HTML DOM và Validate

Khóa học: Webapp Building With JavaScript

Mục tiêu

- Trình bày khái niệm HTML DOM
- Trình bày khái niệm “Cây đối tượng” (Tree of Objects) trong DOM
- Trình bày cách sử dụng DOM trong JavaScript
- Trình bày những thao tác cơ bản với DOM
- Trình bày kỹ thuật validate trong HTML
- Trình bày một số phương pháp validate với Javascript
- Sử dụng các thuộc tính validate trong HTML
- Sử dụng biểu thức chính quy để validate

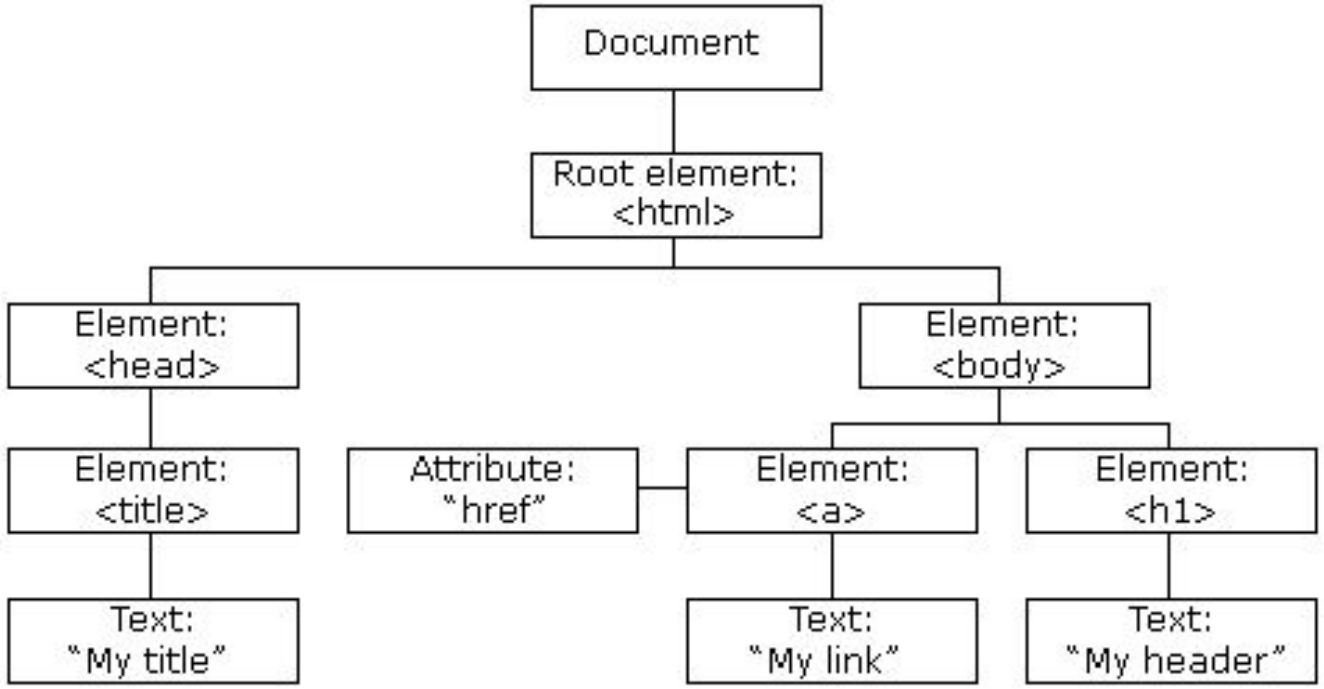
HTML DOM

Khái niệm HTML DOM

- Mô hình đối tượng tài liệu (DOM) biểu diễn dữ liệu của các đối tượng (bao gồm cấu trúc và nội dung) của tài liệu trên web.
- HTML DOM là một tập hợp các câu lệnh lập trình (API) phục vụ những thao tác với đối tượng trong HTML.
  - Định nghĩa cấu trúc logic của trang HTML
  - Cung cấp những phương thức truy cập và làm việc với các đối tượng HTML

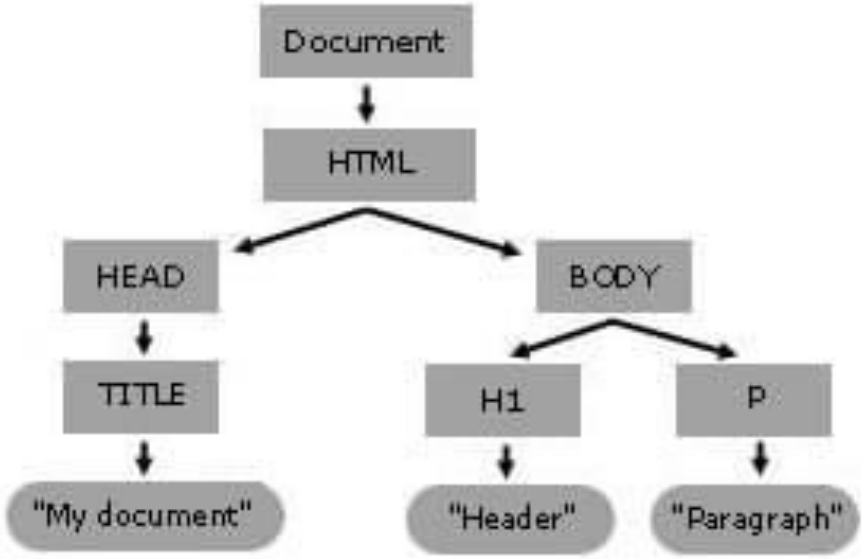
Cây đối tượng

- Khung sườn của một trang HTML là các thẻ (tag).
- Với DOM, mọi thẻ HTML đều là các đối tượng (object).
- Các thẻ lồng nhau là "con" của thẻ bao quanh (Văn bản bên trong thẻ cũng là một đối tượng)



Ví dụ “Cây đối tượng”

```
<html>
<head>
  <title>My Document</title>
</head>
<body>
  <h1>Header</h1>
  <p>Paragraph</p>
</body>
</html>
```



Thao tác cơ bản

Với DOM, chúng ta có thể thực hiện những thao tác sau đây trên một trang web:

- Tham chiếu đến một thành phần (đối tượng) trên trang web. (Nói cách khác, truy cập đến một thành phần trên web)
- Thay đổi nội dung của một thành phần trên trang web
- Thay đổi style của một thành phần trên trang web
- Thêm một thành phần mới vào trang web
- Xóa đi một thành phần đang tồn tại trong trang web
- ...

Truy cập các phần tử trên web

Khi muốn truy cập các phần tử HTML bằng JavaScript, trước tiên chúng ta phải tìm các phần tử đó.

Có những cách dưới đây:

- Tìm các phần tử HTML theo id
- Tìm các phần tử HTML theo tên thẻ (tag)
- Tìm các phần tử HTML theo tên lớp (class name)
- Tìm các phần tử HTML bằng bộ chọn CSS (CSS selector)
- Tìm các phần tử HTML bằng các tập hợp đối tượng HTML

Thuộc tính id trong HTML

- Trong HTML, thuộc tính id được dùng để xác định một id duy nhất cho một phần tử HTML (giá trị phải là duy nhất trong file HTML).
- Giá trị id có thể được dùng trong CSS hoặc JavaScript để thực hiện một tác vụ nhất định cho phần tử duy nhất có giá trị id đó.
- Lưu ý:
  - Thuộc tính **id** có thể dùng với bất kì phần tử HTML nào.
  - Giá trị **id** có phân biệt chữ thường và chữ hoa.
  - Giá trị **id** phải chứa ít nhất 1 kí tự và không chứa khoảng trắng.

Tìm các phần tử HTML theo id

Tìm phần tử HTML theo id là cách dễ nhất để tìm một phần tử HTML trong DOM.

Sử dụng lệnh `document.getElementById(<id-phần tử>)`

Ví dụ: Với mã HTML sau:

```
<p id="demo">Đây là đoạn văn demo trên web</p>
```

Để truy cập vào thẻ <p> có id là “demo”, chúng ta thực thi lệnh:

```
let demoElement = document.getElementById("demo");
```

Thao tác với phần tử HTML

Sau khi tìm được các phần tử HTML, chúng ta có thể thực hiện các thao tác sau với JavaScript:

- Truy xuất nội dung của phần tử HTML
- Thay đổi nội dung của một phần tử HTML
- Thay đổi style của một phần tử HTML

Với một phần tử HTML, chúng ta có thể thao tác với các thuộc tính sau:

- innerHTML
- innerText
- style
- ...

Thuộc tính innerHTML

Thuộc tính *innerHTML* được sử dụng để **truy xuất** hoặc **thay đổi** nội dung bất kỳ phần tử HTML nào, bao gồm <html> và <body>.

Ví dụ: `let demoElement = document.getElementById("demo");`

1. Truy xuất nội dung thẻ có id là “demo”:

```
let demoText = demoElement.innerHTML;
```

1. Thay đổi nội dung thẻ có id là “demo”:

```
demoElement.innerHTML = “Đây là nội dung mới”;
```

Thuộc tính innerText

Tương tự *innerHTML*, thuộc tính *innerText* được sử dụng để **truy xuất** hoặc **thay đổi** nội dung của phần tử HTML.

Sự khác nhau giữa innerHTML và innerText là innerText không xử lý các thẻ HTML bên trong giá trị được gán.

Ví dụ: `let demoElement = document.getElementById("demo");`

1. Thay đổi nội dung thẻ có id là “demo” với innerHTML:

```
demoElement.innerHTML = “Đây là <b>nội dung</b> mới”;
```

1. Thay đổi nội dung thẻ có id là “demo” với innerText:

```
demoElement.innerText = “Đây là <b>nội dung</b> mới”;
```

Với *innerHTML*, chữ “nội dung” sẽ được in đậm trên trang web. Trong khi với *innerText*, cặp thẻ và sẽ được hiển thị như một dạng văn bản thông thường.

Thuộc tính style

Thuộc tính *style* được sử dụng để **truy xuất** hoặc **thay đổi** style của phần tử HTML.

Ví dụ: `let demoElement = document.getElementById("demo");`

1. Thay đổi màu chữ cho nội dung trong thẻ có id là “demo”:

```
demoElement.style.color = “red”;
```

1. Thay đổi màu nền cho nội dung trong thẻ có id là “demo”:

```
demoElement.style.background = “yellow”;
```

Demo

Tìm phần từ HTML theo id
Thay đổi nội dung một phần tử HTML
Thay đổi style một phần tử HTML

Demo

Thêm phần tử vào trang web
Xóa phần tử khỏi trang web

Validation

Khái niệm Validation

- **Input Validation -** Xác thực đầu vào (còn được gọi là xác thực dữ liệu) là việc kiểm tra giá trị hợp lệ của đầu vào (input) do người dùng hoặc ứng dụng bên ngoài cung cấp.
- Xác thực đầu vào ngăn không cho dữ liệu định dạng không hợp lệ xâm nhập vào hệ thống thông tin.
- Việc xác thực đầu vào sẽ xảy ra khi dữ liệu được nhận từ một tác nhân bên ngoài, đặc biệt nếu dữ liệu từ các nguồn không đáng tin cậy.
- Ví dụ:
  - Xác thực giá trị địa chỉ email nhập vào có đúng format và nhà cung cấp hợp lệ hay không
  - Xác thực giá trị ngày/tháng/năm sinh có phù hợp hay không

Các cấp độ validation

Trong một hệ thống phần mềm, có rất nhiều thành phần có thể thực hiện công việc xác thực dữ liệu. Cụ thể:

- Xác thực dữ liệu nhập vào ở phía client (Client-side validation). Ví dụ: các input hoặc form control trên trang web, các vị trí nhập liệu trên giao diện ứng dụng di động, giao diện phần mềm desktop,...
- Xác thực dữ liệu đầu vào ở phía backend (Server-side validation). Ví dụ: Tham số đầu vào của các API,...
- Xác thực dữ liệu đầu vào ở phía cơ sở dữ liệu (database)
- .v.v.

Client-side validation

Ở phía web-client, chúng ta có hai phương pháp validate sau:

- Sử dụng một số thuộc tính (attribute) có sẵn dành cho xác thực dữ liệu đầu vào trên form. (Còn gọi là built-in HTML form validation)
- Sử dụng mã lệnh JavaScript để validate.
  - Phương pháp phổ biến là sử dụng biểu thức chính quy (regular expression) để kiểm tra các chuỗi.

HTML form validation

HTML5 có khả năng validate dữ liệu người dùng mà không cần dựa vào JavaScript. Thực hiện validation bằng cách sử dụng các thuộc tính đặc biệt trên các phần tử form.

Có thể sử dụng các thuộc tính sau:

- **required**: Chỉ định xem trường biểu mẫu có cần được điền trước khi có thể gửi biểu mẫu hay không.
- **minlength** và **maxlength**: Chỉ định độ dài tối thiểu và tối đa của dữ liệu dạng văn bản (chuỗi)
- **min** và **max**: Chỉ định giá trị tối thiểu và tối đa của các loại đầu vào số
- **type**: Chỉ định dữ liệu cần phải là một số, một địa chỉ email hay một số loại cụ thể khác.
- **pattern**: Chỉ định một biểu thức chính quy xác định một mẫu mà dữ liệu đã nhập cần tuân theo.

Demo

Sử dụng required, minlength, maxlength, min/max, type

Sử dụng JavaScript

Để xác thực các giá trị bằng JavaScript, chúng ta có thể **sử dụng biểu thức chính quy** (regular expression, viết tắt là regex). Ví dụ:

1/Kiểm tra giá trị số điện thoại (Việt Nam) có hợp lệ hay không.

```
function is valid phone number(phone number) {
    const phone number regex = /^\+?([0-9]{2})\)?[-. ]?([0-9]{4})[-. ]?([0-9]{4})$/;
    return phone_number.match(phone_number_regex);
}
```

2/ Kiểm tra email hợp lệ

```
function is valid email(email) {
    const email regex = /^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/;
    return email.match(email_regex);
}
```

Demo

Sử dụng biểu thức chính quy (regex)

Tóm tắt

Qua bài học này, chúng ta đã tìm hiểu:

- Khái niệm HTML DOM và “Cây đối tượng” (Tree of Objects) trong DOM
- Cách sử dụng DOM trong JavaScript và những thao tác cơ bản với DOM
- Hai phương pháp validate sau:
  - Sử dụng các thuộc tính (attribute) dành cho validate
    - **required**
    - **minlength** và **maxlength**
    - **min** và **max**
    - **type**
    - **pattern**
  - Sử dụng mã lệnh JavaScript để validate
    - Sử dụng biểu thức chính quy (regular expression)

Hướng dẫn

Hướng dẫn làm bài thực hành và bài tập