

BỘ LAO ĐỘNG - THƯƠNG BINH VÀ XÃ HỘI
TRƯỜNG CAO ĐẲNG NGHỀ KỸ THUẬT CÔNG NGHỆ
---- Ỡ > & Ỡ > ----



GIÁO TRÌNH

MÔ ĐUN: DCCT-JAVASCRIPT
NGHỀ: CÔNG NGHỆ THÔNG TIN
TRÌNH ĐỘ: CAO ĐẲNG

*Ban hành kèm theo Quyết định số: 245/QĐ-CĐNKTCTN ngày 23 tháng 10 năm 2020
của Hiệu trưởng Trường Cao đẳng nghề Kỹ thuật Công nghệ*



Hà Nội, năm 2021
(Lưu hành nội bộ)

TUYÊN BỐ BẢN QUYỀN:

Tài liệu này thuộc loại sách giáo trình nên các nguồn thông tin có thể được phép dùng nguyên bản hoặc trích dùng cho các mục đích về đào tạo và tham khảo.

Mọi mục đích khác mang tính lệch lạc hoặc sử dụng với mục đích kinh doanh thiếu lành mạnh sẽ bị nghiêm cấm.

MÃ TÀI LIỆU: MĐCNTT 24

Tailieu.vn

LỜI GIỚI THIỆU

Trong những năm qua, dạy nghề đã có những bước tiến vượt bậc cả về số lượng và chất lượng, nhằm thực hiện nhiệm vụ đào tạo nguồn nhân lực kỹ thuật trực tiếp đáp ứng nhu cầu xã hội. Cùng với sự phát triển của khoa học công nghệ trên thế giới, lĩnh vực Công nghệ thông tin nói chung đã có những bước phát triển đáng kể.

Chương trình dạy nghề Công nghệ thông tin đã được xây dựng trên cơ sở phân tích nghề, phân kỹ năng nghề được kết cấu theo các mô đun. Để tạo điều kiện thuận lợi cho các cơ sở dạy nghề trong quá trình thực hiện, việc biên soạn giáo trình theo các mô đun đào tạo nghề là cấp thiết hiện nay.

*Mô đun 24: **DCCT-JAVASCRIPT** là mô đun đào tạo chuyên môn nghề được biên soạn theo hình thức tích hợp lý thuyết và thực hành. Trong quá trình thực hiện, nhóm biên soạn đã tham khảo nhiều tài liệu trong và ngoài nước, kết hợp với kinh nghiệm trong thực tế.*

Mặc dầu có rất nhiều cố gắng, nhưng không tránh khỏi những khiếm khuyết, rất mong nhận được sự đóng góp ý kiến của độc giả để giáo trình được hoàn thiện hơn.

Xin chân thành cảm!

Hà Nội, ngày 23 tháng 04 năm 2021

Tham gia biên soạn

1. Chủ biên **Cù Ngọc Quỳnh** giảng viên khoa CNTT
2. Tập thể Giảng viên Khoa CNTT

Mọi thông tin đóng góp chia sẻ xin gửi về hòm thư tienphungktcn@gmail.com, hoặc liên hệ số điện thoại 0913393834-0983393834

MỤC LỤC

LỜI GIỚI THIỆU

Bài 1: TỔNG QUAN VỀ JAVASCRIPT	10
1. Giới thiệu bài học	10
2. JavaScript và hỗ trợ của trình duyệt	10
2.1. Chuẩn hoá ngôn ngữ lập trình JavaScript	10
2.2. Đặc điểm chung của JavaScript	10
2.3. Ứng dụng của JavaScript	11
3. Sơ lược về ngôn ngữ JavaScript	12
4. Thời điểm thực hiện một đoạn Script	14
4.1. Cấu trúc của đoạn Javascript	14
4.2. Javascript trong một trang HTML	14
4.3. Môi trường viết Javascript	16
5. Các biến	16
5.1. Khai báo biến	17
5.2. Quy tắc đặt tên biến	17
5.3. Một số phong cách đặt tên biến	18
6. Các kiểu dữ liệu	18
6.1. Dữ liệu kiểu string	18
6.2. Dữ liệu kiểu number	19
6.3. Dữ liệu kiểu boolean	19
6.4. Dữ liệu kiểu object	20
6.5. Dữ liệu kiểu undefined	20
6.6. Dữ liệu kiểu array	21
6.7. Cách xác định kiểu của dữ liệu	21
7. Các toán tử	22
7.1. Các toán tử số học	22
7.2. Toán tử so sánh	23
7.3. Toán tử logic	24
7.4. Toán tử điều kiện	24
7.5. Toán tử với chuỗi	24
7.6. Toán tử typeof	25
8. Các biểu thức	25
8.1. Toán hạng	25
8.2. Toán tử	26
8.3. Độ ưu tiên của các toán tử	26
8.4. Khoảng trắng giữa các toán hạng và toán tử	27

9. Mảng.....	27
10. Các câu lệnh điều khiển.....	31
11. Các lệnh vòng lặp.....	34
12. Hàm(function).....	38
12.1. Hàm là gì?	38
12.2. Phân loại hàm.....	39
12.3. Cách khai báo & gọi hàm "không có tham số"	40
12.4. Cách khai báo & gọi hàm "có tham số"	41
12.5. Gọi hàm thông qua một sự kiện	42
12.6. Lệnh return.....	43
Bài 2: CÁC ĐỐI TƯỢNG CƠ BẢN TRONG JAVASCRIPT	45
1. Các đối tượng Javascript	45
1.1. Đối tượng	45
1.2. Thuộc tính và phương thức	45
1.3. Cách dùng đối tượng	45
2. Cây phân cấp của đối tượng	45
3. Câu lệnh eval	46
4. Đối tượng String	46
5. Đối tượng Math.....	46
6. Đối tượng Date.....	47
BÀI TẬP:.....	47
Bài 3: CÁC ĐỐI TƯỢNG CỦA TRÌNH DUYỆT TRONG JAVASCRIPT	52
1. Đối tượng event – Khái niệm.....	52
2. Các sự kiện Javascript	52
3. Trình xử lý sự kiện	57
4. Các đối tượng trình duyệt thông thường	57
4.1. Mô hình đối tượng(DOM)	57
4.2. Đối tượng window.....	58
4.3. Đối tượng Document	58
4.4. Đối tượng History.....	58
5. Đối tượng Location	58
BÀI TẬP.....	59
Bài 4: XỬ LÝ FORM VÀ CÁC SỰ KIỆN CHO PHẦN TỬ CỦA FORM	63
1. Đối tượng Form.....	63
2. Đối tượng Textfield(trường văn bản).....	63
3. Đối tượng Command Button	64
4. Đối tượng Checkbox	64
5. Đối tượng Radio Button	65

6. Đối tượng ComboBox/Select.....	65
7.Kiểm tra tính hợp lệ của nội dung và các trường trên form	66
BÀI TẬP:.....	70
BÀI TẬP JAVASCRIPT TỰ GIẢI	76
TÀI LIỆU THAM KHẢO	79

Tailieu.vn

GIÁO TRÌNH MÔ ĐUN

Tên mô đun: DCCT-JAVASCRIPT

Mã mô đun: MĐCNTT 24

Vị trí, tính chất, ý nghĩa và vai trò của mô đun:

- Vị trí: Mô đun JAVASCRIPT
- Tính chất: Mô đun JAVASCRIPT áp dụng trong việc xây dựng các đoạn client-side script. Truyền đạt cho sinh viên những khái niệm quan trọng về thời điểm thực hiện của một đoạn JavaScript, biến và phạm vi sử dụng cũng như việc sử dụng các đối tượng cơ bản của JavaScript như String, Date, Math,... để xử lý các dữ liệu nhập, xuất.
- Ý nghĩa và vai trò của mô đun: Đây là mô đun đào tạo chuyên môn nghề, cung cấp cho sinh viên các kỹ năng cơ bản nhất của nghề Công nghệ thông tin.

Mục tiêu của mô đun:

- Về kiến thức:
 - + Nắm vững cú pháp của ngôn ngữ JavaScript.
 - + Xác định thời điểm một đoạn Script được thực hiện.
- Về kỹ năng:
 - + Sử dụng được các đối tượng cơ sở của JavaScript
 - + Đọc hiểu sơ đồ các đối tượng.
 - + Sử dụng các đối tượng window và document để quản lý trang web
 - + Sử dụng các đối tượng của form, truy cập và kiểm tra các dữ liệu nhập trên form.
- Về năng lực tự chủ và trách nhiệm:
 - + Tích cực ứng dụng kỹ thuật lập trình cho các ứng dụng trong thực tế.
 - + Khả năng tìm tài liệu, đọc hiểu tài liệu
 - + Khả năng làm việc nhóm

Nội dung mô đun

1. Nội dung tổng quát và phân bố thời gian:

Số TT	Tên các bài trong mô đun	Thời gian			
		Tổng số	Lý thuyết	Thực hành	Kiểm tra*
1	Bài 1: Tổng quan về Javascript 1. Giới thiệu bài học 2. JavaScript và hỗ trợ của trình duyệt	10	4	6	

	3. Sơ lược về ngôn ngữ JavaScript 4. Thời điểm thực hiện một đoạn Script 5. Các biến 6. Các kiểu dữ liệu 7. Các toán tử 8. Các biểu thức 9. Mảng 10. Các câu lệnh điều khiển 11. Các lệnh vòng lặp 12. Hàm(function)				
2	Bài 2: Các đối tượng cơ bản trong Javascript 1. Các đối tượng Javascript 2. Cây phân cấp của đối tượng 3. Câu lệnh eval 4. Đối tượng String 5. Đối tượng Math 6. Đối tượng Date	20	3	17	
3	Bài 3: Các đối tượng của trình duyệt trong Javascript 1. Đối tượng event – Khái niệm 2. Các sự kiện Javascript 3. Trình xử lý sự kiện 4. Đối tượng trình duyệt thông thường. 5. Đối tượng location	25	4	20	1
4	Bài 4: Xử lý Form và các sự kiện cho phần tử của Form 1. Đối tượng Form 2. Đối tượng Textfield(trường văn bản) 3. Đối tượng Command Button 4. Đối tượng Checkbox 5. Đối tượng radio 6. Đối tượng ComboBox/lựa chọn	34	4	29	1

	7. Kiểm tra tính hợp lệ của nội dung và các trường trên form				
	Thi kết thúc mô đun	1			1
	Cộng	90	15	72	3

Tailieu.vn

Bài 1: TỔNG QUAN VỀ JAVASCRIPT

Mã bài: MĐCNTT 24.01

Giới thiệu:

Giới thiệu về ngôn ngữ lập trình Javascript, những khái niệm cơ bản.

Mục tiêu:

- Trình bày được những khái niệm quan trọng về thời điểm thực hiện của một đoạn JavaScript, biến và phạm vi sử dụng cũng như việc sử dụng các đối tượng cơ bản của JavaScript như String, Date, Math,... để xử lý các dữ liệu nhập, xuất.
- Nắm vững cú pháp của ngôn ngữ JavaScript
- Xác định thời điểm một đoạn Script được thực hiện
- Sử dụng được các đối tượng cơ sở của JavaScript

Nội dung chính:

1. Giới thiệu bài học

- Ngôn ngữ lập trình Javascript được giới thiệu đầu tiên vào năm 1995. Mục đích là để đưa những chương trình vào trang web ở trình duyệt Netscape Navigator - một trình duyệt web phổ biến những năm 1990.

- JavaScript được phát triển bởi [Brendan Eich](#) tại Hãng truyền thông Netscape với cái tên đầu tiên là Mocha, rồi sau đó đổi tên thành LiveScript, và cuối cùng thành JavaScript. Có lẽ việc đổi tên như vậy là để giúp JavaScript được chú ý nhiều hơn. Bởi tại thời điểm này, Java đang được coi là một hiện tượng và trở nên phổ biến.

2. JavaScript và hỗ trợ của trình duyệt

2.1. Chuẩn hoá ngôn ngữ lập trình JavaScript

Sau khi ngôn ngữ lập trình JavaScript được chấp nhận sử dụng bên ngoài Netscape, nó được chuẩn hoá bởi một tài liệu tên là ECMAScript bởi [Ecma International](#). Do đó, bạn có thể gọi nó là JavaScript hay ECMAScript. Riêng với bản thân thì mình thích gọi nó là JavaScript hơn. Đơn giản vì mình thấy JavaScript phát âm dễ hơn và nhanh hơn.

Tại thời điểm bài viết này được xuất bản (04/2017), phiên bản ECMAScript mới nhất là ECMAScript 2016 hay ECMA-262. Do đó, đã có rất nhiều sự khác biệt giữa JavaScript hiện tại so với phiên bản đầu tiên của nó.

Trong các bài viết sau, mình sẽ chủ yếu giới thiệu với bạn những kiến thức nguyên thủy về JavaScript. Qua đó, bạn có thể tự tìm hiểu những chuẩn mới của JavaScript [một cách dễ dàng](#).

2.2. Đặc điểm chung của JavaScript

- Là ngôn ngữ lập trình bậc cao (high-level) giống như: C/C++, Java, Python, Ruby,... Nó rất gần với ngôn ngữ tự nhiên của con người. Trong khi ngôn ngữ lập trình bậc thấp (low-level) như: Assembly... sẽ gần với máy tính hơn.
- Là ngôn ngữ lập trình động (dynamic programming language): như Python, Ruby, Perl,... Chúng được tối ưu hoá nhằm nâng cao hiệu suất cho lập trình

viên. Trong khi ngôn ngữ lập trình tĩnh (static programming language): như C/C++,... lại được tối ưu hoá để nâng cao hiệu suất cho phần cứng máy tính.

- Là ngôn ngữ lập trình kịch bản (scripting language): nghĩa là không cần biên dịch (compile) hay liên kết (linked) giống như ngôn ngữ lập trình biên dịch (C/C++, Java,...) mà nó sẽ được dịch tại thời điểm chạy.
- Là ngôn ngữ dựa trên đối tượng (object-based): tức nó gần giống như ngôn ngữ lập trình hướng đối tượng, ngoại trừ JavaScript không hỗ trợ tính kế thừa và đa hình.
- Là ngôn ngữ dựa trên nguyên mẫu (prototype-based): là một kiểu của lập trình hướng đối tượng, trong đó các hành vi của đối tượng được sử dụng lại.

2.3. Ứng dụng của JavaScript



JavaScript có rất nhiều ứng dụng trên nhiều môi trường và nền tảng khác nhau:

- JavaScript cùng với HTML, CSS trở thành ngôn ngữ không thể thiếu đối với website
- Ngoài ra, có rất nhiều framework JavaScript khác nhau phía front-end: Angular.js, Angular2, ReactJS, Vue.js, Ember.js, Meteor.js, backbone, ...
- JavaScript có thể sử dụng phía server với framework : Node.js
- Một số database sử dụng JavaScript như là kịch bản và là ngôn ngữ query: MongoDB, CouchDB
- JavaScript có thể dùng để xây dựng ứng dụng Desktop với framework: Electron. Những ứng dụng nổi tiếng có thể kể đến là: Atom, Visual Studio Code, GitKraken, Wordpress.com,...
- Để xây dựng ứng dụng điện thoại đa nền tảng (Android, IOS) có thể dùng: React-native

3.Sơ lược về ngôn ngữ JavaScript

JavaScript là ngôn ngữ lập trình phổ biến nhất trên thế giới trong suốt 20 năm qua. Nó cũng là một trong số 3 ngôn ngữ chính của lập trình web:

1. **HTML**: Giúp bạn thêm nội dung cho trang web.
2. **CSS**: Định dạng thiết kế, bố cục, phong cách, canh lề của trang web.
3. **JavaScript**: Cải thiện cách hoạt động của trang web.

JavaScript có thể **học nhanh** và dễ dàng áp dụng cho nhiều mục đích khác nhau, từ việc cải thiện tính năng của website đến việc chạy game và tạo phần mềm nền web. Hơn nữa, có hàng ngàn mẫu template JavaScript và ứng dụng ngoài kia, nhờ vào sự cống hiến của cộng đồng, đặc biệt là Github.

JavaScript ngày đó và bây giờ

JavaScript được tạo trong mười ngày bởi Brendan Eich, một nhân viên của Netscape, vào tháng 9 năm 1995. Được đặt tên đầu tiên là Mocha, tên của nó được đổi thành Mona rồi LiveScript trước khi thật sự trở thành JavaScript nổi tiếng như bây giờ. Phiên bản đầu tiên của ngôn ngữ này bị giới hạn độc quyền bởi Netscape và chỉ có các tính năng hạn chế, nhưng nó tiếp tục phát triển theo thời gian, nhờ một phần vào cộng đồng các lập trình viên đã liên tục làm việc với nó.

Trong năm 1996, JavaScript được chính thức đặt tên là ECMAScript. ECMAScript 2 phát hành năm 1998 và ECMAScript 3 tiếp tục ra mắt vào năm 1999. Nó liên tục phát triển thành JavaScript ngày nay, giờ đã hoạt động trên khắp mọi trình duyệt và trên khắp các thiết bị từ di động đến máy tính bàn.

JavaScript liên tục phát triển kể từ đó, có lực đạt đến 92% website đang sử dụng JavaScript vào năm 2016. Chỉ trong 20 năm, nó từ một ngôn ngữ lập trình riêng trở thành công cụ quan trọng nhất trên bộ công cụ của các chuyên viên lập trình web. Nếu bạn đang dùng internet, vậy chắc chắn bạn đã từng sử dụng JavaScript rồi.

Điều gì khiến JavaScript trở nên vĩ đại?

JavaScript có rất nhiều ưu điểm khiến nó vượt trội hơn so với các đối thủ, đặc biệt trong các trường hợp thực tế. Sau đây chỉ là một số lợi ích của JavaScript:

1. Bạn không cần một compiler vì web browser có thể biên dịch nó bằng HTML;
2. Nó dễ học hơn các ngôn ngữ lập trình khác;
3. Lỗi dễ phát hiện hơn và vì vậy dễ sửa hơn;
4. Nó có thể được gắn trên một số element của trang web hoặc event của trang web như là thông qua click chuột hoặc di chuột tới;
5. JS hoạt động trên nhiều trình duyệt, nền tảng, vâng vâng;
6. Bạn có thể sử dụng JavaScript để kiểm tra input và giảm thiểu việc kiểm tra thủ công khi truy xuất qua database;
7. Nó giúp website tương tác tốt hơn với khách truy cập;
8. Nó nhanh hơn và nhẹ hơn các ngôn ngữ lập trình khác.

Khuyết điểm của JavaScript là gì?

Mọi ngôn ngữ lập trình đều có các khuyết điểm. Một phần là vì ngôn ngữ đó khi phát triển đến một mức độ như JavaScript, nó cũng sẽ thu hút lượng lớn hacker, scammer, và những người có ác tâm luôn tìm kiếm những lỗ hổng và các lỗi bảo mật để lợi dụng nó. Một số khuyết điểm có thể kể đến là:

1. Dễ bị khai thác;
2. Có thể được dùng để thực thi mã độc trên máy tính của người dùng;
3. Nhiều khi không được hỗ trợ trên mọi trình duyệt;
4. JS code snippets lớn;
5. Có thể bị triển khai khác nhau tùy từng thiết bị dẫn đến việc không đồng nhất.

Cách hoạt động của JavaScript trên trang web là gì?

JavaScript thường được nhúng trực tiếp vào một [trang web](#) hoặc được tham chiếu qua file .js riêng. Nó là ngôn ngữ phía client, tức là script được tải về máy của khách truy cập và được xử lý tại đó thay vì phía server là xử lý trên server rồi mới đưa kết quả tới khách truy cập.

Hãy lưu ý là các trình duyệt web phổ biến cũng hỗ trợ việc người dùng có muốn tắt JavaScript hay không. Đó là lý do bạn nên biết trang web sẽ hoạt động như thế nào trong trường hợp không có JavaScript.

Điểm khác biệt giữa các ngôn ngữ lập trình khác và JavaScript là gì?

Lý do vì sao JavaScript là một trong các ngôn ngữ lập trình phổ biến nhất là nó rất linh hoạt. Trên thực tế, có nhiều lập trình viên chọn nó làm ngôn ngữ chính và chỉ sử dụng các ngôn ngữ khác trong danh sách bên dưới nếu nó cần dùng điều gì đó đặc biệt.

Hãy xem qua các ngôn ngữ lập trình phổ biến nhất bên dưới:

JavaScript	JavaScript hoặc JS sẽ giúp tăng tính tương tác trên website. Script này chạy trên các trình duyệt của người dùng thay vì trên server và thường sử dụng thư viện của bên thứ 3 nên có thể tăng thêm chức năng cho website mà không phải code từ đầu.
HTML	Viết tắt của “Hypertext Markup Language”, HTML là một trong số các ngôn ngữ lập trình phổ biến nhất trên web và xây dựng nên các khối chính của một trang web. Ví dụ về HTML tags là <p> cho đoạn văn và cho hình ảnh.
PHP	PHP là ngôn ngữ phía server, khác với JavaScript chạy trên máy client. Nó thường được sử dụng trong các hệ quản trị nội dung nền PHP như WordPress, nhưng cũng thường được dùng với lập trình back-end và có thể tạo ra kênh truyền thông tin hiệu quả nhất tới và từ database.
CSS	CSS viết tắt của “Cascading Style Sheets”, nó giúp webmaster xác định styles và định nghĩa nhiều loại nội dung. Bạn có thể làm vậy thủ công với mọi yếu tố trong HTML, nhưng nếu vậy bạn sẽ cứ lặp đi lặp lại thành phần đó mà bạn dùng ở nhiều nơi khác nhau.

Nếu xem ngôn ngữ lập trình như là việc xây ngôi nhà, HTML sẽ định dạng kiến trúc của căn nhà, CSS sẽ là thảm và tường để trang trí ngôi nhà đẹp hơn. JavaScript thêm yếu tố tương tác trong ngôi nhà, như là việc mở cánh cửa và làm đèn sáng. Bạn vẫn có thể làm web mà không có JavaScript nhưng rủi ro là website của bạn trông như là những web thời những năm 1995.

Làm thế nào để thêm JavaScript trên website của bạn?

Để thêm chuỗi code JavaScript code vào trang web, bạn sẽ cần gắn tag <script>. Ví dụ như sau:

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

Your JavaScript code

```
</script>
```

Với quy tắc cơ bản, bạn nên gắn JavaScript trong tag <header> cho website của bạn trừ khi bạn muốn nó thực thi tại một thời điểm nhất định hoặc một yếu tố nhất định của trang web. Bạn cũng có thể lưu code JavaScript dưới file riêng và gọi nó lên mỗi khi cần trên website.

Vậy, JavaScript là gì?

JavaScript là ngôn ngữ lập trình mang đến sự sinh động của website. Nó khác với HTML (thường chuyên cho nội dung) và CSS (thường chuyên dùng cho phong cách), và khác hẳn với PHP (chạy trên server chứ không chạy dưới máy client).

Bạn cần biết gì:

1. JavaScript là ngôn ngữ dễ học;
2. Nó được phát triển bởi Netscape, và đang được dùng trên 92% webstie;
3. JS có thể được gắn vào một element của trang web hoặc sự kiện của trang web như cú click chuột;
4. Hoạt động trên đa trình duyệt và đa thiết bị;
5. Nhanh và nhẹ hơn các ngôn ngữ lập trình khác;
6. Có thể ít an toàn hơn vì độ phổ biến của nó;
7. Bạn có thể thêm JavaScript trực tiếp vào HTML hoặc bạn có thể lưu nó trên files riêng biệt và gọi lên khi cần.

4. Thời điểm thực hiện một đoạn Script

4.1.Cấu trúc của đoạn Javascript

```
<Script language="JavaScript">
```

Các lệnh Javascript

```
</Script>
```

4.2.Javascript trong một trang HTML

- Đặt các dòng mã lệnh của Javascript giữa cặp thẻ <script></script>

- Có thể viết nhiều đoạn mã lệnh Javascript trong cùng một tập tin HTML. Các khối mã lệnh Javascript có thể đặt bất kỳ nơi nào của trang HTML. Có thể đặt trong cặp

thẻ <head></head> hoặc trong cặp <body></body> tuy nhiên ta nên đặt trong cặp thẻ <head> để dễ kiểm soát mã lệnh và cũng dễ sửa đổi chương trình.

- Có thể viết một tệp tin Javascript riêng và sau đó kết nối với một hoặc nhiều tệp tin trang web khác nhau.

Cách 1: Viết đoạn mã Script trong cùng trang HTML

Ví dụ 1:

```
<HTML>
<HEAD>
<script language="javascript" >
  document.write("What is your name? ");
</script>
</HEAD>
<BODY>
  Nội dung của trang
</BODY>
</HTML>
```

Ví dụ 2:

```
<HTML>
<BODY>
<script language="javascript">
  document.write("Hello World!")
</script>
</BODY>
</HTML>
```

Ví dụ 3:

```
<html>
<head>
<script type="text/javascript">
    some statements
</script>
</head>
<body>
<script type="text/javascript">
    some statements
</script>
</body>
</html>
```

Cách 2: Mở trình soạn thảo Notepad, viết đoạn chương trình Javascript. Lưu lại với phần mở rộng là .js (lưu ý trong tập tin này không chứa bất kỳ một thẻ nào của ngôn ngữ HTML).

- Liên kết với một file JavaScript.js đã được xây dựng trước

Cú pháp:

```
<HTML>
<BODY>
<Script SRC="fileJavaScript.js" Language="javascript" >
    JavaScript comments
</Script>
</BODY>
</HTML>
```

4.3. Môi trường viết Javascript

Có thể dùng chương trình soạn thảo: Frontpage, Notepate, Visual InterDev, Dreamweaver, chọn chế độ code, Dreamweaver hỗ trợ phân biệt từ khóa bằng màu chữ, hỗ trợ các hàm, thuộc tính của các tag, giúp người sử dụng thuận tiện trong việc thiết kế và viết chương trình.

5. Các biến

Biến trong JavaScript là một định danh dùng để lưu trữ dữ liệu, trạng thái (số, chuỗi, đối tượng, ...), thông qua định danh này chương trình sẽ làm việc với vùng nhớ bằng cách gọi định danh này.

Mỗi biến có một kiểu dữ liệu riêng, dựa vào kiểu dữ liệu của biến có các thao tác khác nhau với biến. Ví dụ với biến kiểu số nguyên, số thực thì có phép cộng, trừ, nhân, chia. Biến kiểu chuỗi thì có độ dài của chuỗi, phép nối 2 chuỗi, ...

5.1. Khai báo biến

Cú pháp	Ví dụ
var <tên biến>;	var n; n = 1;
var <tên biến> = <giá trị>;	var s = "JavaScript string";
<tên biến> = <giá trị>;	PI = 3.14;

5.2. Quy tắc đặt tên biến

- Dùng các ký tự a..z, A..Z, 1..9, dấu gạch dưới '_', dấu '\$'
- Tên biến không trùng với từ khóa JS
- Tên biến không bắt đầu bởi con số
- Tên biến không có ký tự khoảng trắng
- Tên biến là case sensitive.

Lưu ý: trong JavaScript có phân biệt chữ hoa và chữ thường. Ví dụ STDIO và Stdio được hiểu là hai tên khác nhau.

Ví dụ:

```
name_of_cat    // Hợp lệ
_Address      // Hợp lệ
3school        // Không hợp lệ vì bắt đầu bằng số
position%Enemy // Không hợp lệ vì có chứa ký tự đặc biệt %
```

Một số ví dụ về khai báo biến trong JavaScript:

```
let a;
a = 5;
let b = 6;
let c = a + b;

console.log(a);
console.log(c);

let name = "Lập trình JavaScript";

console.log(name);
```

Dòng 1: Khai báo một biến có tên là a.

Dòng 2: Lưu giá trị 5 vào biến a.

Dòng 3: Khai báo một biến có tên là b và lưu giá trị 6.

Dòng 4: Khai báo một biến c và lưu giá trị bằng tổng của biến a vào biến b.

Dòng 9: Khai báo biến name và lưu trữ chuỗi Lập trình JavaScript vào name.

Nếu không gán giá trị cho biến thì biến đó sẽ có kiểu là undefined.

Một số ví dụ khác:

```
var d = 7 + 1 + 4;  
var e = 8.23;  
var f;  
f = d + 3;  
var name = "Nguyễn Nghĩa";  
var address = "Hồ Chí Minh";  
var info = name + address;
```

5.3. Một số phong cách đặt tên biến

Phong cách đặt tên biến cũng rất quan trọng, để dễ dàng hiểu ý nghĩa của biến:

- Tên biến phải là một danh từ.
- Tên biến phải mang ý nghĩa rõ ràng.

Underscore

Mô tả: Mỗi từ cách nhau bởi mỗi dấu gạch dưới.

Ví dụ: name_of_cat, title_website, windows_height, ...

Camel case

Mô tả: Những chữ cái đầu của mỗi từ đều viết hoa, những chữ cái còn lại đều viết thường.

Ví dụ: NameOfCat, TitleWebsite, WindowsHeight, ...

Một phong cách được biến tấu từ Camel Case là những chữ cái đầu tiên viết thường và những chữ sau đó tuân theo Camel Case

Ví dụ: nameOfCat, titleWebsite, windowsHeight, ...

6. Các kiểu dữ liệu

- Trong JavaScript, các kiểu dữ liệu được chia thành những loại cơ bản như sau:

- o string
- o number
- o boolean
- o object
- o undefined
- o array (*đây là một trường hợp đặc biệt của kiểu dữ liệu object*)

6.1. Dữ liệu kiểu string

- Trong JavaScript, các dữ liệu thuộc kiểu string (hay còn được gọi là "chuỗi") là một tập hợp gồm các ký tự, chúng được viết bên trong cặp dấu nháy kép hoặc cặp dấu nháy đơn.

Ví dụ:

- o Giá trị của biến a là một chuỗi được viết bên trong cặp dấu nháy kép.

- o Giá trị của biến b là một chuỗi được viết bên trong cặp dấu nháy đơn.

```
<script>
  var a = "Tài liệu học HTML";
  var b = 'Lập Trình Web';
</script>
```

6.2. Dữ liệu kiểu number

- Trong JavaScript, các dữ liệu thuộc kiểu number (hay còn được gọi là "số") là một tập hợp của các con số (0 - 9) không chứa dấu khoảng trắng và có thể chứa dấu trừ (-) nằm ở đầu để đại diện cho số âm.

Ví dụ:

- o Giá trị của biến a là số 22.
- o Giá trị của biến b là số -1993.

```
<script>
  var a = 22;
  var b = -1993;
</script>
```

- Nếu một số được đặt bên trong cặp dấu nháy kép hoặc cặp dấu nháy đơn thì nó sẽ bị chuyển sang kiểu dữ liệu string.

Ví dụ:

- o Giá trị của biến a là chuỗi 22.
- o Giá trị của biến b là chuỗi -1993.

```
<script>
  var a = "22";
  var b = '-1993';
</script>
```

6.3. Dữ liệu kiểu boolean

- Trong JavaScript, các dữ liệu thuộc kiểu boolean chỉ có thể nhận một trong hai giá trị, đó là:

- § true (đúng)
- § false (sai)

- Có hai cách để nhận giá trị kiểu boolean, đó là:

- § Gán giá trị trực tiếp.
- § Nhận được từ một điều kiện.

Ví dụ:

- o Giá trị của biến a là true.
- o Giá trị của biến b là false.
- o Giá trị của biến c là true, vì điều kiện $(6 > 2)$ là đúng.
- o Giá trị của biến d là false, vì điều kiện $(6 > 10)$ là sai.

```
<script>
  var a = true;
  var b = false;
  var c = 6 > 2;
  var d = 6 > 10;
</script>
```

6.4. Dữ liệu kiểu object

- Trong JavaScript, các dữ liệu thuộc kiểu object (hay còn được gọi là "đối tượng") là một tập hợp gồm những cái tên và mỗi cái tên sẽ chứa đựng một giá trị dữ liệu.

- Lưu ý: Những cái tên còn được gọi là "thuộc tính" của đối tượng, giá trị của những cái tên còn được gọi là "giá trị thuộc tính của đối tượng".

Ví dụ:

Đoạn mã bên dưới dùng để tạo một đối tượng có tên là SinhVien, nó có ba thuộc tính:

- o Thuộc tính name với giá trị là chuỗi Nhân.
- o Thuộc tính gender với giá trị là chuỗi Nam.
- o Thuộc tính year với giá trị là số 1993.

```
<script>
  var SinhVien = {
    name:"Nhân",
    gender:"Nam",
    year:1993
  }
</script>
```

6.5. Dữ liệu kiểu undefined

- Trong JavaScript, khi một biến được khai báo mà không gán giá trị thì biến đó sẽ có giá trị là undefined và kiểu dữ liệu cũng là undefined.

Ví dụ: