị tháng tại thời điểm hiện tại (0 → 11)
getSeconds()	Trả về giá trị giây tại thời điểm hiện tại (0 → 59)
getTime()	Trả về số mili giây kể từ thời điểm 01/01/1970

Date Object Methods (2)

Phương thức	Miêu tả
getTimezoneOffset()	Trả về số phút chênh lệch giữa thời gian quốc tế và thời gian tại máy người dùng
getUTCDate()	Trả về số thứ tự của ngày trong tháng theo UTC (1 → 31)
getUTCDay()	Trả về số thứ tự của ngày trong tuần theo UTC (0 → 6)
getUTCFullYear()	Trả về giá trị năm theo UTC (4 chữ số: 2013, ...)
getUTCHours()	Trả về giá trị giờ theo UTC (0 → 23)
getUTCMilliseconds()	Trả về giá trị milli second theo UTC (0 → 999)
getUTCMilliseconds()	Trả về giá trị phút theo UTC (0 → 999)
getUTCMonth()	Trả về giá trị tháng theo UTC (0 → 11)
getUTCSeconds()	Trả về giá trị giây theo UTC (0 → 59)

Date Object Methods (3)

Phương thức	Miêu tả
setDate()	Thiết lập giá trị ngày, giờ, phút, giây, mili giây cho đối tượng kiểu Date
setFullYear()	Thiết lập giá trị năm, tháng, ngày, giờ, phút, giây, mili giây cho đối tượng kiểu Date
setHours()	Thiết lập giá trị giờ, phút, giây, mili giây cho đối tượng kiểu Date
setMilliseconds()	Thiết lập giá trị mili giây cho đối tượng kiểu Date
setMinutes()	Thiết lập giá trị phút, giây, mili giây cho đối tượng kiểu Date
setMonth()	Thiết lập giá trị tháng, ngày, giờ, phút, giây, mili giây cho đối tượng kiểu Date
setSeconds()	Thiết lập giá trị giây, mili giây cho đối tượng kiểu Date
setTime()	Thiết lập thời gian cho đối tượng kiểu Date bởi số mili giây truyền vào

Date Object Methods (4)

Phương thức	Miêu tả
setUTCDate()	Thiết lập giá trị ngày, giờ, phút, giây, mili giây cho đối tượng kiểu Date (UTC)
setUTCFullYear()	Thiết lập giá trị năm, tháng, ngày, giờ, phút, giây, mili giây cho đối tượng kiểu Date (UTC)
setUTCHours()	Thiết lập giá trị giờ, phút, giây, mili giây cho đối tượng kiểu Date (UTC)
setUTCMilliseconds()	Thiết lập giá trị mili giây cho đối tượng kiểu Date (UTC)
setUTCMilliseconds()	Thiết lập giá trị phút, giây, mili giây cho đối tượng kiểu Date (UTC)
setUTCMonth()	Thiết lập giá trị tháng, ngày, giờ, phút, giây, mili giây cho đối tượng kiểu Date (UTC)
setUTCSeconds()	Thiết lập giá trị giây, mili giây cho đối tượng kiểu Date (UTC)

Date Object Methods (5)

Phương thức	Miêu tả
toDateString()	Chuyển đổi giá trị của một đối tượng Date thành giá trị chuỗi dễ hiểu hơn đối với người sử dụng
toISOString()	Chuyển đổi giá trị của một đối tượng Date thành giá trị chuỗi theo chuẩn ISO (YYYY-MM-DDTHH:mm:ss.sssZ)
toJSON()	Chuyển đổi giá trị của một đối tượng Date thành giá trị chuỗi theo định dạng JSON (YYYY-MM-DDTHH:mm:ss.sssZ)
toLocaleDateString()	Chuyển đổi giá trị của một đối tượng Date thành giá trị chuỗi (ngày, tháng, năm), giá trị sau khi chuyển đổi dễ hiểu hơn đối với người sử dụng (theo định dạng tại máy tính cá nhân)
toLocaleTimeString()	Chuyển đổi giá trị của một đối tượng Date thành giá trị chuỗi (giờ, phút, giây), giá trị sau khi chuyển đổi dễ hiểu hơn đối với người sử dụng (theo định dạng tại máy tính cá nhân)

Date Object Methods (6)

Phương thức	Miêu tả
toLocaleString()	Chuyển đổi giá trị của một đối tượng Date thành giá trị chuỗi, giá trị sau khi chuyển đổi dễ hiểu hơn đối với người sử dụng (theo định dạng tại máy tính cá nhân)
toString()	Chuyển đổi giá trị của một đối tượng Date thành giá trị chuỗi
toTimeString()	Chuyển đổi giá trị của một đối tượng Date thành giá trị chuỗi (giờ, phút, giây)
toUTCString()	Chuyển đổi giá trị thời gian UTC thành giá trị chuỗi, giá trị sau khi chuyển đổi dễ hiểu hơn đối với người sử dụng
valueOf()	Trả về giá trị của đối tượng Date
parse()	Trả về chuỗi số mili giây kể từ thời điểm 01/01/1970 cho đến thời gian truyền vào

JavaScript RegExp Object

- JavaScript RegExp Object dùng để mô tả mẫu của các chuỗi văn bản.
- Để sử dụng thành thạo JavaScript RegExp đòi hỏi phải có kiến thức tốt về biểu thức chính quy (Regular Expression)

```
<script type="text/javascript">

    var patt      =  new RegExp(pattern,modifiers);
    var patt      =  /pattern/modifiers;

</script>
```

Modifiers

Modifier	Miêu tả
i	Không phân biệt chữ hoa và chữ thường
g	Trả về tất cả các giá trị phù hợp
m	Thực hiện trên tất cả các dòng

RegEx Object Properties

Thuộc tính	Miêu tả
global	Trả về kết quả true nếu g modifier được thiết lập, ngược lại trả về false
ignoreCase	Trả về kết quả true nếu i modifier được thiết lập, ngược lại trả về false
lastIndex	Trả về chỉ số tiếp theo của kết quả so sánh khi g modifier được thiết lập
multiline	Trả về kết quả true nếu m modifier được thiết lập, ngược lại trả về false
source	Trả về nội dung của pattern

RegEx Object Methods

Phương thức	Miêu tả
compile()	Biên dịch một biểu thức chính quy
exec()	Kiểm tra mẫu chuỗi được tìm kiếm có tồn tại trong chuỗi nguồn hay không? Nếu có trả về giá trị của mẫu chuỗi được tìm kiếm, ngược lại trả về giá trị null
test()	Kiểm tra mẫu chuỗi được tìm kiếm có tồn tại trong chuỗi nguồn hay không? Nếu có trả về giá trị true, ngược lại trả về giá trị false

Bài tập

1. Sử dụng RegExp Obj để loại bỏ các khoảng trắng dư thừa trong một chuỗi
2. Sử dụng RegExp Obj để thay đổi giá trị của thuộc tính href trong thẻ a
3. Sử dụng RegExp Obj để kiểm tra sự hợp lệ của các giá trị
 - Email
 - Tên đăng nhập

JavaScript Global Object

- Các phương thức và thuộc tính của JavaScript Global có thể được sử dụng bởi tất cả các JavaScript Objects khác.

Global Object Properties

Thuộc tính	Miêu tả
Infinity	Đại diện giá trị âm vô cùng và dương vô cùng trong toán học
NaN	Giá trị này xuất hiện khi biểu thức toán học không thực hiện được ("Not-a-Number")
undefined	Giá trị này được trả về khi một biến chưa được gán giá trị mặc định

Global Object Methods (1)

Phương thức	Miêu tả
decodeURI()	Giải mã chuỗi URI
decodeURIComponent()	Giải mã chuỗi URI được mã hóa bởi encodeURIComponent()
encodeURI()	Mã hóa các ký tự đặc biệt trong chuỗi URI (ngoại trừ , / ? : @ & = + \$ #)
encodeURIComponent()	Mã hóa các ký tự đặc biệt trong chuỗi URI
escape()	Mã hóa các ký tự đặc biệt có trong chuỗi
unescape()	Giải mã các ký tự đặc biệt có trong chuỗi
eval()	Thực hiện biểu thức hoặc các câu lệnh được truyền vào

Global Object Methods (2)

Phương thức	Miêu tả
isFinite()	Kiểm tra giá trị truyền vào có phải là một số hữu hạn hay không.
isNaN()	Kiểm tra giá trị truyền vào có phải là một NaN hay không.
Number()	Chuyển đổi giá trị của các đối tượng khác thành giá trị của đối tượng Number
parseFloat()	Chuyển đổi giá trị chuỗi thành giá trị số
parseInt()	Chuyển đổi giá trị chuỗi thành giá trị số
String()	Chuyển đổi giá trị của các đối tượng khác thành giá trị của đối tượng String

Hướng dẫn tạo đối tượng trong JavaScript

```
<script type="text/javascript">

    function myValidate1() {                      // Cách 01
        }

    this.myValidate2 = function() {                // Cách 02
        }

    var myValidate3 = {                          // Cách 3
        }

</script>
```

Thuộc tính và truy cập thuộc tính

```
<script type="text/javascript">

    this.myValidate = function() {
        this.msg      = "This is a test"; // Thuộc tính
    };

    var myObj = new myValidate();
    document.write("Properties - msg: " + myObj.msg);

</script>
```

Phương thức và truy cập phương thức

```
<script type="text/javascript">

    this.myValidate = function() {

        this.msg      = "This is a test"; // Thuộc tính

        this.getMsg = getMsg;           // Phương thức

        function getMsg() {
            return this.msg;
        }
    };

    var myObj = new myValidate();
    document.write("Methods - getMsg: " + myObj.getMsg());
}

</script>
```

Public - Private

```
<script type="text/javascript">

    this.myValidate = function() {
        var private      = "Private text";      // Private
        this.getPrivate = getPrivate;

        function getPrivate() {
            return private;
        }
    };

    var myObj = new myValidate();
    document.write("Private: " + myObj.private + "<br />"); // undefined
    document.write("getPrivate: " + myObj.getPrivate());          // Private text

</script>
```

Truyền giá trị vào đối tượng

```
<script type="text/javascript">

    this.myValidate = function(val1, val2) {
        this.msg      = val1;
        this.data     = val2;
    };

    var myObj = new myValidate("JavaScript",200);

    document.write("msg: " + myObj.msg + "<br />");           // JavaScript
    document.write("data: " + myObj.data);                         // 200

</script>
```

Truyền giá trị vào phương thức

```
<script type="text/javascript">

    this.myValidate = function() {
        this.msg      = "First value";
        this.setMsg = setMsg;

        function setMsg(val) {
            this.msg = val;
        }
    };

    var myObj = new myValidate();
    myObj.setMsg("Second value")

    document.write("msg: " + myObj.msg + "<br />");

</script>
```

Kế thừa

```
<script type="text/javascript">

    this.myValidate = function() {
        this.myValue = '';

        this.getValue = getValue;
        this.setValue = setValue;

        function getValue()      { return this.myValue; }
        function setValue(val)   { this.myValue = val; }
    };

    this.myValidateChild = function() {
        myValidate.call(this);
    }

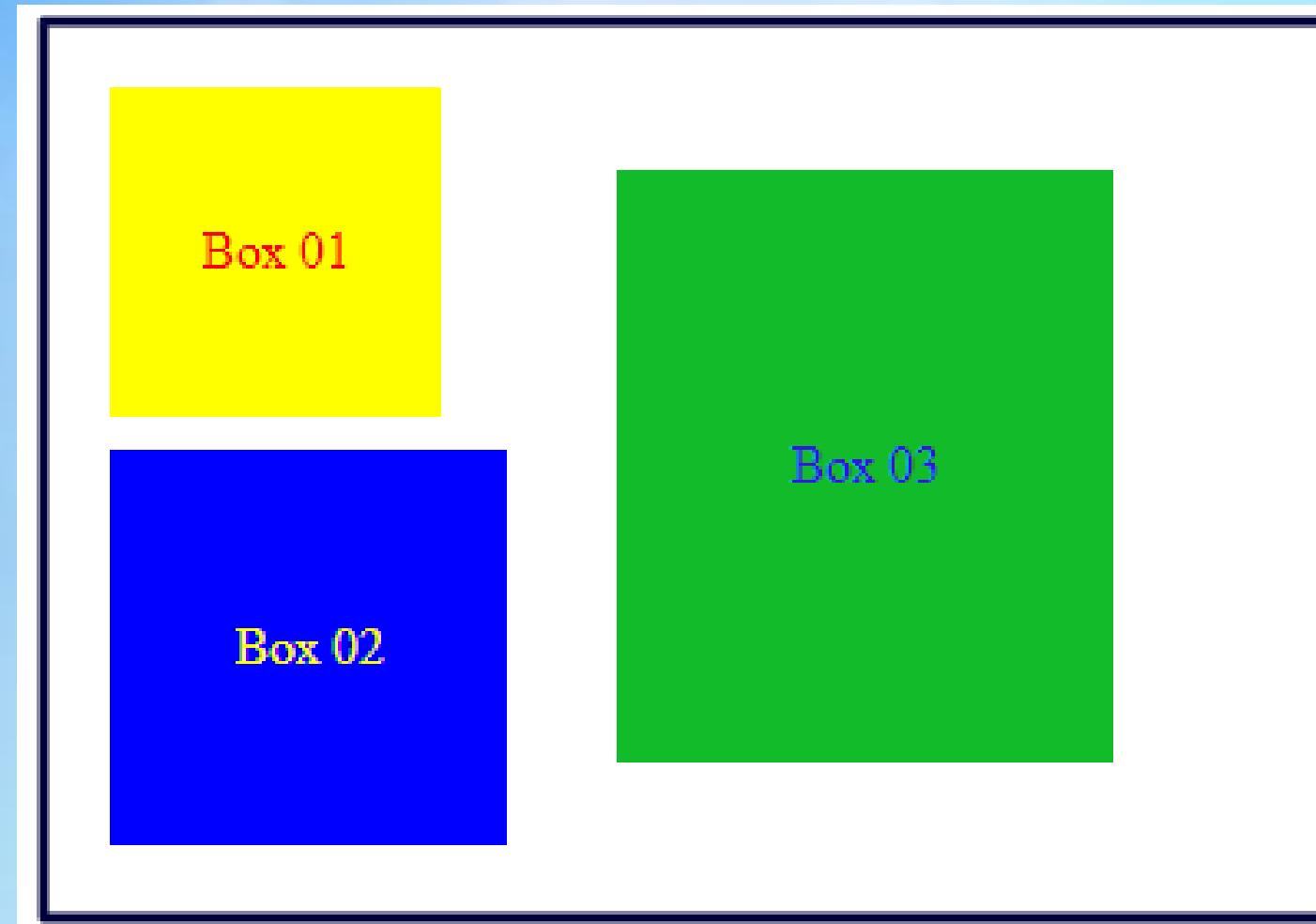
    var myObj = new myValidateChild();

    myObj.setValue("JavaScript");
    document.write("Child: " + myObj.myValue);

</script>
```

Ôn tập

Tạo ra đối tượng box để vẽ các box bên dưới (biết rằng để vẽ được 1 box cần các tham số như sau ["100px","100px","yellow","red"], "Box 01")



Browser Object Model

- Browser Object Model (BOM) cho phép JavaScript thao tác với trình duyệt
- Bao gồm các đối tượng: Window, Screen, Location, History, Navigator, PopupAlert, Timing, Cookies

JavaScript Window

- JavaScript Window Object đại diện cho cửa sổ đang mở trên trình duyệt
- Cung cấp các thuộc tính và phương thức để thao tác với các cửa sổ đang làm việc

Đối tượng con

document	Trả về đối tượng Document của cửa sổ
history	Trả về đối tượng History của cửa sổ
location	Trả về đối tượng Location của cửa sổ
navigator	Trả về đối tượng Navigator của cửa sổ
screen	Trả về đối tượng Screen của cửa sổ

Hộp thoại

alert	Hiển thị hộp thoại thông báo và nút nhấn OK
confirm	Hiển thị hộp thoại thông báo xác nhận cùng nút nhấn OK và Cancel <ul style="list-style-type: none">• Nhấn OK trả về true• Nhấn Cancel trả về false
prompt	Hiển thị hộp thoại thông báo và cho người dùng nhập vào một giá trị nào đó

open: Mở một cửa sổ mới

window.open(URL,name,specs,replace)

URL	URL của trang muốn mở (giá trị mặc định about:blank)
name	Chế độ xuất hiện của cửa sổ mới (_blank, _parent, _self, _top, name)
specs	Thiết lập các thuộc tính của cửa sổ muốn mở: width, height, top, left, ...
replace	Trả về một mảng các frame hiện có tại cửa sổ đang mở

Kích thước cửa sổ

Thuộc tính	Miêu tả
innerHeight	Thiết lập hoặc trả về chiều cao phần chứa nội dung của một cửa sổ
innerWidth	Thiết lập hoặc trả về chiều rộng phần chứa nội dung của một cửa sổ
outerHeight	Thiết lập hoặc trả về chiều cao của cửa sổ (kể cả phần toolbar và scrollbars)
outerWidth	Thiết lập hoặc trả về chiều rộng của cửa sổ (kể cả phần toolbar và scrollbars)

Các thuộc tính của JS Window

defaultStatus	Thiết lập hoặc trả về giá trị nội dung mặc định trên thanh trạng thái của một cửa sổ (Opera only)
frames	Trả về một mảng các frames trong cửa sổ hiện tại
length	Trả về số lượng frames trong cửa sổ hiện tại
name	Thiết lập hoặc trả về giá trị name của cửa sổ
status	Thiết lập hoặc trả về giá trị nội dung mặc định trên thanh trạng thái của một cửa sổ (Opera only)

Tọa độ

screenLeft	Trả về tọa độ x của cửa sổ so với màn hình (không hoạt động trên Firefox)
screenTop	Trả về tọa độ y của cửa sổ so với màn hình (không hoạt động trên Firefox)
screenX	Trả về tọa độ x của cửa sổ so với màn hình (Firefox)
screenY	Trả về tọa độ y của cửa sổ so với màn hình (Firefox)

JavaScript Timing

Trong JavaScript chúng ta có thể thiết lập thời gian để gọi một số đoạn code nào đó. Điều này còn được gọi là “Timing events”

setInterval()	Thực hiện mã lệnh sau một khoảng thời gian quy định
clearInterval()	Ngừng các mã lệnh đã được gọi bởi setInterval()
setTimeout()	Thực hiện mã lệnh tại một khoảng thời gian quy định
clearTimeout()	Ngừng các mã lệnh đã được gọi bởi setTimeout()

Bài tập JavaScript Timing

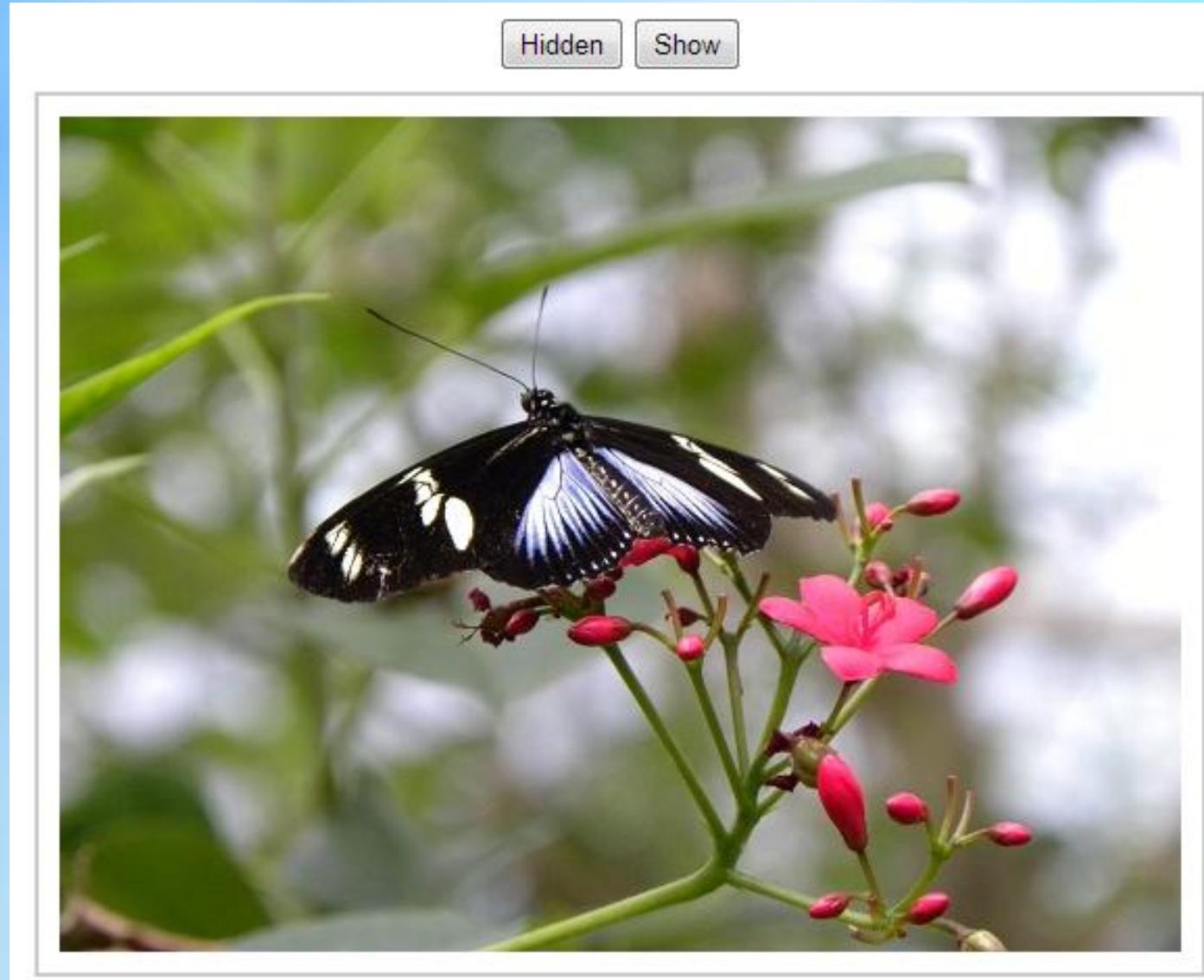
1. Xây dựng chương trình đồng hồ điện tử

My Clock

10:15:53 AM

Bài tập JavaScript Timing

2. Ẩn hiện một bức tranh kết hợp `setInterval()` và thuộc tính `opacity` của CSS



Bài tập JavaScript Timing

3. Xây dựng chương trình đồng hồ bấm giờ

My Clock

Start Pause Reset

00:02:91

focus & blur

Thuộc tính	Miêu tả
focus()	Thiết lập cửa sổ ở trạng thái focus (làm nổi bật)
blur()	Hủy bỏ trạng thái focus của một cửa sổ nào đó

Di chuyển cửa sổ

moveBy()

Di chuyển cửa sổ đến một tọa độ nào đó so với vị trí hiện tại của cửa sổ

moveTo()

Di chuyển cửa sổ đến một tọa độ nào đó so với màn hình

Thay đổi kích thước cửa sổ

resizeBy()

Thay đổi kích thước cửa sổ theo chiều cao và chiều rộng được truyền vào

resizeTo()

Thay đổi kích thước cửa sổ đến chiều cao và chiều rộng được truyền vào

Thao tác với thanh trược

scrollBy(xnum, ynum)	Cuộn nội dung trong cửa sổ theo chiều ngang là xnum (pixel), theo chiều dọc là ynum (pixel)
scrollTo(xpos, ypos)	Cuộn nội dung trong cửa sổ đến vị trí xpos và ypos
pageXOffset	Trả về giá trị pixel khi di chuyển thanh cuộn (theo chiều ngang)
pageYOffset	Trả về giá trị pixel khi di chuyển thanh cuộn (theo chiều dọc)

Mối quan hệ giữa các cửa sổ

Thuộc tính	Miêu tả
opener	Trả về cửa sổ đã mở ra cửa sổ hiện tại
parent	Trả về cửa sổ cha của cửa sổ hiện tại
self	Trả về cửa sổ đang thao tác hiện tại
top	Trả về cửa sổ trên cùng của trình duyệt

JavaScript Navigator

- JavaScript Navigator Object cung cấp các thông tin về trình duyệt web (browser): tên, phiên bản của trình duyệt, các plugin được cài đặt trong trình duyệt, ...
- Đa số các phương thức và thuộc tính của Navigator đều được hỗ trợ bởi các trình duyệt phổ biến hiện nay

Navigator Object Properties (P1)

appCodeName	Trả về “code name” (mã tên) của trình duyệt
appName	Trả về tên của trình duyệt
appVersion	Trả về thông tin phiên bản của trình duyệt
cookieEnabled	Kiểm tra các tập tin cookie đã được bật hay chưa ? Kết quả trả về là giá trị thuộc đối tượng Boolean (true/false, 1/0)

Navigator Object Properties (P2)

online	Kiểm tra trình duyệt có được thiết lập ở trạng thái online không?
platform	Trả về phiên bản (platform) mà trình duyệt đang được thực thi
userAgent	Trả về giá trị user-agent được gửi đến server bởi trình duyệt
language systemLanguage	Trả về ngôn ngữ mà trình duyệt đang được thiết lập
plugins	Trả về danh sách các plugins được tích hợp trong trình duyệt

Navigator Object Methods

javaEnabled()	Kiểm tra trình duyệt có hỗ trợ Java hay không? Kết quả trả về là giá trị thuộc đối tượng Boolean (true/false, 1/0)
taintEnabled()	Kiểm tra trình duyệt có hỗ trợ “data tainting” hay không? Phương thức này chỉ hỗ trợ trên trình duyệt IE và Opera

Bài tập

1. Xác định tên của trình duyệt đang truy cập vào website?
2. Xác định ngôn ngữ được thiết lập bởi trình duyệt ?
3. Kiểm tra trình duyệt đã cài đặt plugin “Shockwave Flash” hay chưa?

Lưu ý khi sử dụng JS Navigator

Các thông tin nhận được từ JS Navigator không nên sử dụng để xác định các phiên bản của trình duyệt

1. Các thông tin nhận được từ JS Navigator có thể được thay đổi bởi người dùng
2. Trình duyệt không xác nhận được phiên bản hệ điều hành được phát triển trước nó
3. Một số trình duyệt tự ngô nhận để vượt qua 1 số trang web

JavaScript Screen

- JavaScript Screen Object cung cấp các thông tin về màn hình của người dùng truy cập website
- Các thuộc tính của Screen đều được hỗ trợ bởi các trình duyệt phổ biến hiện nay

Screen Object Properties

availHeight	Trả về chiều cao của màn hình (không bao gồm phần taskbar)
availWidth	Trả về chiều rộng của màn hình (không bao gồm phần taskbar)
colorDepth	Trả về số bits được sử dụng để chỉ ra màu sắc của một pixel trong một hình ảnh
height	Trả về chiều cao của màn hình (bao gồm phần taskbar)
pixelDepth	Trả về độ phân giải màu (số bit trên một điểm ảnh) của màn hình
width	Trả về chiều rộng của màn hình (bao gồm phần taskbar)

JavaScript History

- JavaScript History Object bao gồm các phương thức và thuộc tính giúp chúng ta thao tác với history của trình duyệt web

History Object Properties

length

Trả về số lượng URL được lưu trong danh sách history của trình duyệt

- IE và Opera bắt đầu bằng giá trị 0
- Firefox, Chrome, và Safari bắt đầu bằng giá trị 1

History Object Methods

back()	Tải lại URL trước đó có trong danh sách history của trình duyệt Tương đương history.go(-1)
forward()	Tải lại URL sau đó có trong danh sách history của trình duyệt Tương đương history.go(1)
go()	Tải lại một trang nào đó có trong danh sách history của trình duyệt

JavaScript Location

- JavaScript Location Object bao gồm các thông tin về URL của cửa sổ hiện tại và chuyển hướng của cửa sổ đến các URL khác

Location Object Properties

hash	Thiết lập hoặc trả về giá trị hash (bắt đầu bởi dấu #) của một URL
host	Thiết lập hoặc trả về giá trị host và port của một URL
hostname	Thiết lập hoặc trả về giá trị hostname của một URL
href	Thiết lập hoặc trả về giá trị href của một URL
pathname	Thiết lập hoặc trả về giá trị pathname của một URL
port	Thiết lập hoặc trả về giá trị port của một URL
protocol	Thiết lập hoặc trả về giá trị protocol của một URL
search	Thiết lập hoặc trả về giá trị search (bắt đầu bởi dấu ?) của một URL

Các thành phần của URL

href

protocol

://

host

/

pathname

hash

hostname

port

Xác định các thành phần của URL

`http://www zend vn:8080/public/tin-cong-nghe/cd-o-viet-nam.html#title`

- Href: `http://www zend vn:8080/public/tin-cong-nghe/cd-o-viet-nam.html#title`
- Protocol: `http`
- Host: `www zend vn:8080`
- Hostname: `www zend vn`
- Port: `8080`
- Pathname: `public/tin-cong-nghe/cd-o-viet-nam.html#title`
- Hash: `#title`

Location Object Methods

assign()

Tải một URL mới

reload (*forceGet*)

Tải lại trang hiện tại. Tham số *forceGet* nhận 2 giá trị

- True: Tải lại trang từ server
- False: Tải lại trang từ cache (mặc định)

replace()

Thay thế URL hiện tại bởi một URL mới

JavaScript Cookies

- Cookie là một dạng bản ghi được tạo ra và lưu lại trên trình duyệt khi người dùng truy cập vào một website nào đó
- Đối với người dùng, cookie làm cho website trở nên tiện lợi hơn, họ có thể truy cập vào website nhanh hơn không phải nhập lại các thông tin nhiều lần.

Một số trường hợp sử dụng cookie

- Đối với các trang web thương mại điện tử, sử dụng cookie để ghi nhớ các thông tin khách hàng để họ không cần nhập lại thông tin nhiều lần khi tiến hành đặt hàng.
- Lưu thông tin ngày gần đây nhất mà người dùng truy cập vào website
- Thống kê số người truy cập trên website
- ...

Các thành phần của cookie

- Expires: Xác định thời điểm cookie sẽ bị hết hiệu lực trên trình duyệt.
- Domain: Xác định tên miền sẽ được sử dụng để gửi cookie đi
- Path: Đường dẫn hoặc thư mục trên trang web đã thiết lập cookie
- Secure: Nếu giá trị này được thiết lập bên trong cookie, thông tin sẽ được mã hoá trong quá trình truyền giữa server và browser
- Name: tên để phân biệt giữa các cookie.

Ôn tập

1. Sử dụng JS Cookie để lưu tên của một khách viếng thăm. Sau này, mỗi khi vị khách đó truy cập vào trang web sẽ nhận được một lời chào.
2. Thống kê số lần truy cập vào trang web của người dùng
3. Lưu trữ các thông tin ở các ô input đã được nhập bởi người dùng

Các đối tượng trong DOM

Node Object
<p>Properties:</p> <ul style="list-style-type: none"> .attributes .childNodes .firstChild .lastChild .nextSibling .nodeName .nodeType .nodeValue .parentElement .parentNode .previousSibling <p>Methods:</p> <ul style="list-style-type: none"> .appendChild(<i>newChild</i>) .cloneNode(<i>deep</i>) .hasChildNodes() .insertBefore(<i>newChild, refChild</i>) .removeChild(<i>oldChild</i>) .removeNode(<i>removeChildren</i>) .replaceChild(<i>newChild, refChild</i>)

NodeList Object
<p>Properties:</p> <ul style="list-style-type: none"> .length <p>Methods:</p> <ul style="list-style-type: none"> .item(<i>index</i>)

NamedNodeMap Object
<p>Properties:</p> <ul style="list-style-type: none"> .length <p>Methods:</p> <ul style="list-style-type: none"> .getNamedItem(<i>attr-name</i>) .item(index) (IE8+) <ul style="list-style-type: none"> .value .textContent (IE9+) .removeAttribute(<i>attr-name</i>) (IE8+) .setNamedItem(<i>arg</i>)

CharacterData Object
<p>Properties:</p> <ul style="list-style-type: none"> .length .data <p>Methods:</p> <ul style="list-style-type: none"> .appendData(<i>str</i>) .deleteData(<i>offset, count</i>) .insertData(<i>offset, str</i>) .replaceData(<i>offset, count, arg</i>) .substringData(<i>offset, count</i>)

Text Object

Properties:

- .length
- .data

Methods:

- .splitText(`offset`)

Element Object

Properties:

- .tagName
- .data

Methods:

- .getAttribute(`name`)
- .getElementsByTagName(`name`)
- .removeAttribute(`name`)
- .setAttribute(`name, value`)

Document Object

Properties:

- .documentElement

Methods:

- .createElement(`tagName`)
- .createTextNode(`data`)
- .getElementById(`elementId`)
- .getElementsByTagName(`tagname`)

CSSStyleDeclaration Object

Properties:

- .cssText
- .fontSize
- .fontWeight
- (thuộc tính CSS của đối tượng HTML)

Methods:

- .removeProperty(`name`)
- .removeAttribute(`name`)

HTMLElement Object

Properties:

- .childNodes
- .children
- .className
- .currentStyle
- .dir
- .document
- .id
- .innerHTML
- .innerText
- .lang
- .offsetHeight
- .offsetLeft

HTMLDocument Object

Properties:

- .childNodes
- .all [IE]
- .anchors
- .applets
- .body
- .cookie
- .domain
- .embeds
- .forms
- .images
- .links

```
.offsetParent  
.offsetTop  
.offsetWidth  
.style  
.title
```

Methods:

```
.attachEvent(event, func)  
.addEventListener(event, func, false)  
.detachEvent(event, func)  
.removeEventListener(event, func,  
false);  
.insertAdjacentHTML(where, HTMLText)
```

```
.location  
.parentWindow  
.plugins  
.referrer  
.scripts  
.styleSheets [IE]  
.title  
.uniqueID [IE]  
.URL
```

Methods:

```
.attachEvent(event, func) [IE]  
.captureEvents(eventTypeList)  
.close()  
.createStyleSheet(url, index) [IE]  
.getElementById(elementId)  
.getElementsByName(elementName)  
.open()  
.tags()  
.write(text)  
.writeln(text)
```

jQuery library

- **jQuery library**
 - **Selectors**
 - Basics
 - Hierarchy
 - Basic Filters
 - Content Filters
 - Visibility Filters
 - Attribute
 - Child Filters
 - Forms
 - **Attributes / CSS (Nhóm thuộc tính và CSS)**
 - Attributes
 - CSS
 - Dimensions
 - Offset
 - Data
 - **Manipulation (Nhóm thao tác)**
 - Copying
 - DOM Insertion, Around
 - DOM Insertion, Inside
 - DOM Insertion, Outside
 - DOM Removal
 - DOM Replacement
 - **Traversing**
 - Filtering
 - Miscellaneous Traversing
 - Tree Traversal
 - **Events (Nhóm sự kiện)**
 - Browser Events
 - Document Loading
 - Event Handler Attachment
 - Form Events
 - Mouse Events
 - Event Object
 - **Effects (Nhóm hiệu ứng)**
 - Basics
 - Custom
 - Fading

- Sliding
- Ajax (Nhóm Ajax)
 - Global Ajax Event Handlers
 - Helper Functions
 - Low-Level Interface
 - Shorthand Methods
- Core (Nhóm Lõi)
 - jQuery Object
 - Utilities
 - DOM Element Methods
 - Internals
 - Deferred Object
 - Callbacks Object

Need to Research

jQuery basic

jQuery()	.addClass()	.html()
.ready()	.appendTo()	.text()
.css()	.click()	jQuery.noConflict()

Core

jQuery Object

jQuery()
jQuery.noConflict()
jQuery.when()

Utilities

jQuery.contains()
jQuery.each()
jQuery.extend()
jQuery.globalEval()
jQuery.grep()
jQuery.inArray()
jQuery.isArray()
jQuery.isEmptyObject()
jQueryisFunction()
jQuery.isNumeric()
jQuery.isPlainObject()
jQuery.isWindow()
jQuery.isXMLDoc()
jQuery.makeArray()
jQuery.map()
jQuery.merge()
jQuery.noop()
jQuery.now()
jQuery.parseHTML()
jQuery.parseJSON()
jQuery.parseXML()
jQuery.proxy()
jQuery.support
jQuery.trim()
jQuery.type()
jQuery.unique()

DOM Element Methods

.get()
.index()
.toArray()

Internals

.jquery
.context
jQuery.error()
.length
.pushStack()
.selector

Callbacks Object

jQuery.Callbacks()
callbacks.add()
callbacks.disable()
callbacks.empty()
callbacks.fire()
callbacks.fired()
callbacks.fireWith()
callbacks.has()
callbacks.lock()
callbacks.locked()
callbacks.remove()

Deferred Object

deferred.always()
deferred.done()
deferred.fail()
deferred.notify()
deferred.notifyWith()
deferred.progress()
deferred.promise()
deferred.reject()
deferred.rejectWith()
deferred.resolve()
deferred.resolveWith()
deferred.state()
deferred.then()
.promise()

Selectors

Basics

```
*  
.class  
element  
#id  
selector1, selectorN, ...
```

Hierarchy

```
parent > child  
ancestor descendant  
prev + next  
prev ~ siblings
```

Basic Filters

```
:animated  
:eq()  
:even  
:first  
:gt()  
:header  
:lang()  
:last  
:lt()  
:not()  
:odd  
:root  
:target
```

Content Filters

```
:contains()  
:empty  
:has()  
:parent
```

Visibility Filters

```
:hidden  
:visible
```

Attribute

```
[name|="value"]  
[name*="value"]  
[name~="value"]  
[name$="value"]  
[name="value"]  
[name!="value"]  
[name^="value"]  
[name]  
[name="value"] [name2="value2"]
```

Child Filters

```
:first-child  
:first-of-type  
:last-child  
:last-of-type  
:nth-child()  
:nth-last-child()  
:nth-last-of-type()  
:nth-of-type()  
:only-child  
:only-of-type()
```

Forms

```
:button  
:checkbox  
:checked  
:disabled  
:enabled  
:focus  
:file  
:image  
:input  
:password  
:radio  
:reset  
:selected  
:submit  
:text
```

Traversing

Filtering

- .eq()
- .filter()
- .first()
- .has()
- .is()
- .last()
- .map()
- .not()
- .slice()

Miscellaneous Traversing

- .add()
- .andSelf()
- .contents()
- .each()
- .end()

Tree Traversal

- .addBack()
- .children()
- .closest()
- .find()
- .next()
- .nextAll()
- .nextUntil()
- .parent()
- .parents()
- .parentsUntil()
- .prev()
- .prevAll()
- .prevUntil()
- .siblings()

Attributes / CSS

Attributes

```
.attr()  
.prop()  
.removeAttr()  
.removeProp()  
.val()
```

Dimensions

```
.height()  
.innerHeight()  
.innerWidth()  
.outerHeight()  
.outerWidth()  
.width()
```

CSS

```
.addClass()  
.css()  
.hasClass()  
.removeClass()  
.toggleClass()
```

Offset

```
.offset()  
.offsetParent()  
.position()  
.scrollLeft()  
.scrollTop()
```

Data

```
jQuery.data()  
.data()  
jQuery.hasData()  
jQuery.removeData()  
.removeData()
```

Manipulation

Copying

`.clone()`

DOM Insertion, Around

`.wrap()`
`.wrapAll()`
`.wrapInner()`

DOM Insertion, Outside

`.after()`
`.before()`
`.insertAfter()`
`.insertBefore()`

DOM Insertion, Inside

`.append()`
`.appendTo()`
`.html()`
`.prepend()`
`.prependTo()`
`.text()`

DOM Removal

`.detach()`
`.empty()`
`.remove()`
`.unwrap()`

DOM Replacement

`.replaceAll()`
`.replaceWith()`

Effects

Basics

.hide()
.show()
.toggle()

Fading

.fadeIn()
.fadeOut()
.fadeTo()
.fadeToggle()

Custom

.animate()
.clearQueue()
.delay()
.dequeue()
jQuery.dequeue()
.finish()
jQuery.fx.interval
jQuery.fx.off
.queue()
jQuery.queue()
.stop()

Sliding

.slideDown()
.slideToggle()
.slideUp()

Ajax

Shorthand Methods

```
.load()
jQuery.get()
jQuery.post()
jQuery.getJSON()
jQuery.getScript()
```

Helper Functions

```
jQuery.param()
.serialize()
.serializeArray()
```

Low-Level Interface

```
jQuery.ajax()
jQuery.ajaxSetup()
```

Global Ajax Event Handlers

```
.ajaxComplete()
.ajaxError()
.ajaxSend()
.ajaxStart()
.ajaxStop()
.ajaxSuccess()
```

Ajax settings

Accepts{ }

async

beforeSend()

cache

complete()

contents{ }

contentType

context{ }

converters{ }

crossDomain

data

dataFilter()

dataType

error()

global

headers{ }

ifModified

isLocal

jsonp

jsonpCallback

mimeType

password

processData

scriptCharset

statusCode{ }

success()

timeout

traditional

type

url

username

xhr()

xhrFields{ }

Bootstrap 3

Biên soạn: ZendVN Group

Hướng dẫn: Lưu Trường Hải Lân

Responsive Web Design

jQuery Master
zend.vn

Vấn đề truy cập website từ phía người dùng



Java Master
zend.vn

Giải pháp cho việc hiển thị giao diện website

- Xử lý mã nguồn để nhận diện thiết bị người dùng đang sử dụng → chuyển hướng người dùng đến trang web được thiết kế dành riêng cho thiết bị
- Responsive design

Responsive design

- Responsive architect: các hệ thống cửa tự động, hệ thống báo cháy, ...
- Responsive design yêu cầu một thiết kế phải có khả năng tự động điều chỉnh để thích nghi với nhiều nhóm người sử dụng khác nhau.
- Tham khảo các mẫu responsive design

Responsive design

- Ưu điểm: thân thiện với người dùng, tiết kiệm thời gian bảo trì & nâng cấp
- Nhược điểm: mất nhiều thời gian xây dựng

Soạn thảo mã nguồn HTML CSS với Emmet Plugin

- Download và cài đặt vào Editor (Zend Studio 9.0.3)
- Thao tác nhanh với mã nguồn HTML
- Thao tác nhanh với mã nguồn CSS
- Một số thủ thuật khác

Sử dụng CSS media để xây dựng RWD

- Type: aural, braille, handheld, print, projection, screen, tty, and tv
- Dimensions
 - Browser: min-width, max-width, min-height, max-height
 - Device: min-device-width, max-device-width, min-device-height, and max-device-height
- Orientation: portrait, landscape
-

Getting started with Bootstrap 3

- Giới thiệu Bootstrap
- Download
- Cấu trúc thư mục
- Sự thích ứng của trình duyệt và hệ điều hành
- Demo đầu tiên

Hệ thống lưới (Grid system)

- Tập trung nội dung của trang trong thẻ div.container
- Sự ảnh hưởng của các class đến kích thước của màn hình: *extra small (.col-xs-), small (.col-sm-), medium (.col-md-), large (.col-lg-)*
- Quy tắc tổng 12
- Offsetting columns
- Nesting columns
- Column ordering: push & pull

Less CSS

- Cách sử dụng LESS
- Khai Báo Biến
- Mixins
- Nested Rules
- Operation
- Scope



NODE.JS



Biên soạn: ZendVN Group

Hướng dẫn: Lê Tấn Tài

Khóa Học: Lập trình Jquery



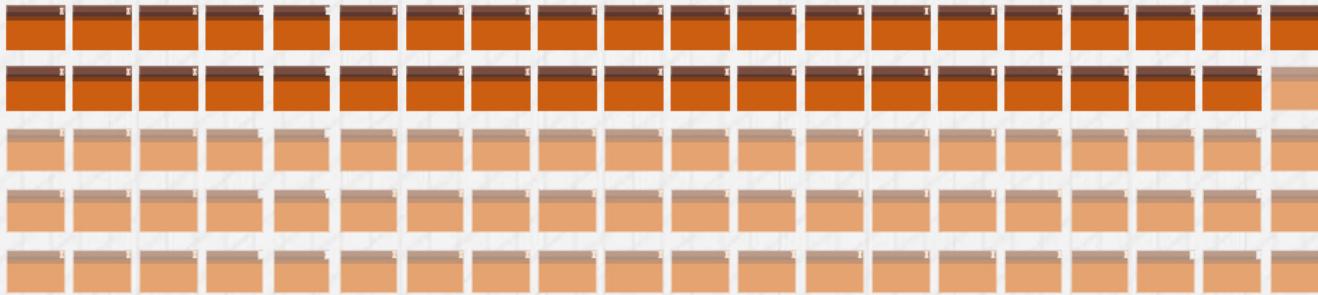


Sự bùng nổ web thời gian thực



2/3

SỐ NGƯỜI MỸ TRUY CẬP INTERNET
THÔNG QUA DI ĐỘNG



34/100 WEBSITE HÀNG ĐẦU THẾ GIỚI SỬ DỤNG HTML5



Sự phát triển
của di động



Sự phát triển của
HTML5



**SỰ BÙNG NỔ WEB
THỜI GIAN THỰC**

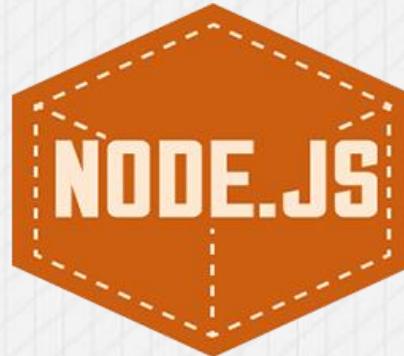


NODE.JS





Hãy Chọn Node.js



Nó là JavaScript



Lập Trình ở bất
kì cấp nào
(Client - Server-
database)



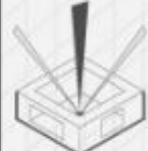
Cộng đồng lớn



Hiệu suất & khả
năng mở rộng cao



Thân thiện với
nhà phát triển



NODE.JS

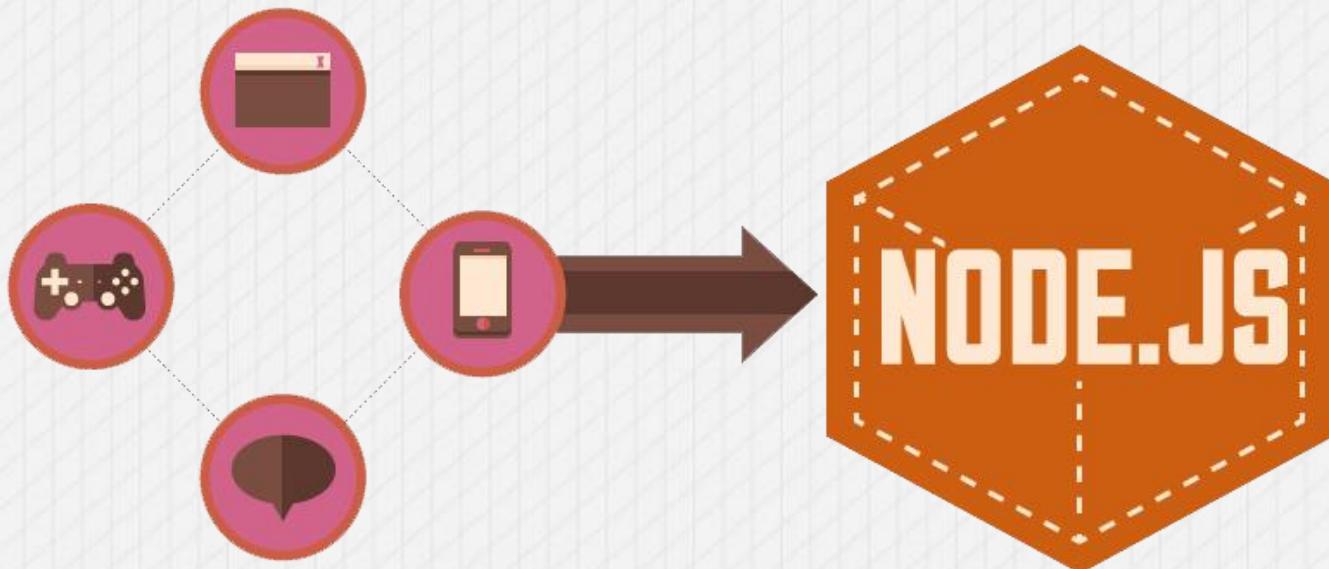




Tương lai Node.js



KHI CỘNG ĐỒNG PHÁT TRIỂN HỌ LUÔN ĐẶT RA YÊU CẦU MỚI



Với sự phát triển nhanh chóng về số lượng ứng dụng web, di động, mạng xã hội và game
cho thấy 1 tương lai tươi sáng với node.js



NODE.JS





Nội dung



- I. Node.js
- II. Node Package Modules (NPM)
- III. Socket.io Package
- IV. Mysql Package
- V. Xây dựng ứng dụng Chat.

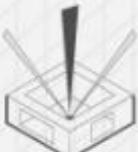




Giới thiệu Node.js

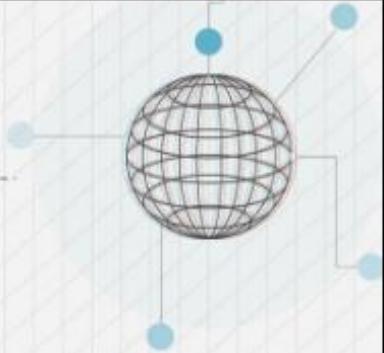


- Là một nền tảng phần mềm cho khả năng mở rộng máy chủ và các ứng dụng mạng.
- Được viết bằng JavaScript.
- Sử dụng kỹ thuật điều hướng theo sự kiện.
- Mô hình Non-Blocking I/O.
- Nhiều thư viện hỗ trợ thông qua **Node Packaged Modules (NPM)**.

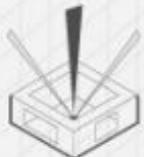




Node.js dùng để làm gì ?



- Xây dựng websocket Server.**chat server**
- Ứng dụng upload file rất nhanh trên client.
- Xây dựng Ad Server.
- Hoặc bất kỳ ứng dụng dữ liệu thời gian thực nào.





Cài đặt & Tài liệu



- Cài đặt

<http://nodejs.org>



INSTALL

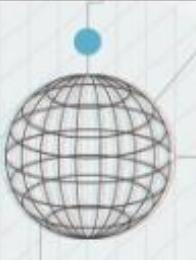
- Tài liệu

<http://nodejs.org/api>





Run “Hello World.”



[hello.js](#)

```
var http = require('http');    require modules

http.createServer(function(request,reponse){
  response.writeHead(200); status code in header
  response.write("Hello World."); response body
  response.end(); close the connection
}).listen(8080); listen for connection on this port

console.log('Listening on port 8080...');
```

node hello.js Run the server

Go url <http://localhost:8080>

-----> listening on port 8080

-----> Hello World.



I. Giới thiệu NODE.js





Giới thiệu NPM



- Dùng để quản lý các gói của Node.js
- Đi kèm và được cài đặt tự động với môi trường máy tính khi cài Node.js
- Cho phép cài đặt các gói Node.js có sẵn trên Registry Npm hoặc gỡ bỏ các gói cài đặt.
- Npm được viết hoàn toàn bằng Javascript , và chạy trên nền tảng Node.js



II. Node Package Modules (NPM)





Sử dụng <https://www.npmjs.org>



- Cài đặt

```
npm install packpage_name
```

----> npm install socket.io

- Gỡ bỏ

```
npm uninstall package_name
```

----> npm uninstall socket.io



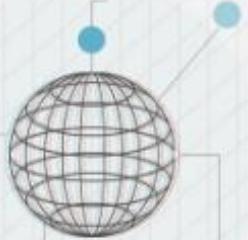
II. Node Package Modules (NPM)



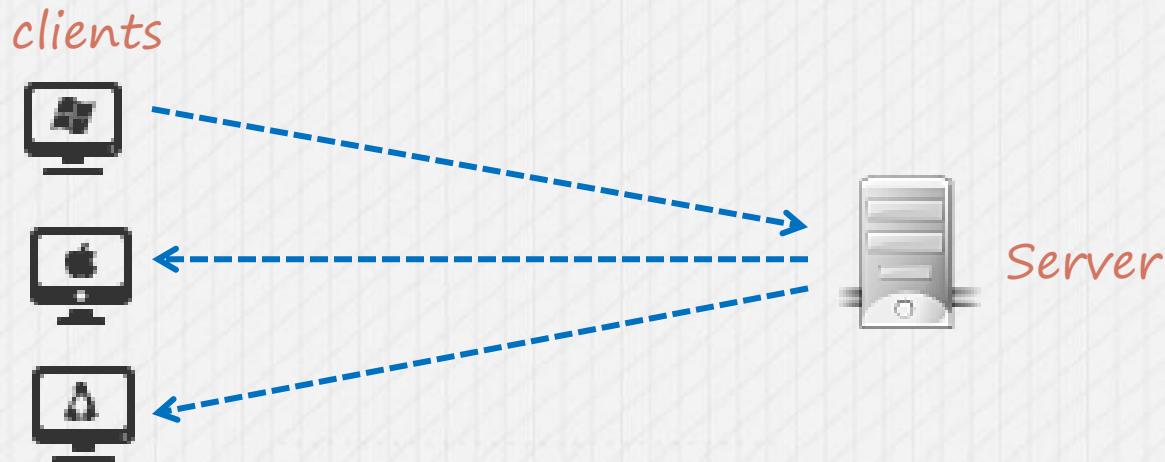


Giới thiệu Socket.io

<http://socket.io>



- Là một mô hình websocket được đóng gói để chạy với Node.js
- Giữ cho Client và Server ở trạng thái luôn kết nối.
- Khả năng phản hồi nhanh với các yêu cầu.
- Hỗ trợ các ứng dụng chạy thời gian thực.

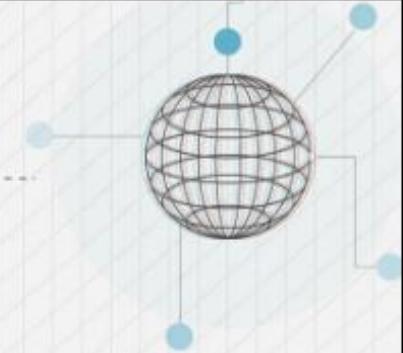


III. Socket.io Package





Cài đặt & Gỡ bỏ



- Cài đặt

```
npm install socket.io
```

- Gỡ bỏ

```
npm uninstall socket.io
```

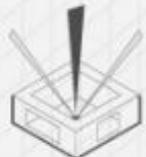


III. Socket.io Package





Kết nối Server & Client

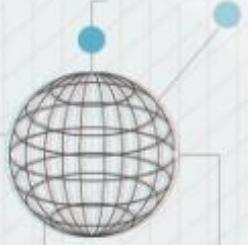


III. Socket.io Package





Kết nối Server & Client



```
var io = require('socket.io').listen(8080);
  io.set('transports',['xhr-polling']);

io.sockets.on('connection',function(socket){
  console.log('Client connected ...');
});
```

server.js

```
<script src="http://localhost:8080/socket.io/socket.io.js" ></script>
<script>
  var socket = io.connect('http://localhost:8080');
</script>
```

index.html

- Run:

node server.js

- Cancel:

Ctrl+C



III. Socket.io Package





Client gửi dữ liệu đến Server



- *Gửi:* `socket.emit('name',data)`

- *Nhận:* `socket.on('name',function(data){
}); // server`

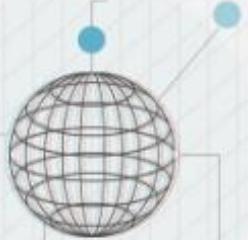


III. Socket.io Package





Client gửi dữ liệu đến Server



server.js

```
io.sockets.on('connection',function(socket){
    console.log('Client connected ...');

    socket.on('messages',function(data){
        console.log(data);
    });
});
```

listen for 'messages' events

index.html

```
<script src="http://localhost:8080/socket.io/socket.io.js" ></script>
<script>
    var socket = io.connect('http://localhost:8080');

    emit the 'messages' event on the server
    socket.emit('messages',{hello:'world'});
</script>
```



III. Socket.io Package





Server gửi dữ liệu đến clients



client



Server



clients



Server



1

2

3

clients



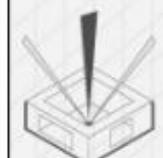
Server



1

2

3



III. Socket.io Package

1. Server gửi đến client đang tạo ra kết nối.

```
socket.emit("name",data);
```

2. Server gửi đến tất cả các clients ngoại trừ client tạo ra kết nối để bắt đầu nó.

```
socket.broadcast.emit("name",data);
```

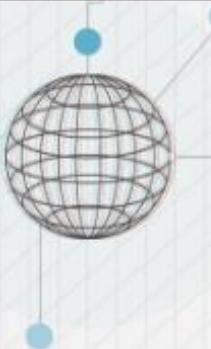
3. Server gửi đến tất cả các clients.

```
io.sockets.emit("name",data)
```





Server gửi dữ liệu đến Client



1. Server gửi đến client đang tạo ra kết nối.

```
io.sockets.on('connection',function(socket){  
    console.log('Client connected ...');  
  
    emit the 'messages' event on the client  
    socket.emit('messages',{hello:'world'});  
});
```

server.js

```
<script src="http://localhost:8080/socket.io/socket.io.js" ></script>  
<script>  
    var socket = io.connect('http://localhost:8080');  
    socket.on('messages',function(data){  
        console.log(data);  
    });  
    listen for 'messages' events  
</script>
```

index.html



III. Socket.io Package





Server phát dữ liệu đến clients



2. Server gửi đến tất cả các clients ngoại trừ client tạo ra kết nối để bắt đầu nó.

```
io.sockets.on('connection',function(socket){  
    socket.broadcast.emit ('messages' ,{hello:'world'});  
    bradcase message to all other clients connected  
});
```

server.js

```
<script src="http://localhost:8080/socket.io/socket.io.js" ></script>  
<script>  
    var socket = io.connect('http://localhost:8080');  
    socket.on('messages',function(data){  
        console.log(data);  
    });  
</script>
```

index.html



III. Socket.io Package





Server phát dữ liệu đến clients



3. Server gửi đến tất cả các clients.

```
io.sockets.on('connection',function(socket){  
    io.sockets.emit ('messages' ,{hello:'world'});  
    bradcse message to all other clients connected  
});
```

server.js

```
<script src="http://localhost:8080/socket.io/socket.io.js" ></script>  
<script>  
    var socket = io.connect('http://localhost:8080');  
    socket.on('messages',function(data){  
        console.log(data);  
    });  
</script>
```

index.html



III. Socket.io Package

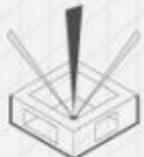




Giới thiệu Mysql package



- Dùng để kết nối và xử lý mysql.
- Được viết bằng JavaScript
- Không yêu cầu biên dịch





Cài đặt

<https://www.npmjs.org/package/mysql>



- Cài đặt

```
npm install mysql
```

- Gỡ bỏ

```
npm uninstall mysql
```





Kết nối với Mysql



server.js

```
var connection = require('mysql').createConnection({  
    host      : 'localhost',  
    user      : 'root',  
    password  : '123',  
    database  : 'database_name'  
});  
connection.connect();
```



IV. Mysql Package





Truy vấn database có 3 cách



Cách 1:

```
connection.query( sql, function(err,results){ } );
```

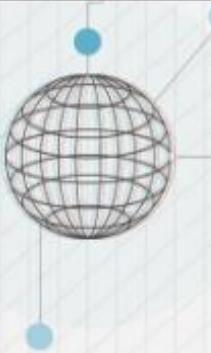
- ❖ Sql : *Câu truy vấn sql*

```
connection.query('SELECT * FROM users ORDER BY id ASC', function(err, results) {  
    if (err) throw err;  
    else {  
        console.log(results);  
    }  
});
```





Truy vấn database có 3 cách



Cách 2:

```
connection.query(sql, selectionArgs, function(err,results){ });
```

- ❖ Sql : *Câu truy vấn sql.*
- ❖ selectionArgs: *Mảng các tham số phụ cho câu điều kiện.*

```
connection.query('SELECT * FROM users WHERE id = ?',[1],function(err, results) {  
    if (err) throw err;  
    else {  
        console.log(results);  
    }  
});
```





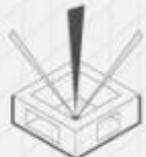
Truy vấn database có 3 cách



Cách 3:

```
connection.query(sql)
  .on('error', function(err) {
    handle error
  }) .on('result', function(row) {
    receive data
  }).on('end', function() {
    all rows have been received
});
```

```
connection.query('SELECT * FROM users')
  .on('error', function(err) {
    throw err;
  })
  .on('result', function(row) {
    console.log(row);
  })
  .on('end', function() {
    console.log('end');
  });
});
```





INSERT



```
connection.query(sql,data, function(err,results){ });
```

- ❖ Sql : Câu truy vấn sql
- ❖ data: dữ liệu cần thêm

```
var data = {name:'zendvn'};
connection.query('INSERT INTO users SET ?',data, function(err, result) {
  if (err) throw err;
  else {
    console.log(result);
  }
});
```





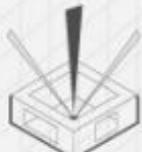
UPDATE



```
connection.query(sql, selectionArgs, function(err,results){ });
```

- ❖ Sql : *Câu truy vấn sql.*
- ❖ selectionArgs: *Mảng các tham số phụ cho câu điều kiện.*

```
var data = ['nodejs',2];
connection.query('UPDATE users SET name = ? WHERE id = ?',data,
  function(err, result) {
    if (err) throw err;
    else {
      console.log(result);
    }
});
```





DELETE



```
connection.query(sql, selectionArgs, function(err, results){ });
```

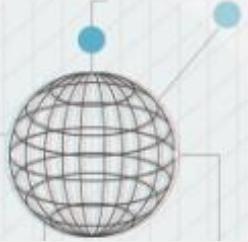
- ❖ Sql : *Câu truy vấn sql.*
- ❖ selectionArgs: *Mảng các tham số phụ cho câu điều kiện.*

```
connection.query('DELETE FROM users WHERE id = ?',[2], function(err, result) {  
    if (err) throw err;  
    else {  
        console.log(result);  
    }  
});
```





Xây dựng ứng dụng chat



Chức năng ?

- Nhiều người chat với nhau
- 2 người chat với nhau

CHAT ROOM

LOGOUT	
Hello, Isco	
Iker Casillas	Online
Varane	Online
Ramos	Online
Marcelo	
Di Maria	
Ronaldo	

Isco hello 2014-04-24 14:29:47

Isco This is a room 2014-04-24 14:29:59

Ramos hello everybody 2014-04-24 14:30:58

Write your message... Send

Ramos X

hello

😊😊



V. Xây dựng ứng dụng Chat





Kiến thức

Cần gì ?

- Html
- Css
- Jquery
- Mysql
- Php
- Node.js



V. Xây dựng ứng dụng Chat





Cài đặt

Install ?

- xampp.
- node.js
 - socket.io package
 - mysql package
 - date-format package

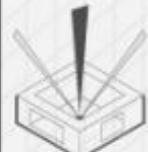
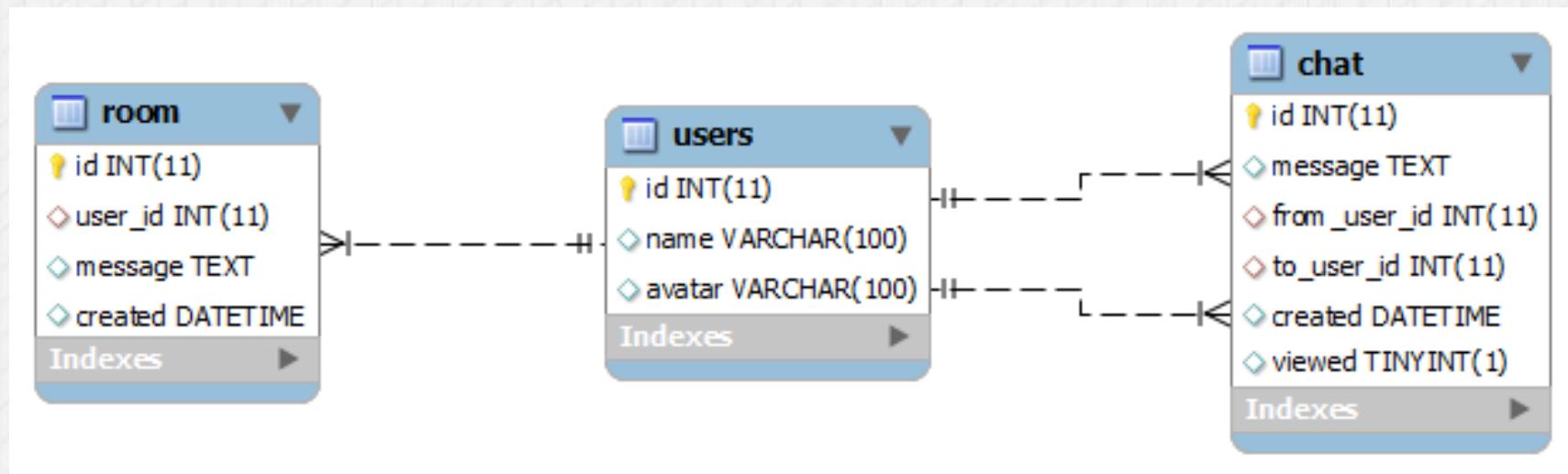


V. Xây dựng ứng dụng Chat





Cơ sở dữ liệu



V. Xây dựng ứng dụng Chat





Xây dựng ứng dụng



3 nhóm chức năng

- o Đăng nhập – đăng xuất
- o Chat giữa nhiều người (chat room)
- o Chat giữa 2 người



V. Xây dựng ứng dụng Chat





Chat giữa 2 người



clients



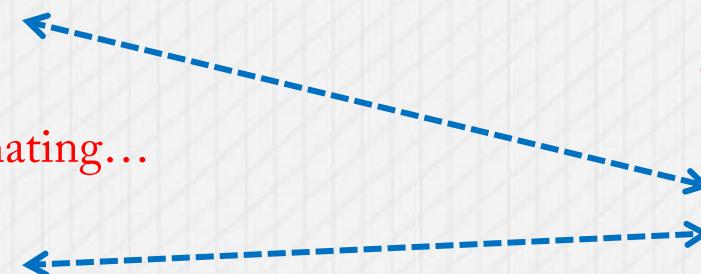
1



2



3



Server



V. Xây dựng ứng dụng Chat

