

AJAX - Tương lai của ứng dụng Web

Bài 1: Các thế mạnh của AJAX.

Các thế mạnh của AJAX:

Bài này bắt đầu đi vào những cái nền tảng nhất, cần học chậm và hiểu được vấn đề cốt yếu:

Ajax là viết tắt của **Asynchronous JavaScript and XML** – công nghệ kết hợp hai tính năng mạnh của JavaScript được các nhà phát triển đánh giá rất cao:

- Gửi yêu cầu (request) đến server
- Phân tách và làm việc với XML

Các ứng dụng Ajax xoay quanh một tính năng có tên là XMLHttpRequest.

Ở các ứng dụng web truyền thống, khi người dùng có một cần thay đổi dữ liệu trên trang Web, yêu cầu thay đổi được gửi về server dưới dạng HTTP request (hay còn gọi postback), server sẽ xử lý yêu cầu này và gửi trả response chứa các thông tin dưới dạng HTML và CSS, trang HTML này sẽ thay thế trang cũ. Quy trình này được mô tả là nhấp-chờ và tải lại (click-wait-and-refresh): ví dụ người dùng sau khi nhấn một nút “Submit” trên trang web phải chờ cho đến khi server xử lý xong mới có thể tiếp tục công việc.

Ajax cho phép tạo ra một Ajax Engine nằm giữa giao tiếp này. Khi đó, các yêu cầu gửi request và nhận response do Ajax Engine thực hiện. Thay vì trả dữ liệu dưới dạng HTML và CSS trực tiếp cho trình duyệt, web server có thể gửi trả dữ liệu dạng XML và Ajax Engine sẽ tiếp nhận, phân tách và chuyển hóa thành XHTML + CSS cho trình duyệt hiển thị.

Việc này được thực hiện trên client nên giảm tải rất nhiều cho server, đồng thời người sử dụng cảm thấy kết quả xử lý được hiển thị tức thì mà không cần nạp lại trang. Mặt khác, sự kết hợp của các công nghệ web như CSS và XHTML làm cho việc trình bày giao diện trang web tốt hơn nhiều và giảm đáng kể dung lượng trang phải nạp. Đây là những lợi ích hết sức thiết thực mà Ajax đem lại.

Rõ ràng điểm khác biệt là thay vì phải tải cả trang web thì với AJAX ta chỉ cần tải về phần của trang Web muốn thay đổi. Điều này giúp cho ứng dụng web của phản hồi nhanh hơn, thông minh hơn. Ngoài ra, điểm đặc biệt quan trọng trong công nghệ Ajax nằm ở chỗ Asynchronous - bất đồng bộ - tức là gửi yêu cầu của mình tới server và tiếp tục thực thi tác vụ hiện tại mà không cần chờ trả lời. Khi nào server xử lý xong yêu cầu của, nó sẽ báo hiệu và ta có thể đón để thể hiện những thay đổi cần thiết

Bài 2: Các công nghệ trong AJAX - CSS - Giới thiệu.

Từ bài này, chúng ta sẽ tìm hiểu các công nghệ trong AJAX và mối liên hệ giữa chúng.

AJAX là một tập hợp các công nghệ bổ sung lẫn nhau. JavaScript có vai trò chất keo kết dính các ứng dụng lại với nhau. Giao diện người dùng được tạo và tải nạp bằng cách dùng JavaScript để điều khiển Document Object Model, tạo và tổ chức biểu diễn dữ liệu cho người dùng, đồng thời xử lý các tương tác trên chuột và bàn phím.

Cascading Style Sheets (CSS) cung cấp một sự nhất quán trên cảm quan “look and feel” cho ứng dụng và khả năng thao tác mạnh mẽ với DOM. Đối tượng XMLHttpRequest (hay một cơ chế tương đương nào đó) được dùng để liên lạc một cách bất đồng bộ với server, đảm bảo việc gửi yêu cầu người dùng và tải nạp dữ liệu trong khi người dùng vẫn làm việc.

Cascading Style Sheet – CSS

Phần này khá là dài, vnit sẽ viết cố gắng thật dễ hiểu.

Cascading Style Sheet – tạm dịch là bảng kiểu xếp chồng - là một phần không thể thiếu trong thiết kế Web, nó được dùng rất nhiều trong các ứng dụng Web truyền thống cũng như trong Ajax. Một stylesheet đưa ra cách kiểm soát các loại định dạng trực quan, nó có thể được áp dụng cho các thành phần riêng lẻ trên các trang.

Hơn nữa, cho các thành phần định dạng trực quan như màu sắc, lề, hình nền, tính trong suốt, kích cỡ, stylesheet có thể xác định cách mà các phần tử được bố trí quan hệ với các phần tử khác và tương tác với người dùng, cho phép các hiệu ứng khá mạnh mẽ.

Trong ứng dụng Web truyền thống, stylesheet cung cấp một cách hiệu quả để xác định cách thể hiện vị trí và có thể được dùng lại trong nhiều trang web khác nữa. Với AJAX, stylesheet cung cấp một “kho chứa” các giao diện xác định trước có thể áp dụng cho các phần tử động với độ dài các đoạn mã nguồn là nhỏ nhất.

CSS định dạng một trang web theo ba cách :

1. Sử dụng trực tiếp kèm với các thẻ HTML (Inline Style Sheet)
2. Định nghĩa trong một trang web (Internal Style Sheet).
3. Định nghĩa thành một file CSS riêng (External Style Sheet). Trang web của chúng ta sẽ tham chiếu đến file CSS này.

Một quy tắc định dạng và bố trí gồm có hai phần: thành phần lựa chọn - selector và phần khai báo - style declaration. Selector đặc tả các phần tử được định dạng và bố trí, và style declaration khai báo các thuộc tính định dạng sẽ được áp dụng. Giả sử muốn tạo ra các dòng text trong level-

1 heading trong tài liệu (đó là đoạn nằm trong thẻ <H1>) có màu đỏ.
Có thể khai báo thuộc tính CSS như sau:

Code:

```
h1 {color: red}
```

Chúng ta cũng nên phân tích:

Các ưu điểm của CSS trong thiết kế web

a. CSS giúp tiết kiệm được rất nhiều thời gian và công sức cho việc thiết kế web.

Style trong phiên bản HTML 4.0 qui định cách thức thể hiện các thẻ. Style thường được lưu trong các file nằm ngoài trang web. Chúng giúp thay đổi cách thức định dạng và cách bố trí các trang web chỉ bằng cách thay đổi riêng file CSS.

b. CSS cho phép điều khiển cách định dạng và cách bố trí của cùng lúc nhiều trang web với chỉ duy nhất một lần thay đổi tại một vị trí.

c. Có thể định nghĩa nhiều style vào một thẻ HTML .

CSS cho phép đưa các thông tin định nghĩa thẻ thông qua nhiều con đường khác nhau. Style có thể được qui định ở trong chỉ một thẻ HTML, được qui định trong một trang web hoặc ở trong một file CSS bên ngoài.

d. Thứ tự áp dụng các định dạng

Như trên đã nói, có thể sử dụng nhiều cách khác nhau để làm CSS. Điều gì sẽ xảy ra nếu áp dụng nhiều cách định dạng cho một thẻ HTML? Theo một cách chung nhất ra có thể nói các style sẽ được “xếp tầng” (cascade). Việc xếp tầng này tuân theo thứ tự ưu tiên giảm dần như sau:

- Inline Style (Style được qui định trong một thẻ HTML cụ thể)
- Internal Style (Style được qui định trong phần <HEAD> của một trang HTML)
- External Style (style được qui định trong file CSS ngoài)
- Browser Default (thiết lập mặc định của trình duyệt)

Bài sau chúng ta sẽ đi vào: Cú pháp cơ bản của CSS.

Bài 3: Các công nghệ trong AJAX - CSS - Cú pháp & thuộc tính CSS Style.

Cú pháp cơ bản của CSS

Cú pháp của CSS gồm ba thành phần:

- Thành phần lựa chọn (thường là một thẻ HTML) (Selector)
- Thuộc tính (Property)
- Giá trị (Value)

Thể hiện của cú pháp CSS

Trích:

```
Selector {  
Property1: Value1;  
Property2: Value2;  
}
```

Selector có thể là các thẻ/nhóm thẻ HTML, các lớp khai báo, hay bằng định danh duy nhất của phần tử.

Khi chèn các đoạn mã CSS vào trang web, trình duyệt sẽ hiển thị trang web theo cách CSS đã qui định cho nó, có ba cách để chèn CSS vào trang web.

a. External Style Sheet (sử dụng file CSS được định nghĩa thành trong file riêng)

Mỗi trang web sử dụng file CSS ngoài này đều phải sử dụng thẻ <LINK>. Thẻ <LINK> được đặt bên trong thẻ <HEAD>.

Trích:

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

b. Internal Style Sheet (định nghĩa các style sheet ngay trong trang web)

Trong trường hợp mỗi trang web của sử dụng các định dạng khác nhau, dùng Internal Style Sheet. Để định nghĩa Internal Style Sheet, sử dụng thẻ <STYLE> đặt bên trong thẻ <HEAD>.

Trích:

```
<head>  
<style type="text/css">  
hr {color: sienna}
```

```
p {margin-left: 20px}
body {background-image: url("images/back40.gif")}
</style>
</head>
```

c. Internal Style Sheet (style được qui định ngay trong mỗi thẻ HTML)

Đây là phương pháp kém hiệu quả nhất, không nên sử dụng phương pháp này vì đã làm mất các ưu điểm của CSS.

Trích:

```
<p style="color: sienna; margin-left: 20px">
This is a paragraph
</p>
```

Các thuộc tính của CSS Style

Mỗi phần tử trong trang HTML có thể được qui định theo nhiều kiểu. Một phần text của một phần tử có thể được quy định theo các thuộc tính color, font size, độ đậm của phông, và kiểu chữ sử dụng. Có rất nhiều tùy chọn được áp dụng cho thuộc tính trên. Ví dụ để qui định cho một paragraph:

Trích:

```
.robotic{
font-size: 14pt;
font-family: courier new, courier, monospace;
font-weight: bold;
color: gray;
}
```