

# Tổng quan về JavaScript

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.

## 1. 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.

### 1.1. 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.

## 1.2. 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.

## 2. Cú pháp Javascript

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;
```

### 2.1. 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)
- Chú thích (Comment)

### 2.2. 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).

### 2.3. Literal

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

### 2.4. Chuỗi

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

### 2.5. 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
```

### 2.6. 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.

$(5 + 6) * 10$

## 2.7. 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:  $x * 10$

## 2.8. 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.

## 2.9. 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.

`*/`

## 2.10. 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`.

## 3. Các cách nhúng JavaScript vào trang web

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

### 3.1. 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.

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

```
<!DOCTYPE html>

<html>

<body>

    <h1>A Web Page</h1>

    <p id="demo">A Paragraph</p>

    <button type="button" onclick="myFunction()">Try it</button>
```

```
<script>

function myFunction() {

    document.getElementById("demo").innerHTML = "Paragraph changed.";

}

</script>

</body>

</html>
```

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.

### 3.2. 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 riêng 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.

### 3.3. 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).

```
<button onclick="alert('Xin chào bạn!');">Click me</button>
```

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