Buổi 6
Tổng quan về JavaScript

I. Các khái niệm về JavaScript

JavaScript[1] là 1 ngôn ngữ kịch bản (*script*) cho phép lập trình viên cài đặt các hành động cho trang web phản hồi lại các thao tác từ người dùng.

Ví dụ:

- Khi người dùng nhấn vào 1 nút, một cửa sổ thông báo xuất hiện.
- Khi người dùng di chuyển chuột lên 1 tấm ảnh, tấm ảnh được phóng to.
- …

Có 3 cách để cài đặt JS cho 1 file HTML:

- External JS: Mã nguồn JS được lưu trong 1 file có đuôi *.js* và được chèn vào trang web thông qua thẻ `<script>` nằm trong thẻ `<head>` hoặc thẻ `<body>`:

```
<script src="js/script.js">
```

- Internal JS: Mã nguồn JS được đặt trong thẻ `<script>` nằm trong thẻ `<head>` hoặc thẻ `<body>`:

```
<script>
    // Mã nguồn JS
</script>
```

- Inline JS: Mã nguồn JS được đặt trong các sự kiện của thẻ HTML:

```
<button onclick="…">Đăng nhập</button>
```

JS không chỉ được dùng trong các trang web mà còn ở các ứng dụng khác, ví dụ như Node.js. Ngoài ra, một số loại CSDL như MongoDB hay CouchDB cũng sử dụng JS làm ngôn ngữ lập trình.

Lưu ý: JavaScript và Java là 2 ngôn ngữ khác nhau.

II. Các thành phần cơ bản của JavaScript

Tương tự ngôn ngữ C/C++, câu lệnh của JS phải kết thúc bằng dấu chấm phẩy (;)

1. Biến và kiểu dữ liệu

JS không ràng buộc kiểu dữ liệu khi khai báo biến, do đó, 1 biến có thể lưu nhiều kiểu dữ liệu khác nhau tùy vào giá trị mà nó được gán.

3 kiểu dữ liệu thường gặp nhất trong JS là Number, String và Boolean.

Cú pháp khai báo biến:

```
var <tên biến>;
```

Ví dụ:

```
var x;                  // Tại đây x là undefined
x = 5;                  // Tại đây x thuộc kiểu Number
x = "Hello";            // Tại đây x thuộc kiểu String
x = 16 + 4.0;           // Tại đây x có giá trị 20
x = 16 + 4 + "JS";      // Tại đây x có giá trị là "20JS"[2]
x = 16 + "JS" + 4;      // Tại đây x có giá trị là "16JS4"
```

[1] Viết tắt là JS

[2] Phép cộng (+) khi thực hiện trên String sẽ là phép nối chuỗi

JS hỗ trợ hàm `isNaN()` để kiểm tra 1 giá trị có phải là số hay không, và hàm `Number()` để chuyển 1 kiểu dữ liệu thành dạng số.

2. Toán tử

Tương tự ngôn ngữ C/C++, JS hỗ trợ 1 số toán tử như sau:

- Toán tử số học: `+ - * / % ++ -- **` (toán tử ** là phép lũy thừa).
- Toán tử gán: `= += -= *= /= %= **=`
- Toán tử so sánh: `== != > >= < <= === !==` (toán tử === và !== xét cả kiểu dữ liệu, còn toán tử == và != chỉ xét giá trị).
- Toán tử logic: `! && ||`
- Toán tử điều kiện: `?`

Phân biệt toán tử == và ===, != và !==: Giả sử `x = 5`:

Toán tử	Ý nghĩa	Biểu thức	Giá trị
==	bằng giá trị	x == 8 x == 5 x == "5"	false true true
===	bằng giá trị **và** cùng kiểu	x === 5 x === "5"	true false
!=	khác giá trị	x != 8	true
!==	khác giá trị **hoặc** khác kiểu	x !== 8 x !== 5 x !== "5"	true false true

Khi thực hiện phép + trên chuỗi và số, JS sẽ chuyển số về dạng chuỗi và thực hiện nối chuỗi.

Khi thực hiện phép so sánh trên chuỗi và số, JS sẽ chuyển chuỗi về dạng số trước khi tiến hành so sánh. Nếu chuỗi không thể chuyển về dạng số thì xem như giá trị là NaN[3] và kết quả phép so sánh luôn là `false`:

Biểu thức	Giá trị
2 < 12	true
2 < "12"	true
2 < "John"	false
2 > "John"	false
2 == "John"	false
"2" < "12"	false
"2" > "12"	true
"2" == "12"	false

3. Mảng

Trong JS, mảng có thể được gán giá trị ngay khi khai báo theo 2 cách:

- Cách 1: `var monHoc = ["Toán", "Lý", "Hóa", 10];`
- Cách 2: `var monHoc = new Array("Toán", "Lý", "Hóa", 10);`

Cách truy cập vào các phần tử của mảng là thông qua chỉ số `i`, cách duyệt mảng cũng sử dụng vòng lặp tương tự như ngôn ngữ C/C++.

[3] Not A Number

Thuộc tính `length` cho biết số phần tử của mảng: `monHoc.length` sẽ cho giá trị là 4.

Để thêm phần tử vào cuối mảng, chúng ta có thể dùng hàm `push()` hoặc dùng chỉ số như sau:

```
monHoc.push("Tin học");
monHoc[monHoc.length] = "Mỹ thuật";
```

Toàn bộ mảng có thể được truy cập thông qua tên mảng:

```
document.write(monHoc);
```

4. Cấu trúc rẽ nhánh và vòng lặp

Cấu trúc rẽ nhánh (`if`, `if-else` và `switch`) và vòng lặp (`for`, `while`, `do-while`) của JS tương tự như ngôn ngữ C/C++.

Ngoài ra, JS cung cấp vòng lặp `for/of` và `for/in` để duyệt qua từng phần tử của các kiểu dữ liệu có tính lặp (điển hình là mảng):

```
var x;
for (x of monHoc) {
    document.write(x + "<br>");
}
for (x in monHoc) {
    document.write(monHoc[x] + "<br>");
}
```

5. Hàm

Cú pháp khai báo và cài đặt hàm trong JS:

```
function <Tên hàm>(<Danh sách tham số>) {
    // Các câu lệnh cần thực hiện
}
```

Ví dụ:

```
function Cong(a, b) {
    var c = a + b;
    document.write(c);
}
```

Hàm và tham số của hàm trong JS không cần quy định kiểu dữ liệu.

III. Tương tác với trang web

1. Truy cập đến các đối tượng HTML

Trong JS, đối tượng[4] `document` tượng trưng cho toàn bộ trang web.

Để truy cập được những đối tượng trong trang web, JS cung cấp các phương thức[5] sau dành cho đối tượng `document`:

- `getElementById()`: Tìm thẻ HTML theo `id`:

```
document.getElementById("demo");
```

- `getElementByClassName()`: Tìm thẻ HTML theo `class`:

```
document.getElementByClassName("imgSkin");
```

- `getElementByTagName()`: Tìm thẻ HTML theo tên thẻ:

```
document.getElementByTagName("p");
```

- `querySelector()` và `querySelectorAll()`: Tìm thẻ HTML theo selector CSS:

```
document.querySelectorAll("p.inDam");
```

---

[4] Object. Hiện tại, hiểu đơn giản thì đối tượng nghĩa là biến (variable).

[5] Method. Hiện tại, hiểu đơn giản thì phương thức nghĩa là hàm (function).

Ngoại trừ phương thức đầu tiên trả về 1 thẻ duy nhất, 3 phương thức còn lại sẽ trả về danh sách thẻ dưới dạng mảng.

2. Tương tác với các đối tượng HTML

Giả sử có thẻ HTML sau:

```
<a id="demo" href="http://www.google.com" style="color: red;">Hello</a>
```

Câu lệnh JS để truy cập được thẻ `<a>` này:

```
var e = document.getElementById("demo");
```

JS cho phép lấy và thay đổi nội dung, thuộc tính, định dạng CSS của thẻ theo những cách sau:

- Dùng thuộc tính `innerHTML` để lấy và thay đổi nội dung thẻ (nếu là control của form như textbox, checkbox… thì dùng thuộc tính `value`):

```
document.write(e.innerHTML);
e.innerHTML = "Xin chào";
```

- Dùng tên thuộc tính HTML để lấy và thay đổi giá trị thuộc tính HTML:

```
document.write(e.href);
e.href = "http://www.facebook.com";
```

- Dùng phương thức `setAttribute()` để thay đổi giá trị thuộc tính HTML:

```
e.setAttribute("href", "http://www.facebook.com");
```

- Dùng thuộc tính `style` để lấy và thay đổi giá trị thuộc tính CSS:

```
document.write(e.style.color);
e.style.color = "blue";
```

3. Xuất dữ liệu lên trang web

JS hỗ trợ 4 cách xuất dữ liệu lên trang web:

- Hiển thị trực tiếp vào thẻ HTML bằng thuộc tính `innerHTML` (xem phần III.2 ở trên).
- Hiển thị vào nội dung trang web bằng phương thức `document.write()`.
- Hiển thị bằng 1 cửa sổ popup bằng các phương thức `alert()`, `confirm()` và `prompt()`.
- Ghi vào console[6] của trình duyệt bằng phương thức `console.log()`.

IV. Xử lí sự kiện

Sự kiện (*event*) là 1 hành động được diễn ra khi người dùng tương tác với 1 đối tượng nào đó của trang web. Các đối tượng HTML hầu như đều hỗ trợ các sự kiện để giúp lập trình viên cài đặt các đoạn mã JS phản hồi lại với các tương tác của người dùng.

Danh sách các sự kiện thường gặp trong trang web:

Tên event	Ý nghĩa
`blur`	Đối tượng bị mất focus
`change`	Nội dung của đối tượng bị thay đổi (thường áp dụng cho các control trong form)
`click`	Click vào 1 đối tượng
`dblclick`	Double-click vào 1 đối tượng
`focus`	Đối tượng có focus
`keydown`	Nhấn 1 phím
`keyup`	Thả 1 phím

[6] Xem ở chế độ Developer (phím tắt F12)

keypress	Nhấn và thả 1 phím
mousedown	Nhấn chuột
mouseup	Thả chuột
mouseover	Con trỏ chuột đi vào 1 đối tượng
mouseout	Con trỏ chuột đi ra khỏi đối tượng
mousemove	Con trỏ chuột đi chuyển bên trong đối tượng
submit	Form được submit

Việc cài đặt các đoạn mã JS cho các sự kiện được gọi là xử lí sự kiện (*event handling*). Mỗi sự kiện đều có thể được cài đặt thông qua 1 thuộc tính tương ứng của đối tượng HTML. Tên thuộc tính này có dạng on<tên event>.

Ví dụ: Khi click vào 1 nút trên trang web, hiển thị dòng chữ "Xin chào!" bằng popup:

- Cách 1:

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

- Cách 2:

```
<script>
	function ShowMsg() {
		alert("Xin chào!");
	}
</script>
<button onclick="ShowMsg()">Click me</button>
```

V. Bài tập

Thực hiện các bài tập dưới đây, mỗi bài tập nằm trong 1 trang web riêng:

1. Thiết kế 1 trang web gồm 1 textbox để nhập họ tên và 1 button Đăng nhập.
   Yêu cầu: Sau khi nhập họ tên và nhấn nút Đăng nhập, hiển thị lời chào "Xin chào abc!" (thay abc bằng họ tên người dùng vừa nhập).
2. Thiết kế 1 trang web gồm 2 textbox để nhập tên tài khoản, mật khẩu và 1 button Đăng nhập.
   Yêu cầu:
   - Khi di chuyển chuột lên textbox Mật khẩu thì hiển thị mật khẩu ở dạng plain text, khi di chuyển chuột ra khỏi textbox Mật khẩu thì hiển thị mật khẩu ở dạng password như bình thường.
   - Sau khi nhập thông tin và nhấn nút Đăng nhập, kiểm tra nếu tên tài khoản là admin và mật khẩu là 123456 thì thông báo "Đăng nhập thành công", ngược lại thông báo "Đăng nhập thất bại".
3. Thiết kế 1 trang web gồm 1 textbox cho phép người dùng nhập vào 1 số nguyên dương.
   Yêu cầu: Nếu người dùng không nhập đúng số nguyên dương thì thông báo lỗi, ngược lại kiểm tra xem số đó có phải là số nguyên tố không.