



Trường Trung Cấp Hồng Hà Cần Thơ

NHẬP MÔN LẬP TRÌNH WEB

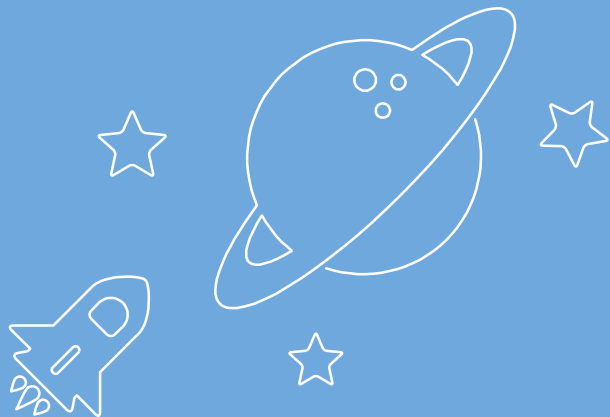
Ths. Nguyễn Minh Triết



BÀI 4. NGÔN NGỮ JAVASCRIPT

Ths. Nguyễn Minh Triết

1



GIỚI THIỆU VỀ JAVASCRIPT

GIỚI THIỆU JAVASCRIPT

- ❑ HTML nguyên thủy không có khả năng xử lý các tương tác của người dùng
- ❑ Mọi tương tác với trang web cần phải xử lý ở server -> chi phí về thời gian, thông lượng quá mức cần thiết
- ❑ Các trình duyệt có khả năng hỗ trợ thông dịch ngôn ngữ kịch bản

GIỚI THIỆU JAVASCRIPT

- ❑ JavaScript (*viết tắt là JS*) là một ngôn ngữ lập trình kịch bản phía máy khách (client-side script), mã lệnh được thực thi bởi trình duyệt của người dùng.
- ❑ Chia sẻ xử lý trong ứng dụng web. Giảm các xử lý không cần thiết trên server.
- ❑ JavaScript được sử dụng rộng rãi trong việc kết hợp với HTML/CSS để thiết kế web động.

GIỚI THIỆU JAVASCRIPT

❑ Client-Side Script:

Script được thực thi tại Client-Side (trình duyệt): Thực hiện các tương tác với người dùng (tạo menu chuyển động,...), kiểm tra dữ liệu nhập, ...

❑ Server-Side Script:

Script được xử lý tại Server-Side, nhằm tạo các trang web có khả năng phát sinh nội dung động. Một số xử lý chính: kết nối CSDL, truy cập hệ thống file trên server, phát sinh nội dung html trả về người dùng...

GIỚI THIỆU JAVASCRIPT

❑ Khả năng thường được dùng của JavaScript

- Nhận và kiểm tra tính hợp lệ của dữ liệu
- Tính toán dữ liệu tạm thời
- Tạo các hiệu ứng hoạt ảnh, xử lý các sự kiện

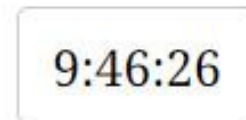
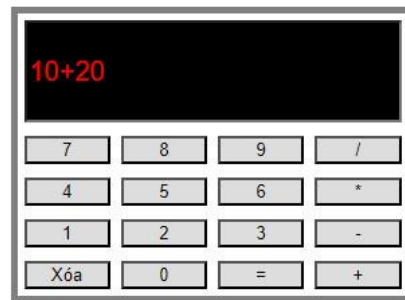
❑ Bị quy định không được phép truy cập vào các tài nguyên mức hệ thống

CẬP NHẬT THÔNG TIN CÁ NHÂN

Họ tên:

Ngày sinh:

Giới tính: ☒ Nam ☐ Nữ



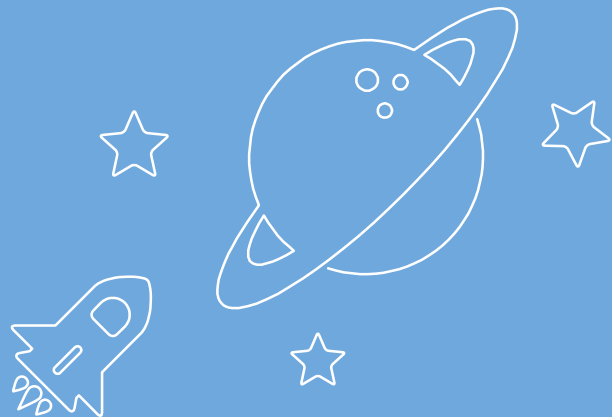
GIỚI THIỆU JAVASCRIPT

- ❑ JavaScript là ngôn ngữ dưới dạng script có thể gắn với các file HTML và dùng thông dịch (interpreter)
- ❑ JavaScript là ngôn ngữ dựa trên đối tượng: **math**, **document**, **windows**,...
- ❑ JavaScript không phải là ngôn ngữ hướng đối tượng như C++, Java,...
- ❑ Thiết kế độc lập với hệ điều hành

GIỚI THIỆU JAVASCRIPT

- ❑ Khi trình duyệt (**Client browser**) truy cập trang web có chứa các đoạn mã xử lý tại server-side. Server (**run-time engine**) sẽ thực hiện các lệnh Server-side Scripts và trả về nội dung HTML cho trình duyệt.
- ❑ Nội dung html trả về chủ yếu bao gồm: mã html, client-script.

2



NHÚNG JAVASCRIPT VÀO TRANG WEB

NHÚNG JAVASCRIPT VÀO TRANG WEB

🟡 Định nghĩa script trực tiếp trong trang HTML:

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

```
<body>  
  <h2>Tài liệu học HTML</h2>  
  <h2 id="demo1"></h2>  
  <h2>Tài liệu học JavaScript</h2>  
  <h2 id="demo2"></h2>  
  <h2 id="demo3"></h2>  
  <script>  
    document.getElementById("demo1").innerHTML = "Tài liệu học CSS";  
    document.getElementById("demo2").innerHTML = "Tài liệu học MySQL";  
    document.getElementById("demo3").innerHTML = "Tài liệu học PHP";  
  </script>  
</body>
```



Tài liệu học HTML

Tài liệu học CSS

Tài liệu học JavaScript

Tài liệu học MySQL

Tài liệu học PHP

ĐỊNH NGHĨA SCRIPT TRỰC TIẾP TRONG TRANG WEB

Lưu ý:

- Các đoạn `<script>` có thể được đặt ở bất kỳ nơi nào trong trang web. Tuy nhiên nên đặt chúng ở vị trí cuối cùng trong phần tử `<body>` giống ví dụ phía trên. Điều đó giúp cải thiện tốc độ tải giao diện của trang web và tránh một số trường hợp xảy ra lỗi thực thi ngoài mong đợi.
- Trong một trang web có thể sử dụng nhiều đoạn `<script>` và trong một đoạn `<script>` thì không giới hạn số lượng câu lệnh.

ĐỊNH NGHĨA SCRIPT TRỰC TIẾP TRONG TRANG WEB

```
<body>
  <h2>Tài liệu học HTML</h2>
  <h2 id="demo1"></h2>
  <h2>Tài liệu học JavaScript</h2>
  <h2 id="demo2"></h2>
  <h2 id="demo3"></h2>
  <script>
    document.getElementById("demo1").innerHTML = "Tài liệu học CSS";
    document.getElementById("demo2").innerHTML = "Tài liệu học MySQL";
    document.getElementById("demo3").innerHTML = "Tài liệu học PHP";
  </script>
  <h2 id="demo4"></h2>
  <h2 id="demo5"></h2>
  <script>
    document.getElementById("demo4").innerHTML = "Tài liệu học jQuery";
    document.getElementById("demo5").innerHTML = "Tài liệu học Bootstrap";
  </script>
</body>
```



Tài liệu học HTML

Tài liệu học CSS

Tài liệu học JavaScript

Tài liệu học MySQL

Tài liệu học PHP

Tài liệu học jQuery

Tài liệu học Bootstrap



- Bên trong đoạn `<script>` chỉ dùng để chứa các mã lệnh JavaScript và tuyệt đối không được đặt vào đó những đoạn `<script>` khác.

NHÚNG JAVASCRIPT VÀO TRANG WEB

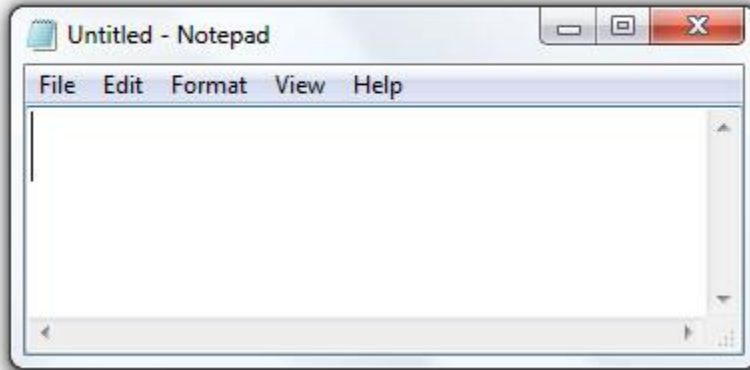
Viết mã lệnh vào tập tin Javascript

- Bước 1: Tạo tập tin Javascript
- Bước 2: Viết mã lệnh trong tập tin Javascript
- Bước 3: Nhúng tập tin Javascript vào trang web

VIẾT MÃ LỆNH VÀO TẬP TIN JAVASCRIPT

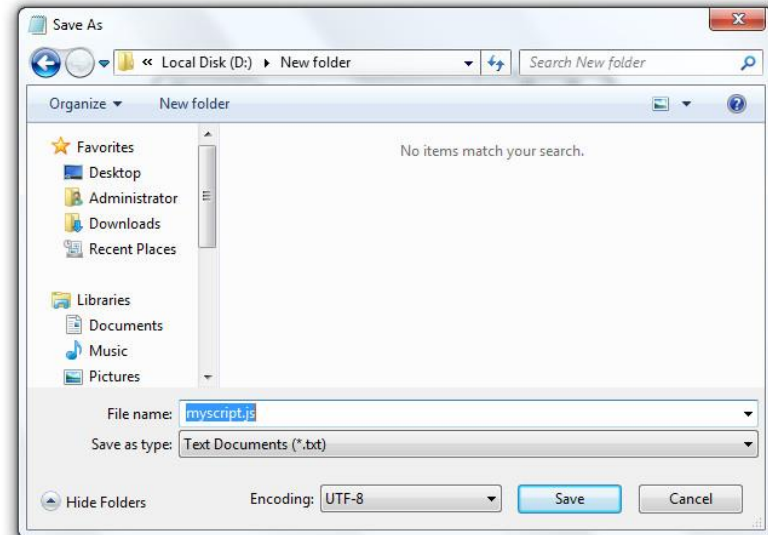
1. Tạo tập tin Javascript

- Bước 1: Mở Notepad lên.



- Bước 2: Lưu tập tin.

- Bấm vào "File" rồi chọn "Save As"
- Chọn thư mục để lưu tập tin
- Tên tập tin phải có phần mở rộng là `.js`
(Ở hướng dẫn này, tôi tạo tập tin javascript có tên là `myscript`)
- Chỗ Encoding chọn "UTF-8"
- Cuối cùng, bấm nút "Save" để lưu tập tin




VIẾT MÃ LỆNH VÀO TẬP TIN JAVASCRIPT

2. Viết mã lệnh trong tập tin Javascript

- Cách viết mã lệnh trong tập tin JavaScript cũng giống như cách viết mã lệnh trực tiếp vào trang web. Điểm khác nhau là khi viết mã lệnh trong tập tin JavaScript thì ta **không cần đặt các câu lệnh bên trong cặp thẻ `<script> </script>`**

```
document.getElementById("demo1").innerHTML = "Tài liệu học CSS";  
document.getElementById("demo2").innerHTML = "Tài liệu học MySQL";  
document.getElementById("demo3").innerHTML = "Tài liệu học PHP";
```



```
<script>  
document.getElementById("demo1").innerHTML = "Tài liệu học CSS";  
document.getElementById("demo2").innerHTML = "Tài liệu học MySQL";  
document.getElementById("demo3").innerHTML = "Tài liệu học PHP";  
</script>
```


VIẾT MÃ LỆNH VÀO TẬP TIN JAVASCRIPT

3. Nhúng tập tin Javascript vào trang web

`<script src="đường dẫn đến tập tin JavaScript"></script>`

```
<body>
  <h2>Tài liệu học HTML</h2>
  <h2 id="demo1"></h2>
  <h2>Tài liệu học JavaScript</h2>
  <h2 id="demo2"></h2>
  <h2 id="demo3"></h2>
  <script src="../file/myscript.js"></script>
</body>
```

```
<body>
  <h2>Tài liệu học HTML</h2>
  <h2 id="demo1"></h2>
  <h2>Tài liệu học JavaScript</h2>
  <h2 id="demo2"></h2>
  <h2 id="demo3"></h2>
  <script src="../file/myscript.js"></script>
  <h2 id="demo4"></h2>
  <h2 id="demo5"></h2>
  <script src="../file/myscript_2.js"></script>
```

VIẾT MÃ LỆNH VÀO TẬP TIN JAVASCRIPT

🟡 Lưu ý:

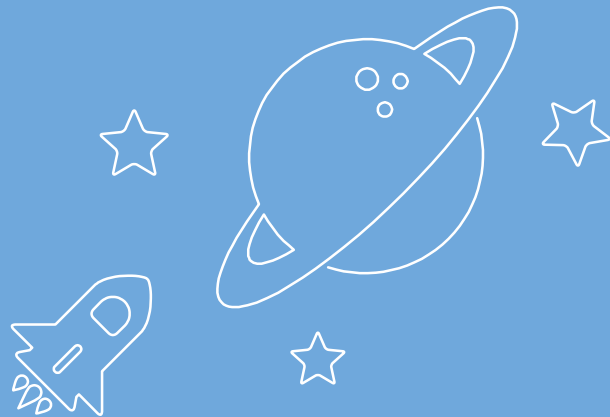
- Các đoạn `<script>` có thể được đặt ở bất kỳ nơi nào trong trang web. Tuy nhiên nên đặt chúng ở vị trí cuối cùng trong phần tử `<body>` giống ví dụ phía trên. Điều đó giúp cải thiện tốc độ tải giao diện của trang web và tránh một số trường hợp xảy ra lỗi thực thi ngoài mong đợi.
- Trong một trang web có thể sử dụng nhiều đoạn `<script>` và trong một đoạn `<script>` thì không giới hạn số lượng câu lệnh.

NHỮNG JAVASCRIPT VÀO TRANG WEB

● Kết luận:

- Giữa việc viết mã lệnh trong tập tin JavaScript và viết trực tiếp vào trang web thì chúng ta nên dùng cách viết mã lệnh trong tập tin JavaScript.
- Bởi vì đối với những chương trình lớn khoảng vài trăm hoặc vài nghìn dòng, việc viết vào tập tin JavaScript sẽ giúp ta dễ quan sát, quản lý cũng như cập nhật nâng cấp mã nguồn hơn.

3



CÁCH HIỂN THỊ DỮ LIỆU TRONG JAVASCRIPT

HIỂN THỊ DỮ LIỆU TRONG JAVASCRIPT

- Trong HTML, việc hiển thị một nội dung gì đó lên màn hình thật đơn giản, ta chỉ cần gõ trực tiếp nội dung muốn hiển thị là xong
- Còn đối với JavaScript thì không được hỗ trợ hiển thị dữ liệu trực tiếp giống như trong HTML, mà phải thông qua một phương thức

Tài liệu học Lập Trình Web

```
<script>  
    document.write("Tài liệu học Lập Trình Web");  
</script>
```

HTML

JAVASCRIPT

HIỂN THỊ DỮ LIỆU TRONG JAVASCRIPT

● Ba cách cơ bản dùng để hiển thị nội dung lên màn hình trong JavaScript:

- Cách 1: Sử dụng **alert()**
- Cách 2: Sử dụng **document.write()**
- Cách 3: Sử dụng **document.getElementById().innerHTML**

alert()

- Khi sử dụng **alert()**, nội dung mà bạn muốn hiển thị sẽ được hiển thị bên trong một hộp thoại thông báo.
- Cú pháp:

```
alert("nội dung muốn hiển thị");
```

```
<html>
<body>

  <script>
    alert("Tài liệu học Lập Trình Web");
  </script>

</body>
</html>
```

webcoban.vn cho biết

Tài liệu học Lập Trình Web

OK

document.write()

- Khi sử dụng **document.write()**, nội dung mà bạn muốn hiển thị sẽ được hiển thị ngay đúng vị trí mà nó được đặt trong trang web.
- Cú pháp:

```
document.write("nội dung muốn hiển thị");
```

```
<body>

<h1>Tài liệu học HTML</h1>

<script>document.write("<h1>Tài liệu học CSS</h1>");</script>

<h1>Tài liệu học JavaScript</h1>

<script>document.write("<h1>Tài liệu học MySQL</h1>");</script>

<script>document.write("<h1>Tài liệu học PHP</h1>");</script>

</body>
```

Tài liệu học HTML

Tài liệu học CSS

Tài liệu học JavaScript

Tài liệu học MySQL

Tài liệu học PHP

document.getElementById().innerHTML

- Khi sử dụng **document.getElementById().innerHTML**, nội dung mà bạn muốn hiển thị sẽ được hiển thị bên trong một phần tử xác định.
- Cú pháp:

```
document.getElementById("id của phần tử").innerHTML = "nội dung muốn hiển thị";
```

```
<body>
  <h1>Tài liệu học HTML</h1>
  <h1 id="skud"></h1>
  <h1>Tài liệu học JavaScript</h1>
  <h1>Tài liệu học MySQL</h1>
  <h1 id="raw"></h1>
  <script>
    document.getElementById("skud").innerHTML = "Tài liệu học CSS";
    document.getElementById("raw").innerHTML = "Tài liệu học PHP";
  </script>
</body>
```

Tài liệu học HTML

Tài liệu học CSS

Tài liệu học JavaScript

Tài liệu học MySQL

Tài liệu học PHP

document.getElementById().innerHTML

- *Lưu ý:* Sau khi sử dụng **document.getElementById().innerHTML**, nội dung mà bạn muốn hiển thị sẽ thay thế tất cả nội dung ban đầu của phần tử.

```
<h1>Tài liệu học HTML</h1>
<h1 id="skud">Tài liệu học JavaScript</h1>
<h1>Tài liệu học CSS</h1>
<script>
    document.getElementById("skud").innerHTML = "Lập Trình Web";
</script>
```

Tài liệu học HTML

Lập Trình Web

Tài liệu học CSS

=> Tất cả nội dung bên trong phần tử có **id** là **skud** sẽ bị thay thế bởi câu *"Lập Trình Web"*

HIỂN THỊ DỮ LIỆU TRONG JAVASCRIPT

- Nếu nội dung mà bạn muốn hiển thị là một chuỗi ký tự, thì chuỗi ký tự đó phải được đặt bên trong cặp dấu nháy kép hoặc cặp dấu nháy đơn.
- Nếu nội dung mà bạn muốn hiển thị là một số, thì số đó có thể đặt *(hoặc không đặt)* bên trong cặp dấu nháy kép hoặc cặp dấu nháy đơn.

```
<script>
  document.getElementById("demo1").innerHTML = "Lập Trình Web"; //ĐÚNG
  document.getElementById("demo2").innerHTML = 'Lập Trình Web'; //ĐÚNG
  document.getElementById("demo3").innerHTML = "1993"; //ĐÚNG
  document.getElementById("demo4").innerHTML = '1993'; //ĐÚNG
  document.getElementById("demo5").innerHTML = 1993; //ĐÚNG
  document.getElementById("demo6").innerHTML = "Lập Trình Web"; //SAI
  document.getElementById("demo7").innerHTML = 'Lập Trình Web'; //SAI
  document.getElementById("demo8").innerHTML = Lập Trình Web; //SAI
</script>
```

HIỂN THỊ DỮ LIỆU TRONG JAVASCRIPT

- Nếu chuỗi ký tự được đặt bên trong cặp dấu nháy kép thì chuỗi đó không được chứa ký tự là dấu nháy kép (*tuy nhiên nó được phép chứa ký tự là dấu nháy đơn*)
- Nếu chuỗi ký tự được đặt bên trong cặp dấu nháy đơn thì chuỗi đó không được chứa ký tự là dấu nháy đơn (*tuy nhiên nó được phép chứa ký tự là dấu nháy kép*)

```
<script>
  document.getElementById("demo1").innerHTML = "Lập "Trình Web"; //SAI
  document.getElementById("demo2").innerHTML = 'Lập "Trình Web'; //SAI
  document.getElementById("demo3").innerHTML = "Lập "Trình Web"; //ĐÚNG
  document.getElementById("demo4").innerHTML = 'Lập "Trình Web'; //ĐÚNG
</script>
```

HIỂN THỊ DỮ LIỆU TRONG JAVASCRIPT

- Có thể dùng chuỗi để tạo các phần tử HTML

```
<script>  
    document.write("<h1>Tài liệu học HTML</h1>");  
    document.write("<p><i><u>Tài liệu học HTML</u></i></p>");  
</script>
```

Tài liệu học HTML

Tài liệu học HTML

4



CÂU LỆNH VÀ CHƯƠNG TRÌNH TRONG JAVASCRIPT

CÂU LỆNH VÀ CHƯƠNG TRÌNH TRONG JAVASCRIPT

1. Câu lệnh Javascript

- Câu lệnh JavaScript có thể tạm hiểu là một công việc sẽ được thực thi bởi trình duyệt.
- Mỗi câu lệnh cần phải được kết thúc bởi dấu chấm phẩy ;

```
<script>  
    document.write("Lập Trình Web");  
</script>
```

=> Câu lệnh với công việc là hiển thị dòng chữ Lập Trình Web lên màn hình

CÂU LỆNH VÀ CHƯƠNG TRÌNH TRONG JAVASCRIPT

2. Chương trình Javascript

- Chương trình JavaScript là một **tập hợp gồm nhiều câu lệnh JavaScript**.
- Trong một chương trình, các câu lệnh sẽ được thực thi lần lượt theo thứ tự **từ trên xuống dưới**.

(Câu lệnh phía trên phải thực thi xong thì câu lệnh bên dưới mới được thực thi)

```
<script>
  var a = 10;
  var b = 25;
  var c = a + b;
  document.write(c);
</script>
```


CÂU LỆNH VÀ CHƯƠNG TRÌNH TRONG JAVASCRIPT

- Nếu câu lệnh phía trên bị lỗi thì tất cả những câu lệnh còn lại ở bên dưới sẽ không được thực thi.

```
<script>
  var a = 10;
  var b = 25;
  var c = a + b;
  document.write(c);
  skjdfbshvg; //Câu lệnh này bị lỗi, do đó hai câu lệnh bên dưới không được thực thi
  var x = 100 + 200;
  document.write(x);
</script>
```

- **Lưu ý:** Việc xáo trộn thứ tự các câu lệnh sẽ khiến kết quả thực thi không như mong đợi hoặc thậm chí bị lỗi.

CÂU LỆNH VÀ CHƯƠNG TRÌNH TRONG JAVASCRIPT

3. Một vài quy tắc trong việc viết mã lệnh

3.1. Xuống dòng sau mỗi câu lệnh

```
<script>  
  var a = 10; var b = 25; var c = a + b; document.write(c);  
</script>
```

```
<script>  
  var a = 10;  
  var b = 25;  
  var c = a + b;  
  document.write(c);  
</script>
```

CÂU LỆNH VÀ CHƯƠNG TRÌNH TRONG JAVASCRIPT

3.2. Thụt đầu dòng

- Trong JavaScript, việc thụt đầu dòng sẽ không gây ảnh hưởng gì đến chương trình.
- Do đó, ta có thể thụt đầu dòng cho các câu lệnh để dễ xem mã lệnh hơn.

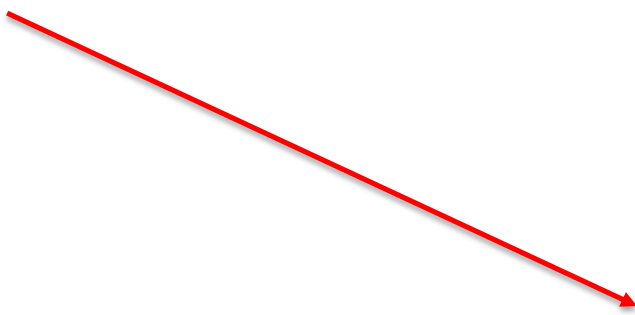
```
<script>  
var a = 10;  
var b = 25;  
var c = a + b;  
document.write(c);  
</script>
```

```
<script>  
    var a = 10;  
    var b = 25;  
    var c = a + b;  
    document.write(c);  
</script>
```

CÂU LỆNH VÀ CHƯƠNG TRÌNH TRONG JAVASCRIPT

3.3. Khoảng trắng giữa các toán tử

- Các khoảng trắng nằm phía trước hoặc sau các toán tử (*như* +, -, *, /, =,) là không quan trọng.



```
<script>
```

```
var a = 5 + 10;
```

```
var a    =    5+    10;
```

```
var a=5+10;
```

```
</script>
```

5



NGUYÊN LÝ THỰC THI MÃ LỆNH TRONG JAVASCRIPT

NGUYÊN LÝ THỰC THI MÃ LỆNH

1. Thứ tự thực thi của các câu lệnh trong kịch bản

- Trong một kịch bản, các câu lệnh Javascript được thực thi lần lượt theo thứ tự từ trên xuống dưới

```
<script>  
    document.write("<h2>Tài liệu học HTML</h2>");  
    document.write("<h2>Tài liệu học CSS</h2>");  
    document.write("<h2>Tài liệu học JavaScript</h2>");  
</script>
```

NGUYÊN LÝ THỰC THI MÃ LỆNH

- Có những kịch bản mà trong đó câu lệnh bên dưới phải cần đến dữ liệu từ câu lệnh ở trên. Vì vậy, việc xác định thứ tự của các câu lệnh là vô cùng quan trọng (*Nếu xác định sai có thể dẫn đến trường hợp kịch bản thực thi không như mong đợi hoặc xảy ra lỗi*)

```
<script>  
  var a = 100;  
  var b = 70;  
  var c = a + b;  
  document.write(c);  
</script>
```

Phải tồn tại a và b thì c mới có thể tính tổng của a và b được

NGUYÊN LÝ THỰC THI MÃ LỆNH

2. Thứ tự thực thi của các kịch bản trong trang web

- Trong một trang web, các kịch bản được thực thi lần lượt theo thứ tự từ trên xuống dưới

```
<script>
  document.write("<p>Tài liệu học CSS</p>");
  document.write("<p>Tài liệu học JavaScript</p>");
</script>
<p>Tài liệu học MySQL</p>
<p>Tài liệu học PHP</p>
<script>
  var a = 100;
  var b = 70;
  var c = a + b;
  document.write(c);
</script>
```

Thứ tự thực thi của các tập tin JavaScript được nhúng vào trang web cũng giống các đoạn *<script>*

NGUYÊN LÝ THỰC THI MÃ LỆNH

3. Mối liên hệ giữa HTML và mã lệnh Javascript

- Không riêng gì thứ tự thực thi giữa các câu lệnh JavaScript mới quan trọng, thứ tự thực thi giữa HTML và JavaScript cũng quan trọng không kém.

```
<h2 id="demo">Tài liệu học HTML</h2>  
<script>  
    document.getElementById("demo").innerHTML = "Tài liệu học CSS";  
</script>
```

=> Thẻ <h2> thực thi trước để **tạo phần tử có id là demo**, sau đó câu lệnh script được thực thi với ý nghĩa **thay thế nội dung của phần tử có id demo thành Tài liệu học CSS**

NGUYÊN LÝ THỰC THI MÃ LỆNH

- Ở ví dụ phía trên, phần tử có id là demo phải tồn tại trước thì câu lệnh JavaScript mới có thể thay thế nội dung của nó thành câu *Tài liệu học CSS* được. Còn nếu thay thế nội dung mà trong khi phần tử đó chưa tồn tại thì sẽ dẫn đến trường hợp bị lỗi.

```
<script>  
  document.getElementById("demo").innerHTML = "Tài liệu học CSS";  
</script>  
<h2 id="demo">Tài liệu học HTML</h2>
```

- Trong JavaScript, **khi một câu lệnh bị lỗi thì việc thực thi sẽ kết thúc** (tức là những câu lệnh JavaScript nằm sau câu lệnh bị lỗi sẽ không được thực thi tiếp)
- Không giống như những ngôn ngữ lập trình khác, **khi bị lỗi JavaScript không hiển thị thông báo lên màn hình, nó chỉ hiển thị lỗi trong phần Console** của trình duyệt

6



CÁCH GHI CHÚ THÍCH TRONG JAVASCRIPT

CHÚ THÍCH

1. Giới thiệu chú thích

- Chú thích là những đoạn mã mà **khi trình duyệt gặp phải chúng thì trình duyệt sẽ bỏ qua** (*có nghĩa là trình duyệt sẽ không thực thi những đoạn mã đó*)
- Chú thích thường được dùng để ghi những lời như: *nhận xét, giải thích, lưu ý, nhắc nhở,...* nhằm phục vụ cho việc nếu sau này có xem lại mã lệnh để nâng cấp thì cũng dễ dàng hơn.
- Ngoài ra, chúng ta cũng có thể **kiểm tra sự thực thi của một đoạn mã JavaScript nào đó** bằng cách sử dụng việc ghi chú thích để **vô hiệu hóa những đoạn mã JavaScript còn lại.**

CHÚ THÍCH

2. Cách viết chú thích

2.1. Viết chú thích trên một dòng

- Nếu phần nội dung chú thích nằm gọn trên một dòng thì chỉ cần đặt cặp dấu // phía trước nó.
- Ngoài ra, cặp dấu ***/* */*** cũng có thể dùng để chú thích cho phần nội dung nằm gọn trên một dòng.

```
<script>
  document.write("Tài liệu học HTML<br>");
  //document.write("Tài liệu học CSS<br>");
  document.write("Tài liệu học JavaScript<br>");
  document.write("Tài liệu học MySQL<br>");
  document.write("Tài liệu học PHP<br>");
</script>
```

```
<script>
  document.write("Tài liệu học HTML<br>");
  document.write("Tài liệu học CSS<br>");
  document.write("Tài liệu học JavaScript<br>");
  /*document.write("Tài liệu học MySQL<br>");*/
  document.write("Tài liệu học PHP<br>");
</script>
```

CHÚ THÍCH

2.2. Viết chú thích trên nhiều dòng

- Nếu phần nội dung chú thích trải dài trên nhiều dòng thì ta dùng cặp dấu `/* */` để bao quanh chúng.

```
<script>
    document.write("Tài liệu học HTML<br>");
    /*document.write("Tài liệu học CSS<br>");
    document.write("Tài liệu học JavaScript<br>");
    document.write("Tài liệu học MySQL<br>");*/
    document.write("Tài liệu học PHP<br>");
</script>
```

7



CÁCH SỬ DỤNG BIẾN (VARIABLE) TRONG JAVASCRIPT

CÁCH SỬ DỤNG BIẾN (VARIABLE)

1. Biến là gì ?

- Biến cũng giống như là **một cái thùng chứa** dùng để lưu trữ một giá trị dữ liệu
- Ví dụ, kịch bản bên dưới có một biến tên là **name** và nó lưu trữ giá trị **Nguyễn Thành Nhân**

```
<script>  
    var name = "Nguyễn Thành Nhân";  
</script>
```

- Biến giúp người lập trình đơn giản hóa việc sử dụng một giá trị nhiều lần bằng cách gọi tên biến.

CÁCH SỬ DỤNG BIẾN (VARIABLE)

```
<script>
  document.write("Họ tên là " + "Nguyễn Thành Nhân");
  document.write("Giới tính của " + "Nguyễn Thành Nhân" + " là nam");
  document.write("Năm sinh của " + "Nguyễn Thành Nhân" + " là 1993");
</script>
```



```
<script>
  var name = "Nguyễn Thành Nhân";
  document.write("Họ tên: " + name);
  document.write("Giới tính của " + name + " là nam");
  document.write("Năm sinh của " + name + " là 1993");
</script>
```

CÁCH SỬ DỤNG BIẾN (VARIABLE)

2. Cách khai báo một biến

2.1 Khai báo một biến

- Để khai báo một biến (*hay còn gọi là tạo một biến*), ta sử dụng cú pháp như sau:

```
var tên_biến;
```

- Ví dụ, khai báo một biến có tên là *hoten*

```
var hoten;
```

CÁCH SỬ DỤNG BIẾN (VARIABLE)

2. Cách khai báo một biến

2.1 Khai báo một biến

- Để khai báo một biến (*hay còn gọi là tạo một biến*), ta sử dụng cú pháp như sau:

```
var tên_biến;
```

- Ví dụ, khai báo một biến có tên là *hoten*

```
var hoten;
```

CÁCH SỬ DỤNG BIẾN (VARIABLE)

2.2 Gán giá trị cho biến

- Sau khi khai báo, nếu biến vẫn chưa được gán giá trị thì mặc định nó sẽ có giá trị là undefined
- Để gán giá trị cho một biến, ta sử dụng cú pháp như sau:

tên_biến= giá_trị;

- Ví dụ, gán giá trị cho biến hoten là Nguyễn Văn A

```
<script>
  var hoten;
  hoten = "Nguyễn Văn A";
  document.write("Giá trị của biến hoten là: " + hoten);
</script>
```

CÁCH SỬ DỤNG BIẾN (VARIABLE)

2.3 Cập nhật giá trị cho biến

- Sau khi một biến đã được gán giá trị, nếu muốn cập nhật cho nó một giá trị mới thì ta sử dụng cú pháp như sau:

tên_biến= giá_trị_mới;

- Cập nhật lại giá trị của biến hoten thành Trần Văn C

```
<script>
  var hoten;

  hoten = "Nguyễn Văn A";
  document.write("Giá trị của biến hoten là: " + hoten);

  hoten = "Trần Văn C";
  document.write("<br>Giá trị của biến hoten là: " + hoten);
</script>
```

Giá trị của biến hoten là: Nguyễn Văn A
Giá trị của biến hoten là: Trần Văn C

CÁCH SỬ DỤNG BIẾN (VARIABLE)

2.4 Khai báo biến và gán giá trị tạm thời

- Thay vì phải chia thành hai bước: (1) khai báo biến (2) gán giá trị thì ta có thể gom chung thành một bước khai báo biến rồi gán giá trị luôn.

```
var tên_biến=giá trị;
```

- Ví dụ, khai báo một biến *hoten* có giá trị là Nguyễn Văn A

```
<script>  
  var hoten = "Nguyễn Văn A";  
  document.write("Giá trị của biến hoten là: " + hoten);  
</script>
```

CÁCH SỬ DỤNG BIẾN (VARIABLE)

3. Khai báo nhiều biến đồng thời

- Có thể khai báo nhiều biến một lượt bằng cách dùng dấu phẩy ngăn giữa các tên biến, rồi sau đó gán giá trị cho từng biến.

```
<script>  
  var a, b, c;  
  a = "Nguyễn Thành Nhân";  
  b = 1993;  
  c = "Nam";  
</script>
```

- Ta cũng có thể vừa khai báo vừa gán giá trị cho biến (*Tuy nhiên, không bắt buộc phải gán giá trị cho tất cả các biến cùng một lượt*)

```
<script>  
  var a = "Nguyễn Thành Nhân", b, c = 1993;  
</script>
```

CÁCH SỬ DỤNG BIẾN (VARIABLE)

4. Giá trị của biến

- Giá trị của biến rất đa dạng, có thể là: *một chuỗi ký tự, một số, một biến khác, một biểu thức, undefined,*

4.1 Giá trị của biến là **một chuỗi ký tự**

- Nếu giá trị của biến là một chuỗi ký tự thì ta phải đặt chuỗi ký tự đó bên trong cặp dấu nháy kép `" "` hoặc cặp dấu nháy đơn `' '`

```
var hoten = "Nguyễn Thành Nhân";
```

```
var thanhpho = 'Cần Thơ';
```


CÁCH SỬ DỤNG BIẾN (VARIABLE)

4.2 Giá trị của biến là **một số**

- Nếu giá trị của biến là một số thì ta cứ để nguyên.

```
var namsinh = 1993;
```

- Nếu giá trị của biến là một số mà ta lại đặt bên trong cặp dấu nháy kép hoặc nháy đơn thì nó sẽ bị ép sang kiểu chuỗi ký tự

```
<script>
  var nam1 = 1993;
  var nam2 = "1993";

  document.write("Kiểu dữ liệu của biến nam1 là: " + typeof nam1);
  document.write("<hr>");
  document.write("Kiểu dữ liệu của biến nam2 là: " + typeof nam2);
</script>
```

Kiểu dữ liệu của biến nam1 là: number

Kiểu dữ liệu của biến nam2 là: string

CÁCH SỬ DỤNG BIẾN (VARIABLE)

4.3 Giá trị của biến là **một biến khác**

- Giá trị của biến có thể là một biến khác.
- Khi đó, giá trị thực của biến sẽ là giá trị của biến mà nó lưu trữ.

```
var hoten = "Nguyễn Thành Nhân";
```

```
var name = hoten; //biến name sẽ có giá trị là Nguyễn Thành Nhân
```

CÁCH SỬ DỤNG BIẾN (VARIABLE)

4.4 Giá trị của biến là **một biểu thức**

- Giá trị của biến có thể là một biểu thức.
- Khi đó, giá trị thực của biến sẽ là kết quả của biểu thức *(sau khi đã được tính toán xong)*.

```
var a = 100;
```

```
var b = 50;
```

```
var c = "Thành Nhân";
```

```
var result_1 = (35 + 10)*2; //biến result_1 có giá trị là 90
```

```
var result_2 = a + b - 30; //biến result_2 có giá trị là 120
```

```
var result_3 = c + " Cần Thơ"; //biến result_3 có giá trị là Thành Nhân Cần Thơ
```

CÁCH SỬ DỤNG BIẾN (VARIABLE)

4.5 Giá trị của biến là **undefined**

- Khi một biến được khai báo mà chưa được gán giá trị thì biến đó sẽ nhận giá trị là *undefined* (không xác định).

```
var a; //Khi này biến a sẽ có giá trị là undefined
```

CÁCH SỬ DỤNG BIẾN (VARIABLE)

5. Quy tắc đặt tên biến

- Tên biến là một tập hợp gồm một hoặc nhiều ký tự.
- Tên biến có thể chứa các ký tự trong danh sách bên dưới:

Các chữ cái in hoa	A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z
Các chữ cái thường	a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z
Các chữ số	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Dấu gạch dưới	_
Dấu đô la	\$ (khuyến khích KHÔNG NÊN sử dụng dấu đô la trong việc đặt tên biến)

- Tên biến tuyệt đối không được phép chứa các ký tự đặt biệt (Ví dụ như: @, #, !, %, ^, &,)
- Tên biến không được bắt đầu bằng một chữ số.

CÁCH SỬ DỤNG BIẾN (VARIABLE)

5. Quy tắc đặt tên biến

Một số ví dụ đặt tên biến đúng quy tắc	Một số ví dụ đặt tên biến sai quy tắc
w webcoban Webcoban webCobAn we9co3an _webc_oban	9webcoban (sai vì bắt đầu bằng chữ số) web%^coban (sai vì chứa ký tự đặc biệt)

- Tên biến có phân biệt trường hợp chữ in hoa và chữ thường (Ví dụ, *webcoban* và *Webcoban* là hai biến khác nhau)
- Tên biến phải duy nhất (không được khai báo một biến có tên trùng với tên của một biến đã được khai báo trước đó)
- Không được đặt tên biến trùng với các từ dành riêng trong JavaScript



Thank you !