

LAB 7: JAVASCRIPT

1. Mục tiêu

- Giải thích được công dụng của Javascript.
- Định nghĩa các khái niệm cơ bản trong Javascript.
- Tạo và sử dụng hàm trong Javascript.
- Sử dụng Javascript để kiểm tra form.

2. Tóm tắt lý thuyết

3.1 Javascript là gì?

- Javascript là một ngôn ngữ lập trình được xây dựng sẵn trong các trình duyệt web, dùng để bổ sung tính tương tác cho các website như thay đổi màu sắc, kích thước, hình ảnh, kiểm tra giá trị nhập...
- Hạn chế của Javascript là không thể giao tiếp với máy chủ, làm việc khác nhau trên các trình duyệt khác nhau.

3.2 Sử dụng Javascript

Các đoạn Javascript đặt trong cặp tag `<script> </script>`. Cặp tag này có thể đặt trong `<head></head>` hay trong `<body></body>`. Ngoài ra các đoạn Javascript có thể đặt trong 1 file có phần mở rộng là .js. Để sử dụng các đoạn script trong 1 file riêng (ví dụ như abc.js)

```
<script type="text/javascript" src="abc.js"> </script>
```

Ví dụ:

```
<head>
  <title>Ví dụ Javascript</title>
  <script type="text/javascript">
    alert("Hello world!");
  </script>
</head>
```

3.3 Các khái niệm cơ bản trong Javascript

a. Biến

Khai báo biến: `var tên_biến = giá_trị_khởi_tạo;`

Ví dụ: `var i = 10;`

Từ khóa `var` có thể bỏ qua (có nghĩa là không cần khai báo biến)

Quy tắc đặt tên biến: có thể dùng chữ cái, số và ký tự “_” để đặt tên biến, nhưng tên biến không được bắt đầu bằng số. Tên biến không được trùng với các từ khóa của Javascript. Javascript phân biệt chữ hoa và chữ thường nên biến `a` và biến `A` là 2 biến khác nhau.

b. Ghi chú

Sử dụng `//` : ghi chú 1 dòng

Sử dụng `/* */`: ghi chú nhiều dòng

3.4 Hàm trong Javascript

Hàm do người định nghĩa:

```
function ten_ham(các tham số)
{
    //khởi lệnh
}
```

Một số hàm thông dụng trong Javascript:

- **eval(s)**: thực thi chuỗi s. Ví dụ `eval("1+1")`, cho kết quả là 2.
- **isNaN(n)**: kiểm tra n có phải là số hay không. Ví dụ: `isNaN("abc")` cho kết quả là true, `isNaN("7")` cho kết quả là false.
- **Number(o)**: chuyển đổi tượng o thành kiểu số. Ví dụ: `Number(true)` cho kết quả là 1, `Number("123")` cho kết quả là 123.
- **parseInt(s [,x])**: chuyển chuỗi s, 1 số có cơ số x, sang kiểu số nguyên. Nếu không chuyển được sẽ trả về NaN. Ví dụ: `parseInt("101",2)` cho kết quả là 5, `parseInt("A",16)` cho kết quả là 10, `parseInt("abc")` cho kết quả là NaN.

3.5 Sự kiện trong Javascript

Các sự kiện cung cấp các tương tác với các cửa sổ trình duyệt và tài liệu hiện hành được load trong trang web và các đối tượng có trong trang web. Một vài sự kiện trong Javascript:

onblur	: Xảy ra khi một đối tượng not active
onchange	: Xảy ra khi giá trị của một đối tượng bị thay đổi
onclick	: Xảy ra khi click vào đối tượng
ondblclick	: Xảy ra khi double click vào đối tượng
onerror	: Xảy ra khi javascript có lỗi
onfocus	: Xảy ra khi đối tượng có quyền điều khiển
onload	: Xảy ra khi khởi tạo một trang
onreset	: Xảy ra khi reset form
onscroll	: Xảy ra khi cuộn (scroll) trang
onselect	: Xảy ra khi chọn text trong textfield hay textarea.
onsubmit	: Xảy ra khi User xác nhận đã nhập xong dữ liệu
onmousedown	: Xảy ra khi nhấn chuột
onmousemove	: Xảy ra khi di chuyển chuột
onmouseout	: Xảy ra khi mouse di chuyển ra khỏi đối tượng
onmouseover	: Xảy ra khi mouse di chuyển qua đối tượng
onmouseup	: Xảy ra khi nhả chuột
onkeydown	: Xảy ra khi nhấn 1 phím
onkeypress	: Xảy ra khi nhấn và nhả 1 phím

onkeyup : Xảy ra khi nhấn 1 phím

Javascript có thể thực hiện một đoạn code sau 1 khoảng thời gian xác định gọi là sự kiện thời gian.

var t=setTimeout("s",x): thực thi đoạn code s sau x millisecond.

clearTimeout(t): hủy bỏ thiết lập setTimeout được gán cho biến t.

var t=setInterval("s",x): thực thi đoạn code s sau **mỗi** x millisecond.

clearInterval(t): hủy bỏ thiết lập setInterval được gán cho biến t.

Sử dụng sự kiện trong Javascript:

`<html_tag các_thuộc_tính sự_kiến= "khởi_lệnh">`

Ví dụ khi click vào nút Cộng (sự kiện **onclick** xảy ra trên đối tượng nút Cộng), Javascript lấy giá trị trong ô số 1 + giá trị trong ô số 2 và gán giá trị kết quả vào ô kết quả.

Số 1:	<input type="text" value="3"/>	txt1
Số 2:	<input type="text" value="4"/>	txt2
	<input type="button" value="Cộng"/>	
KQ:	<input type="text" value="7"/>	

```

<input type="button" name="btnKQ" id="btnKQ"
value="Cộng"
onclick="cong(document.getElementById('txt1').
value,document.getElementById('txt2').value);" />
        
```

Hình 7.1: Ví dụ cộng 2 số

3.6 Javascript & CSS

2.6.1. Thay đổi thuộc tính

Tham chiếu đến html tag (element) theo mô hình DOM (lab8) hay sử dụng hàm getElementById() hay getElementsByName(). Sau đó thay đổi thuộc tính cho element đó.

Ví dụ:

```

function doiHinh()
{
    document.all.hinh.src="images/hinh2.jpg"
}
        
```

2.6.2. Thay đổi style

Giả sử 1 tham chiếu đến element cần thay đổi style được gán cho biến o.

- Thay đổi thuộc tính style
- o.style.tên_thuộc_tính = giá_trị;

Ví dụ:

```

function doiMau()
{
    var o= document.getElementById("p1");
    o.style.color="red";
    //có thể viết rút gọn như sau:
    // document.getElementById("p1").style.color="red";
}
        
```

```
}
```

➤ Thay đổi tên lớp

```
o.style.className=tên_lớp;
```

Ví dụ:

```
function doiStyle()
{
    var o= document.getElementById("p1");
    o.className="newClass";
    //có thể viết rút gọn như sau:
    // document.getElementById("p1").className="newClass";
}
```

3. Ví dụ minh họa

3.1 Ví dụ 1

Sử dụng Javascript để kiểm tra form trong ví dụ trên bao gồm: giá trị trong ô Số 1, Số 2 phải là số và không được để trống khi click vào nút Cộng. Xem trang **lab7_vidu1.html**

3.2 Ví dụ 2

Giả sử website đang xây dựng được lưu trong thư mục lab7 và trong thư mục này có thư mục images/pro_img/phones và trang **lab7_vidu2.html**. Trong thư mục images/pro_img/phones có 2 hình là 6Splus1.png và 6Splus2.png. Ví dụ 2 sẽ thực hiện như sau: chèn hình 6Splus1.png vào trang lab7_vidu2.html, khi rê chuột vào hình 6Splus1 thì hình này sẽ chuyển thành hình 6Splus2. Xem trang **lab7_vidu2.html**.

4. Thời gian thực hành

120 phút

5. Thang điểm

Thang điểm tối đa: 10 điểm/Lab

6. Thực hành

Bài1: (1đ)

Trong ví dụ 1, 2 ô nhập giá trị số 1 và số 2 được **kiểm tra** khi nào? Hãy chỉnh sửa để 2 giá trị chỉ được kiểm tra khi người sử dụng chuyển quyền điều khiển sang đối tượng khác. Lưu kết quả làm được với tên file **lab7_bai1.html**

Bài 2 (1.5 đ)

Thay đổi file **lab7_vidu1.js** thành **lab7_bai2.js** để thay đổi màu txt1 và txt2 thành màu đỏ nếu giá trị nhập vào txt1 và txt2 không hợp lệ. Chuyển lại màu trắng nếu người dùng sửa lại đúng.

Bài 3 (2.5 đ)

Tạo 1 trang **lab7_bai3.html** có màu nền thay đổi ngẫu nhiên mỗi giây.

Gợi ý: Dùng `Math.random()` để tạo số ngẫu nhiên từ 0 đến 255, chuyển các số ngẫu nhiên vừa tạo sang hệ 16. Ghép 3 thành phần Red, Green, Blue thành 1 chuỗi kèm theo “#” trước chuỗi.

Bài 4 (3 đ)

Trong thư mục `lab7/images/layout` có các hình `banner1.png`, `banner2.png`, `banner3.png`. Hãy thiết kế trang web đơn giản như hình 7.1 và viết code javascript cho các nút Hình 1, Hình 2, Hình 3, Hình trước, Hình sau để các nút này thực hiện chức năng sau:



Chức năng nút **Hình trước** dùng để chuyển sang hình trước đó. Ví dụ `banner2.png` đang được hiển thị thì sẽ chuyển sang `banner1.jpg`. Nếu click thêm lần nữa sẽ chuyển sang `banner3.png`.

Chức năng nút **Hình sau** dùng để hiển thị hình kế tiếp. Nếu `banner3.png` đang được hiển thị sẽ chuyển sang `banner1.png`

Chức năng các nút **Hình 1**, **Hình 2**, **Hình 3** dùng để hiển thị hình tương ứng.

Bài 5 (2 đ)

Sử dụng **lab7_bai5_mau.html** và kết hợp với bài 4 để đạt được kết quả như sau: Các hình tự động thay đổi từ hình `banner1.png` sang `banner2.png`, `banner3.png` và quay về `banner1.png` mỗi 1 giây. Khi đưa chuột vào khung hình thì dừng tại hình đó, lấy chuột ra sẽ chuyển sang hình kế tiếp như cũ.