[Bài đọc] Tổng quan về JavaScript

Mục tiêu

Bài đọc cung cấp cho học viên những kiến thức sau:

* Định nghĩa JavaScript
* Ý nghĩa của ngôn ngữ JavaScript
* Mục tiêu của JavaScript
* Lý do nên học JavaScript

Nội dung

JavaScript là một ngôn ngữ script được dùng để tạo các script ở máy client và máy server. Các script ở máy client có thể thực thi được trong trình duyệt trong khi các script ở máy server lại thực hiện trên server.

**Giới thiệu về Javascript**

HTML lúc đầu được phát triển như là một định dạng tài liệu dùng để chuyển dữ liệu trên Internet Tuy nhiên, không lâu sau đó, trọng tâm của HTML nặng tính hàn lâm và khoa học dần chuyển hướng vào người dùng thường nhật vì ngày nay người dùng xem Internet như là một nguồn thông tin và giải trí. Các trang Web ngày càng mang tính sáng tạo và đẹp mắt hơn nhằm thu hút nhiều người dùng hơn. Nhưng thực chất các kiểu và nội dung bên trong vẫn cố định. Và người dùng hầu như không thể điều khiển trên trang Web mỗi khi nó được hiển thị.

Javascript được phát triển như là một giải pháp cho vấn đề nêu trên. Javascript là một ngôn ngữ script thật sự quan trọng đã được Sun Microsystems và Netscape phát triển. Nó được dùng để tạo các trang Web động và tương tác trên Internet. Đối với những người phát triển HTML, Javascript rất hữu ích trong việc xây dựng các hệ thống HTML có thể tương tác với người dùng.

**JavaScript là gì?**

Sun Microsystems đã phát minh thành một ngôn ngữ phức tạp và mạnh mẽ mà chúng ta đã biết đó là ngôn ngữ Java. Mặc dù Java có tính khả dụng cao nhưng nó lại phù hợp nhất đối với các nhà lập trình có kinh nghiệm và cho các công việc phức tạp hơn. Netscape Communications đã thấy được nhu cầu đối với một ngôn ngữ ‘nửa nọ nửa kia’ - một ngôn ngữ sẽ cho phép các cá nhân thiết kế các trang Web có khả năng tương tác với người dùng hoặc với Java applets nhưng sẽ đòi hỏi phải có kinh nghiệm lập trình tối thiểu.

LiveScript chỉ là một ngôn ngữ phát thảo hơn là thực tế, nhưng nó lại khiến cho người ta tin rằng nó ‘bắt cầu’ cho ý tưởng tốt hơn. LiveScript được thiết kế theo tinh thần của nhiều ngôn ngữ script đơn giản nhưng nó lại có tính khả dụng cao được thiết kế đặc biệt để xây dựng các trang Web (chẳng hạn như HTML và các form tương tác). Để giúp ‘bán chạy’ ngôn ngữ mới này, Netscape hợp tác với Sun cho ra đời ngôn ngữ Javascript. Trên thực tế, Microsoft là người tiên phong thực thi phiên bản của Javascript (còn có tên là Jscript), nhưng họ vẫn chưa tham chiếu đến những đặc tả chính thức của Javascript.

Mục tiêu của JavaScript là nhằm cung cấp cho các nhà phát triển trang Web một số khả năng và sự điều khiển trên chức năng của một trang Web. Mã Javascript có khả năng nhúng trong tài liệu HTML để điều khiển nội dung của trang Web và xác nhận tính hợp lý của dữ liệu mà người dùng nhập vào. Khi một trang hiển thị trong trình duyệt, các câu lệnh được trình duyệt thông dịch và thực thi.

**JavaScript là một ngôn ngữ script dựa trên đối tượng nhằm phát triển các ứng dụng Internet dựa trên client và server.**

**Tại sao nên học Javascript**

Hiện tại tính ứng dụng của Javascript rất lớn từ

* Lập trình web front end với các framework hàng đầu: JQuery, React, Angular, VueJS
* Backend với các framework của NodeJS
* Lập trình ứng dụng di động với React Native, Ionic, NativeScript, vv
* Lập trình game với Unity
* Lập trình robot, IoT

Xu hướng gần đây của giới lập trình là full stack developer. Để tránh việc dùng nhiều ngôn ngữ thì lựa chọn Javascript là khả dĩ nhất.

 Kết luận

Sau bài đọc, học viên đã được cung cấp kiến thức về:

* Định nghĩa JavaScript
* Ý nghĩa của ngôn ngữ JavaScript
* Mục tiêu của JavaScript
* Lý do nên học JavaScript

[Bài đọc] Cú pháp JavaScript

Mục tiêu

Bài đọc cung cấp cho học viên những kiến thức sau:

* Cú pháp Javascript
* Câu lệnh Javascript
* Giá trị
* Literal
* Biến
* Từ khoá
* Toán tử
* Biểu thức
* Chú thích
* Đặt tên

Nội dung

Cú pháp là gì?

Cú pháp Javascript bao gồm những quy tắc để viết mã Javascript. Chúng ta phải tuân thủ cú pháp khi xây dựng một chương trình.

Một chương trình máy tính (computer program) là một danh sách các hướng dẫn (instruction) được máy tính thực thi (execute).

Trong các ngôn ngữ lập trình thì các hướng dẫn này được gọi là câu lệnh.

Trong Javascript, các câu lệnh được phân cách bởi dấu hai chấm (;).

Ví dụ về các câu lệnh:

var x, y, z;  
x = 5;  
y = 6;  
z = x + y;

Câu lệnh Javascript

Các câu lệnh trong Javascript được tạo thành từ các yếu tố sau: Giá trị (value), Toán tử (Operator), Biểu thức (Expression), Từ khoá (Keyword) và Chú thích (Comment).

Giá trị

Javascript quy định 2 kiểu giá trị: Giá trị cố định (fixed value) và giá trị có thể thay đổi (variable value).

Các giá trị cố định được gọi là literal. Các giá trị có thể thay đổi được gọi là biến (variable).

Literal

Số. Bao gồm số nguyên và số thập phân. Ví dụ, 10 hoặc 10.5.

Chuỗi. Được viết trong dấu nháy (đơn hoặc kép). Ví dụ: "Nam", hoặc 'Hà Nội'.

Biến (variable)

Biến được sử dụng để lưu trữ giá trị.

Một biến cần được khai báo trước khi sử dụng.

Để khai báo một biến trong Javascript, chúng ta sử dụng từ khoá let (trước đây là var, hiện tại ít được sử dụng). Ví dụ:

var pass;  
let math;  
let phys;  
let mech;

Để gán một giá trị cho biến, chúng ta sử dụng dấu bằng (=). Dấu (=) được gọi là toán tử gán (assignment operator). Ví dụ:

pass = 4.5*;*  
*math = 9*  
*phys = 9.5*  
*mech = 8*

Toán tử

Có nhiều loại toán tử khác nhau, chẳng hạn như toán tử số học, toán tử so sánh, toán tử logic... Chúng ta sẽ tìm hiểu về các loại toán tử ở phần sau.

Trong ví dụ sau đây, chúng ta sử dụng các toán tử số học (hay còn gọi là toán học) để tính toán.

(5 + 6) \* 10

Biểu thức

Biểu thức là kết hợp của giá trị, biến, toán tử. Tính chất quan trọng nhất của biểu thức là nó sẽ trả về một kết quả sau khi được đánh giá xong.

Chẳng hạn: *5 \* 10* là một biểu thức. Giá trị của biểu thức này khi được đánh giá sẽ là *50*.

Biểu thức cũng có thể chứa biến, chẳng hạn: *x \* 10*

Từ khoá

Từ khoá là danh sách các từ được quy định sẵn trong ngôn ngữ lập trình. Các từ này được dùng để quy định một hành động được thực thi.

Chẳng hạn, từ khoá let được dùng để thực hiện hành động khai báo một biến mới.

Chúng ta sẽ làm quen dần với các từ khoá khác của Javascript.

Chú thích

Chú thích là đoạn mã thường được dùng để giải thích, làm rõ một câu lệnh khác.

Chú thích không được thực thi (bị bỏ qua khi chạy chương trình).

Trong Javascript, chúng ta có 2 cách để viết chú thích.

Viết chú thích trên một dòng. Sử dụng 2 dấu gạch chéo:

*// Đây là một chú thích trên 1 dòng*

Viết chú thích trên nhiều dòng. Sử dụng dấu /\* và \*/:

*/\**  
*Đây cũng là một chú thích.*  
*Nhưng có thể viết trên nhiều dòng.*  
*\*/*

Đặt tên

Trong các chương trình máy tính, chúng ta thường xuyên phải đặt tên. Chẳng hạn như đặt tên cho biến, đặt tên cho hàm...

Việc đặt tên là rất quan trọng và cần phải tuân thủ các quy tắc và quy ước.

Quy tắc đặt tên:

* Tên phải bắt đầu bằng ký tự chữ cái (từ a-z hoặc A-Z), ký tự gạch dưới (\_), hoặc dấu đô-la ($). Không được bắt đầu tên bằng một ký tự nào khác.
* Tên chỉ có thể chứa chữ cái (viết hoa và viết thường), chữ số, dấu gạch dưới và dấu đô-la. Không được phép sử dụng các ký tự đặc biệt (chẳng hạn như dấu # hoặc ! hoặc ?...) trong tên.
* Javascript phân biệt chữ hoa và chữ thường. Có nghĩa là một biến có tên là myName sẽ khác với một biến có tên là myname.

[Bài đọc] Các cách nhúng JavaScript vào trang web

Mục tiêu

Bài đọc cung cấp cho học viên cách khác nhau để nhúng mã Javascript vào một trang web.

Nội dung

Có một số cách khác nhau để nhúng mã Javascript vào một trang web:

* Sử dụng thẻ <script> bên trong mã HTML
* Sử dụng file Javascript riêng biệt
* Viết mã Javascript ngay trong thẻ các thẻ html

Thẻ <script>

Chúng ta có thể sử dụng thẻ <script> để nhúng mã Javascript như sau:

<script>  
     document.getElementById("demo").innerHTML = "My First JavaScript";  
</script>

Đoạn mã Javascript trên sẽ được thực thi và ghi dòng chữ "My First JavaScript" vào trong thẻ có id là "demo".

Chúng ta có thể đặt thẻ <script> bên trong thẻ <head> hoặc vào bất cứ vị trí nào mong muốn bên trong thẻ <body>.

**Lưu ý:** Các đoạn mã Javascript thường được đặt ở cuối thẻ <body> nhằm tăng tốc độ hiển thị của trang web. Tại sao lại thế?

Trong ví dụ sau đây, đoạn mã Javascript được đặt trước thẻ đóng của thẻ <body>:

**Lưu ý:** Trong ví dụ trên chúng ta có sử dụng khái niệm hàm (function) và sự kiện (event), sẽ được đề cập đến ở phần sau.

File Javascript riêng biệt

Chúng ta cũng có thể viết mã Javascript vào các file riêng biệt, sau đó nhúng vào trang HTML sử dụng thẻ <script>. Các file Javascript được lưu với phần đuôi là .js.

Chẳng hạn, đây là nội dung của file **my-script.js**:

function myFunction() {  
 document.getElementById("demo").innerHTML = "Paragraph changed.";  
}

Chúng ta có thể nhúng file này vào trong trang HTML như sau:

<!DOCTYPE html>  
<html>  
    <body>  
        <script src="myScript.js"></script>  
    </body>  
</html>

**Lưu ý:** Nên nhúng file Javascript ở phần cuối của tài liệu HTML để tăng tốc độ hiển thị của trang web.

Các ưu điểm của việc sử dụng các file Javascript rriêng biệt:

* Tách rời mã Javascript và mã HTML khiến cho việc viết mã trở nên dễ dàng hơn, dễ bảo trì, dễ đọc
* Các file Javascript này thường được lưu trong bộ cache của trình duyệt, khiến cho việc tải trang trở nên nhanh chóng hơn
* Có thể tái sử dụng được các file Javascript cho nhiều trang web khác nhau, thậm chí là nhiều website khác nhau.

Viết mã Javascript ngay khi khai báo các thẻ html

Trong ví dụ sau đây, chúng ta viết mã HTML ngay trong thuộc tính onclick của thẻ button (onclick là một sự kiện - chúng ta sẽ tìm hiểu về các sự kiện ở phần sau).

**Lưu ý:** Việc viết mã Javascript trực tiếp ở trong các thẻ html không được khuyến khích và ít được dùng. Lí do tại sao?

Kết luận

Sau bài đọc, học viên đã hiểu rõ và phân biệt được 3 cách để nhúng mã Javascript vào một trang web.

[Bài đọc] Chương trình JavaScript đơn giản

Mục tiêu

Bài đọc cung cấp cho học viên những kiến thức sau:

Ý nghĩa của JavaScrip:

* JavaScript có thể làm thay đổi nội dung file HTML
* JavaScript có thể làm thay đổi thuộc tính của các thẻ HTML

Nội dung

Javascript có thể làm thay đổi nội dung HTML

Một trong nhiều phương thức của HTML JavaScript là getElementById(). Ví dụ dưới đây sử dụng nó để "tìm" một phần tử HTML (với id là "demo") và thay đổi nội dung  html bên trong (innerHTML) nó thành "Hello JavaScript":

JavaScript có thể thay đổi thuộc tính thẻ HTML

Ví dụ dưới đây thay đổi một hình ảnh HTML bằng cách thay đổi thuộc tính src (source) của thẻ <img>:

Tham khảo: <https://www.w3schools.com/js/js_intro.asp>

Kết luận

Sau bài đọc, học viên đã hiểu được 2 ý nghĩa của JavaScript với HTML là:

* JavaScript có thể làm thay đổi nội dung file HTML
* JavaScript có thể làm thay đổi thuộc tính của các thẻ HTML

[Bài đọc] Quy tắc đặt tên file

Mục tiêu

Bài đọc cung cấp cho học viên những quy tắc đặt tên file hợp lý:

* Tên file và lưu ý khi lưu file
* Sử dụng các phần mở rộng file thích hợp khi lưu file
* Không bao giờ sử dụng dấu cách trong tên file
* Tên file có phân biệt chữ hoa, chữ thường
* Tên file ngắn

Nội dung

Cho đến bây giờ, có thể bạn đã không phải nghĩ đến cách bạn đặt tên file của mình. Khi làm việc trên web, các loại máy tính khác nhau phải làm việc cùng nhau. Điều này có nghĩa là một số công ước phải được tuân thủ.

Tên file

Tên file gồm một tập các ký tự trong bảng chữ thường (a-z) và có thể bao gồm dấu gạch dưới (\_) hoặc dấu gạch ngang (-), nhưng không có dấu chấm câu bổ sung.

Chú ý khi lưu các file của bạn

* Luôn luôn nhớ đặt tên file của bạn một cách phù hợp. Tên tập tin nên ngắn gọn nhưng rõ ràng.
* Chú ý đến nơi chính xác bạn đang lưu các file của bạn. Nếu bạn không biết file ở đâu, bạn sẽ không thể liên kết đến file đó.

Luôn sử dụng các phần mở rộng file thích hợp khi lưu file

Tất cả các file (ngay cả trên máy Mac) đều có đuôi file mở rộng. Đảm bảo bạn đang sử dụng đúng phần mở rộng khi lưu file.

* Các file HTML kết thúc bằng .html
* Các file CSS kết thúc bằng .css
* file hình ảnh kết thúc bằng .jpg, .png hoặc .gif, tùy thuộc vào kiểu file đó.
* Các file JavaScript kết thúc bằng .js

Không bao giờ sử dụng dấu cách trong tên file

* Trên máy Mac hoặc PC, bạn có thể lưu file có tên "my awesome.html". Điều này có thể sẽ không hoạt động trên một trang web.
* Sử dụng dấu gạch ngang trong tên file của bạn thay vì để chỉ khoảng trống. Tên file của bạn sẽ là "my-awesome-website.html" thay thế.

Không sử dụng ký tự đặc biệt trong tên file

* Các ký tự như ?,%, #, /,:,;, •, vv đều có ý nghĩa đặc biệt trong một url web. Do đó, chúng không nên được sử dụng trong tên file.
* Chỉ sử dụng chữ cái (a-z), số (0-9), dấu gạch dưới (\_), dấu nối (-).
* Dấu chấm (.) chỉ nên được sử dụng trong phần mở rộng

Tên file có phân biệt chữ hoa chữ thường

* Giả sử rằng tên file của bạn luôn luôn phân biệt chữ hoa chữ thường.
* Chữ viết hoa và chữ thường đều khác nhau.
* Để tránh nhầm lẫn, thực tiễn tiêu chuẩn luôn luôn sử dụng các chữ thường.

Giữ tên file của bạn ngắn

Cố gắng không làm cho tên file của bạn thật sự dài.

Ví dụ: một trang về bạn và những gì bạn đang làm ngay bây giờ:

about-me-and-what-im-doing-right-now.html

Tên này quá dài và khó gõ.

a.html

Tên này thì quá ngắn và bí ẩn.

about.html

Tên rõ ràng ngắn gọn mà vẫn đủ mô tả.

Nguồn tham khảo:

* <https://google.github.io/styleguide/jsguide.html#file-name>
* <http://web.simmons.edu/~grovesd/comm244/notes/week1/naming-conventions>

Kết luận

Sau bài đọc, học viên đã được cung cấp các quy tắc đặt tên và lưu file hợp lý như:

* Tên file và lưu ý khi lưu file
* Sử dụng các phần mở rộng file thích hợp khi lưu file
* Không bao giờ sử dụng dấu cách trong tên file
* Tên file có phân biệt chữ hoa, chữ thường
* Tên file ngắn

[Thực hành] Hiển thị đầu ra

Mục tiêu

Luyện tập hiển thị các thông báo dưới nhiều cách khác nhau

Mô tả

Trong phần này, chúng ta sẽ luyện tập các cách khác nhau để quan sát kết quả khi thực thi mã JavaScript.

Để hoàn thành bài thực hành, học viên cần:

Hướng dẫn

Javascript

function showMessage(){

alert('Xin chào!');

}

Html

<button type="button" onclick="showMessage();">Click Me!</button>

**Bài tập 2:** Hiển thị kết quả sử dụng thẻ HTML:

Javascipt

function showMessage(){

document.getElementById('messageDiv').innerHTML = 'Xin chào!';

}

Html

<div id="messageDiv"></div>

<button type="button" onclick="showMessage();">Click Me!</button>

**Bài tập 3:** Hiển thị kết quả sử dụng hàm document.write():

Javascript

function showMessage(){

document.write('Xin chào!');

}

Html

<button type="button" onclick="showMessage();">Click Me!</button>

**Bài tập 4:** Hiển thị kết quả sử dụng hàm console.log():

Javascript

function showMessage(){

console.log('Xin chào!');

}

Html

<button type="button" onclick="showMessage();">Click Me!</button>

**Lưu ý:** Bật màn hình console của trình duyệt để có thể xem được kết quả.

[Thực hành] Nhúng JavaScript

Mục tiêu

Luyện tập các cách nhúng mã Javascript vào trong trang HTML.

Mô tả

Trong phần này, chúng ta sẽ luyện tập cách để nhúng mã JavaScript vào trong trang HTML.

Hướng dẫn

**Bài tập 1:** Tạo một trang web với một nút. Chèn mã Javascript vào trong thẻ <button> để hiển thị một lời chào khi người dùng nhấn chuột vào nút đó.

Html

<button type="button" onclick="alert('Xin chào!');">Click Me!</button>

**Bài tập 2:** Tạo một trang web với một nút. Sử dụng thẻ <script> và khai báo một hàm đơn giản. Gọi hàm này khi người dùng nhấn chuột vào nút.

Javascript

function showMessage(){

alert('Xin chào!');

}

Html

<button type="button" onclick="showMessage();">Click Me!</button>

**Bài tập 3:**Hãy tách đoạn mã Javascript ở Bài tập 2 ra một file Javascript riêng và nhúng vào trang HTML.