THIẾT KẾ WEB

Phần II:

Java Script và CSS

Nội dung

- Chương 1: CSS.
- Chương 2: Javascript.

Chương I: CSS

CSS

- CSS=Cascading Style Sheet: Mẫu quy định cách thức thể hiện các thẻ HTML
- Style được đưa vào HTML 4.0 để giải quyết một số vấn đề.
- Giúp tiết kiệm được rất nhiều thời gian và công sức cho việc thiết kế web.
- Có thể định nghĩa nhiều style vào một thẻ HTML (Cascading)

Các loại style

- Có 4 loại style:
  - *Inline Style* (Style được qui định trong 1 thẻ HTML cụ thể)
  - *Internal Style* (Style được qui định trong phần <HEAD> của 1 trang HTML)
  - *External Style* (style được qui định trong file .CSS ngoài)
  - *Browser Default* (thiết lập mặc định của trình duyệt)
- Thứ tự ưu tiên: Mức ưu tiên giảm dần từ trên xuống

Cách chèn CSS

- Đặt trong <head>…</head>
- Với Internal style:

```
<style type="text/css">
<!--
  Nội dung định nghĩa style
-->
</style>
```

Cách chèn CSS (tt)

- Với External style:
  - Định nghĩa style trong file riêng (thường có đuôi .CSS)
  - Nhúng file CSS đã định nghĩa vào trang web:

```
<link href="địa chỉ file"
rel="stylesheet" type="text/css">
```

- Với Inline style:

```
<tên_thẻ style="tt1:gt1;tt2:gt2;…">
```

Khai báo và sử dụng style

Chú ý khi viết style

- Style phân biệt chữ hoa, chữ thường
- Để ghi chú trong style sử dụng:

```
/*
  Đoạn ghi chú
*/
```

Khai báo style

```
selector {
    Property1: Value1;
    Property2: Value2;
}
```

Style áp dụng cho thẻ cụ thể

- Trường hợp 1 thẻ: Đặt *selector* là *tên_thẻ*

```
p {
  color: red;
}
```

- Khai báo đồng thời nhiều thẻ: Viết danh sách tên thẻ phân cách bởi dấu phảy

```
h1,h2,h3,h4,h5,h6{
  font-family:arial;
}
```

Tạo lớp

- Gắn với thẻ cụ thể: Đặt *selector* là **`tên_thẻ.tên_lớp`**

```
p.loai1{
  color:red;
}
p.loai2{
  color:blue;
}
```

- Không gắn với thẻ cụ thể: bỏ phần *tên_thẻ* đi, giữ lại dấu chấm:

```
.loai3{
  color:green;
}
```

Sử dụng lớp

- Đặt thuộc tính `class` của thẻ=“tên_lớp”:

```
<tên_thẻ class=“tên_lớp”>
```

- Ví dụ:

```
<p class=“loai1”>Đoạn này màu đỏ</p>
<h1 class=“loai2”>Style không có hiệu
  lực</h1>
<h2 class=“loai3”>Tiêu đề màu xanh</h3>
```

Định danh

- Tương tự như class. Thay dấu chấm (.) thành dấu thăng (#).
- Cho thẻ cụ thể: **tên_thẻ#định_danh{…}**
- Tổng quát: **#định_danh{…}**
- Ví dụ:

```
p#doan1{
  color:red;
}
#loai2{
  color:blue;
}
```

Sử dụng định danh

- Mỗi định danh là duy nhất trên trang
- Đặt thuộc tính `id` của thẻ = `định_danh`

```
<tên_thẻ id="định_danh">
```

- Ví dụ:

```
<p id="doan1">Đoạn này màu đỏ</p>
<h1 id="loai2">Tiêu đề xanh</h1>
```

Một số trường hợp cụ thể

CSS Basic

1. CSS Background
2. CSS Text
3. CSS Font
4. CSS Border
5. CSS Margin
6. CSS Padding
7. CSS List

NAM CAN THO UNIVERSITY

CSS Advanced

1. CSS Dimension
2. CSS Classification
3. CSS Positioning
4. CSS Pseudo-class
5. CSS Pseudo-element
6. CSS Media Types

CSS cho nền

Property	Description	Values	NN	IE	W3C
background	A shorthand property for setting all background properties in one declaration	*background-color* *background-image* *background-repeat* *background-attachment* *background-position*	6.0	4.0	CSS1
background-attachment	Sets whether a background image is fixed or scrolls with the rest of the page	scroll fixed	6.0	4.0	CSS1
background-color	Sets the background color of an element	*color-rgb* *color-hex* *color-name* transparent	4.0	4.0	CSS1

background-image	Sets an image as the background	*url* none	4.0	4.0	CSS1
background-position	Sets the starting position of a background image	top left top center top right center left center center center right bottom left bottom center bottom right *x-% y-%* *x-pos y-pos*	6.0	4.0	CSS1
background-repeat	Sets if/how a background image will be repeated	repeat repeat-x repeat-y no-repeat	4.0	4.0	CSS1

Property	Description	Possible Values	NN	IE	W3C
color	Sets the color of a text	*color*	4.0	3.0	CSS1
direction	Sets the text direction	ltr rtl			CSS2
letter-spacing	Increase or decrease the space between characters	normal *length*	6.0	4.0	CSS1
text-align	Aligns the text in an element	left right center justify	4.0	4.0	CSS1
text-decoration	Adds decoration to text	none underline overline line-through blink	4.0	4.0	CSS1
text-indent	Indents the first line of text in an element	*length* *%*	4.0	4.0	CSS1

CSS và cho bản (tt)

text-shadow		none *color* *length*			
text-transform	Controls the letters in an element	none capitalize uppercase lowercase	4.0	4.0	CSS1
unicode-bidi		normal embed bidi-override		5.0	CSS2
white-space	Sets how white space inside an element is handled	normal pre nowrap	4.0	5.5	CSS1
word-spacing	Increase or decrease the space between words	normal *length*	6.0	6.0	CSS1

CSS và font

Property	Description	Values	NN	IE	W3C
font	A shorthand property for setting all of the properties for a font in one declaration	*font-style* *font-variant* *font-weight* *font-size/line-height* *font-family* caption icon menu message-box small-caption status-bar	4.0	4.0	CSS1
font-family	A prioritized list of font family names and/or generic family names for an element	*family-name* *generic-family*	4.0	3.0	CSS1

CSS và font (tt)

font-size	Sets the size of a font	xx-small x-small small medium large x-large xx-large smaller larger *length* *%*	4.0	3.0	CSS1
font-size-adjust	Specifies an aspect value for an element that will preserve the x-height of the first-choice font	none *number*			CSS2

CSS và font (tt)

font-stretch	Condenses or expands the current font-family	normal wider narrower ultra-condensed extra-condensed condensed semi-condensed semi-expanded expanded extra-expanded ultra-expanded			CSS2
font-style	Sets the style of the font	normal italic oblique	4.0	4.0	CSS1

CSS và font (tt)

font-variant	Displays text in a small-caps font or a normal font	normal small-caps	6.0	4.0	CSS1
font-weight	Sets the weight of a font	normal bold bolder lighter 100 200 300 400 500 600 700 800 900	4.0	4.0	CSS1

CSS và đường viền

Property	Description	Values	NN	IE	W3C
border	A shorthand property for setting all of the properties for the four borders in one declaration	*border-width* *border-style* *border-color*	4.0	4.0	CSS1
border-bottom	A shorthand property for setting all of the properties for the bottom border in one declaration	*border-bottom-width* *border-style* *border-color*	6.0	4.0	CSS1
border-bottom-color	Sets the color of the bottom border	*border-color*	6.0	4.0	CSS2
border-bottom-style	Sets the style of the bottom border	*border-style*	6.0	4.0	CSS2
border-bottom-width	Sets the width of the bottom border	thin medium thick *length*	4.0	4.0	CSS1
border-color	Sets the color of the four	*color*	6.0	4.0	CSS1

border-left	A shorthand property for setting all of the properties for the left border in one declaration	*border-left-width* *border-style* *border-color*	6.0	4.0	CSS1
border-left-color	Sets the color of the left border	*border-color*	6.0	4.0	CSS2
border-left-style	Sets the style of the left border	*border-style*	6.0	4.0	CSS2
border-left-width	Sets the width of the left border	thin medium thick *length*	4.0	4.0	CSS1

border-right	A shorthand property for setting all of the properties for the right border in one declaration	*border-right-width* *border-style* *border-color*	6.0	4.0	CSS1
border-right-color	Sets the color of the right border	*border-color*	6.0	4.0	CSS2
border-right-style	Sets the style of the right border	*border-style*	6.0	4.0	CSS2
border-right-width	Sets the width of the right border	thin medium thick *length*	4.0	4.0	CSS1
border-style	Sets the style of the four borders, can have from one to four styles	none hidden dotted dashed solid double groove ridge inset outset	6.0	4.0	CSS1

border-top	A shorthand property for setting all of the properties for the top border in one declaration	*border-top-width* *border-style* *border-color*	6.0	4.0	CSS1
border-top-color	Sets the color of the top border	*border-color*	6.0	4.0	CSS2
border-top-style	Sets the style of the top border	*border-style*	6.0	4.0	CSS2
border-top-width	Sets the width of the top border	thin medium thick *length*	4.0	4.0	CSS1
border-width	A shorthand property for setting the width of the four borders in one declaration, can have from one to four values	thin medium thick *length*	4.0	4.0	CSS

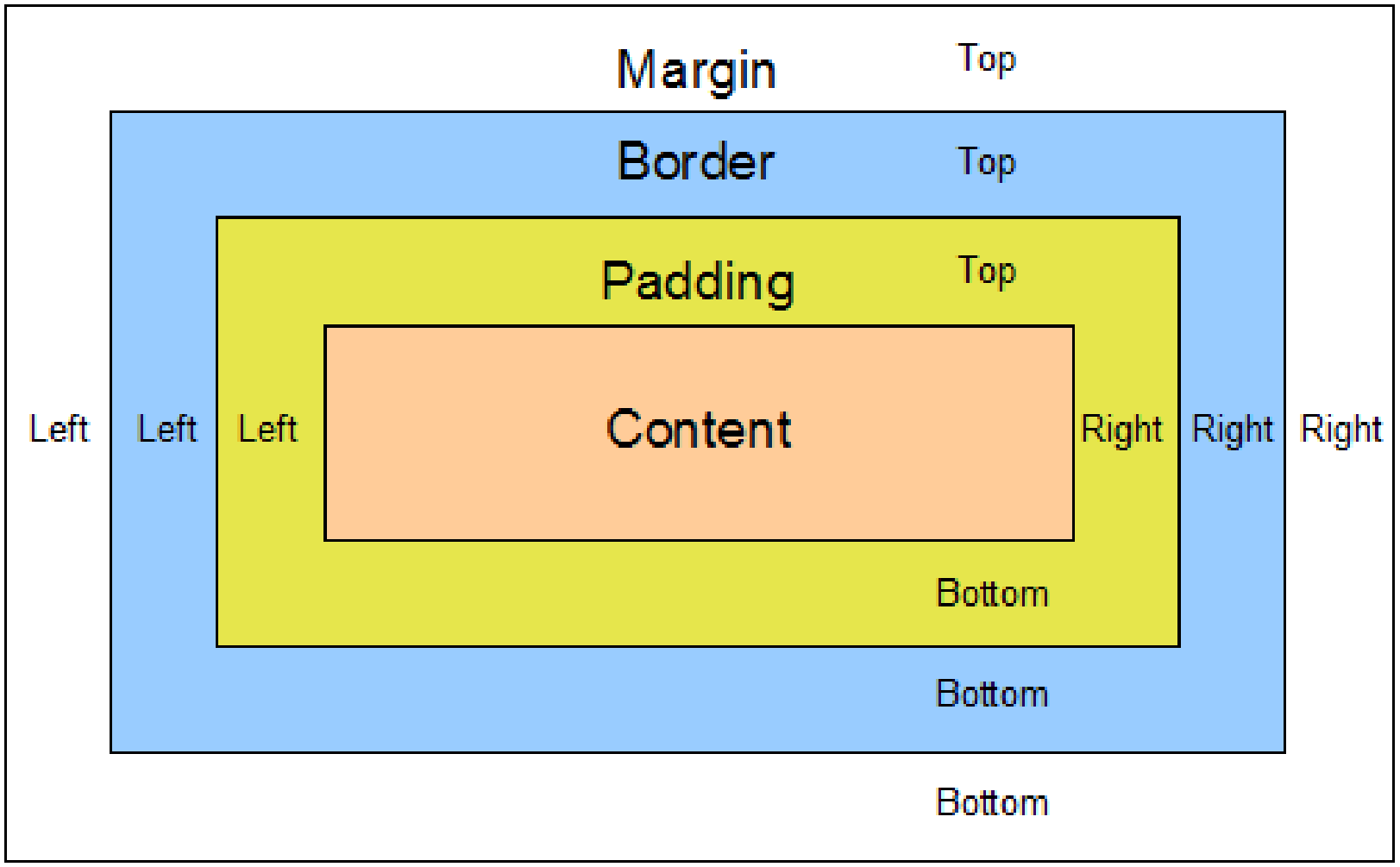
Property	Description	Values	NN	IE	W3C
clear	Sets the sides of an element where other floating elements are not allowed	left right both none	4.0	4.0	CSS1
cursor	Specifies the type of cursor to be displayed	*url* auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help	6.0	4.0	CSS2

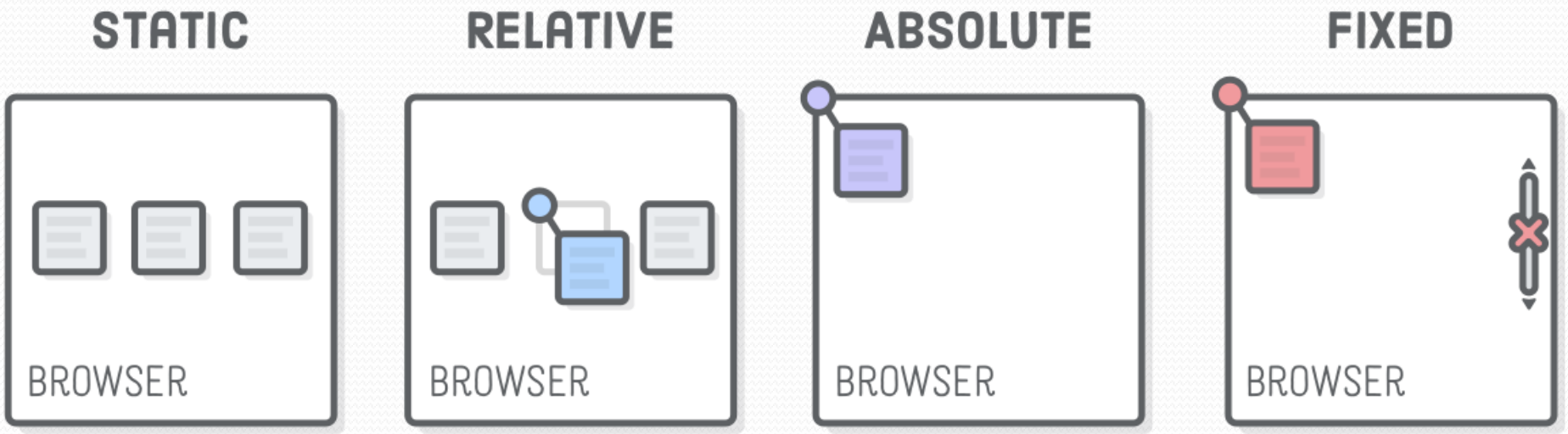
display	Sets how/if an element is displayed	none inline block list-item run-in compact marker table inline-table table-row-group table-header-group table-footer-group table-row table-column-group table-column table-cell table-caption	4.0	4.0	CSS1

float	Sets where an image or a text will appear in another element	left right none	4.0	4.0	CSS1
position	Places an element in a static, relative, absolute or fixed position	static relative absolute fixed	4.0	4.0	CSS2
visibility	Sets if an element should be visible or invisible	visible hidden collapse	6.0	4.0	CSS2

CSS thông dụng

Border-radius: bo tròn khung đường viền, tính bằng pixel


Margin
Top
Border
Top
Padding
Top
Content
Left
Left
Left
Right
Right
Right
Bottom
Bottom
Bottom


STATIC
BROWSER
RELATIVE
BROWSER
ABSOLUTE
BROWSER
FIXED
BROWSER

CSS Event

1. Active: giữ chuột trái
2. Hover: rê chuột vào
3. Focus: chọn vào đối tượng
4. Transition:Thời gian thực hiện hiệu ứng

Thực hành

1. Áp dụng CSS dạng thẻ, dạng class, dạng ID
2. Set background color
3. Set Background image
4. Áp dụng font chữ in đậm, nghiêng
5. Set boder solid, màu, width
6. Set Border từng phần: Bottom, top, left, right
7. Set Position
8. Set visibility

Thực hành

1. Div 1,2 dạng class có kèm thẻ và Div 3 không kèm thẻ
2. Div 4,5,6 dạng Id

Chương II:
Ngôn ngữ Javascript

JavaScript

- JavaScript là ngôn ngữ kịch bản dùng để tạo các client-side scripts và server-side scripts.
- JavaScript làm cho việc tạo các trang Web động và tương tác dễ dàng hơn
- JavaScript là một ngôn ngữ kịch bản được hãng Sun Microsystems và Netscape phát triển.
- JavaScript được phát triển từ Livescript. Của Netscape
- Các ứng dụng client chạy trên một trình duyệt như Netscape Navigator hoặc Internet Explorer.

Khả năng của Javascript

- JavaScript có thể tăng cường tính động và tính tương tác của các trang web.
  - Cung cấp sự tương tác người dùng
  - Thay đổi nội dung động
  - Xác nhận tính hợp lệ của dữ liệu

Quy tắc

- Sử dụng quy tắc chữ hoa
- Using Pairs
- Using Spaces
- Using Comments

Công cụ và môi trường thực thi

- Các công cụ sinh mã JavaScript
  - Thuận lợi khi soạn thảo
  - Mã lệnh sẵn có
- Môi trường thực thi
  - Các Scripting ở Client
  - Java Script trên Web Server

Chèn Javascript vào HTML

- Sử dụng thẻ SCRIPT:

```
<script language="JavaScript">
<!--
  JavaScript statements;
//-->
</script>
```

- Sử dụng một file JavaScript ở ngoài

```
<script language="JavaScript" src="filename.js">
</script>
```

- Sử dụng các biểu thức JavaScript trong các giá trị thuộc tính của thẻ
- Sử dụng trong các trình điều khiển sự kiện

Ví dụ

```
<HTML>
    <HEAD>
     <SCRIPT LANGUAGE = "Javascript">
      confirm ("Are you Sure?");
      alert("OK");
      document.write(" Thank You !");
     </SCRIPT>
   </HEAD>
</HTML>
```

Biến

- Biến là một vật chứa tham chiếu đến một vị trí ở bộ nhớ máy tính
- Nó được sử dụng để giữ giá trị và có thể thay đổi trong khi kịch bản thực thi
- Các biến tuân theo quy tắc đặt tên.
- Một biến được khai báo sử dụng từ khoá '**var**'.

  ví dụ: `var A = 10;`
- Các biến có một phạm vi được xác định trong khi chúng khai báo trong script.
  - Biến toàn cục
  - Biến cục bộ
- Nguyên dạng là các giá trị không đổi được dùng trong script.

Các kiểu dữ liệu

- JavaScript có một tập các kiểu dữ liệu.
  - Số (number)
  - Giá trị logic (boolean)
  - Chuỗi (String)
  - Giá trị rỗng Null
- Trong JavaScript, hai biến khác kiểu có thể kết hợp với nhau.

  ví dụ: `A = " This apple costs Rs." + 5`

  sẽ có kết quả là một chuỗi với giá trị là `"This apple costs Rs. 5"`.

Ví dụ

```
<HTML>
 <HEAD>
 <SCRIPT LANGUAGE = "Javascript">
   var A = "12" + 7.5;
   document.write(A);
   </SCRIPT>
  </HEAD>
</HTML>
```

Các kiểu nguyên thủy

- Integer – là các hệ thống số thập phân, thập lục phân và nhị phân.
- Floating- point(số thực) – Các số thập phân có phần thập phân sử dụng “e” or “”E”và theo sau là các số nguyên.
- String – là một chuỗi rỗng hay chuỗi ký tự được đặt trong cặp ngoặc đơn hoặc ngoặc kép
- Boolean–Kiểu này có hai giá trị: True or False.
- null - Kiểu null chỉ có một giá trị: null. Null hàm ý không có dữ liệu.

Toán tử

- Các toán tử xử lý một hoặc nhiều biến hoặc các giá trị (các toán hạng) và trả lại giá trị kết quả.
- JavaScript sử dụng cả hai toán tử một ngôi và hai ngôn.
- Các toán tử được phân loại phụ thuộc quan hệ chúng thực hiện như:
  - Toán tử số học
  - Toán tử so sánh
  - Toán tử logic
  - Toán tử chuỗi
  - Toán tử lượng giá
- Mức ưu tiên của toán tử

Ví dụ

```
<HTML>
   <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
    var x = 10;
    var y = 5;
    alert ("The value of x is "
           + x + "The value of y is  " + y);
   alert("x AND y = " + (x && y));
   alert("x OR y = " + (x || y));
   alert("NOT x = " + (!x));
  </SCRIPT>
 </HEAD>
</HTML>
```

Mảng

- Mảng được dùng để lưu trữ một dãy các biến với cùng một tên.
- Một số (chỉ số) dùng để phân biệt các giá trị khác nhau.
- Các mảng bắt đầu với chỉ số o trong JavaScript.
- Tạo mảng: Cú pháp

```
arrayObjectName = new Array([element0, element1,
..., elementN])
```

- Cộng các phần tử: Chúng ta có thể cộng các phần tử mảng khi chúng ta tạo nó. Ví dụ. emp[o] = "Ryan Dias"
- Các phần tử của một mảng có thể truy cập bằng tên Name hoặc chỉ số Index của phần tử.

Mảng (tt)

- Các phương thức của đối tượng mảng có thể dùng thao tác trên mảng.
- Các phương thức của đối tượng mảng bao gồm:
  - join
  - pop
  - push
  - reverse
  - shift
  - sort
- JavaScript hỗ trợ mảng nhiều chiều.

Lệnh rẽ nhánh

- Câu lệnh điều kiện được dùng để kiểm tra điều kiện. Kết quả xác định câu lệnh hoặc khối lệnh được thực thi.
- Các câu lệnh điều kiện bao gồm:
  - `If….. Else`
  - `Switch`

Lặp

- Cấu trúc điều khiển lặp trong chương trình là các lệnh lặp.
- Các kiểu lệnh lặp bao gồm:
  - For
  - Do .... While
  - While
  - Break & continue
  - For....in
  - with

Hàm

- JavaScript có sẵn các hàm đinh nghĩa trước dùng trong script.
- Một vài hàm định nghĩa trước trong JavaScript bao gồm:
  - Hàm eval
  - Hàm isNaN
- Hàm do người dùng tự tạo

```
function funcName(argument1,argument2,…){
        statements;
}
```

- Gọi hàm
- Câu lệnh Return

Các đối tượng cơ bản trong Javascript

Đối tượng

- Thuộc tính (biến) dùng để định nghĩa đối tượng và các phương thức (hàm) tác động tới dữ liệu đều nằm trong đối tượng.
- Ví dụ: một chiếc xe hơi là một đối tượng. Các thuộc tính của nó là cấu tạo, kiểu dáng và màu sắc. Hầu hết các chiếc xe hơi đều có một vài phương thức chung như go(), brake(), reverse().

Thuộc tính và phương thức

- Để truy cập thuộc tính của đối tượng, chúng ta phải chỉ ra tên đối tượng và thuộc tính của nó:

```
objectName.propertyName
```

- Để truy cập phương thức của đối tượng, chúng ta phải chỉ ra tên đối tượng và thuộc tính của nó:

```
objectName.method()
```

Sử dụng đối tượng

- Khi tạo trang web, chúng ta cần sử dụng:
  - Các đối tượng trình duyệt
  - Các đối tượng có sẵn (thay đổi phụ thuộc vào ngôn ngữ kịch bản được sử dụng)
  - HTML elements
- Chúng ta cũng có thể tạo ra các đối tượng để sử dụng theo yêu cầu của mình.

Từ khóa this

- Giá trị của nó chỉ ra đối tượng hiện hành và có thể có các thuộc tính chuẩn chẳng hạn như tên, độ dài, và giá trị được áp dụng phù hợp.

Lệnh for…in

- Câu lệnh `for...in` được dùng để lặp mỗi thuộc tính của đối tượng hoặc mỗi phần tử của một mảng.
- Cú pháp:

```
for (variable in object)  {
    statements;
}
```

with

- Câu lệnh with được dùng để thực thi tập hợp các lệnh mà các lệnh này dùng các phương thức của cùng một loại đối tượng.
- thuộc tính được gán cho đối tượng đã được xác định trong câu lệnh with.
- Cú pháp:

```
with (object) {
    statements;
}
```

Toán tử new

- Toán tử **new** được dùng để tạo ra một thực thể mới của một loại đối tượng
- Đối tượng có thể có sẵn hoặc do người dùng định nghĩa
  - objectName = new objectType (param1 [,param2] ...[,paramN])
  - Trong đó:
- objectName là tên của thực thể đối tượng mới.
- ObjectType là một hàm quyết định loại của đối tượng. Ví dụ Array.
- Param[1, 2, . .] là các giá trị thuộc tính của đối tượng.

Hàm eval

- Hàm `eval` được dùng để đánh giá một chuỗi mã lệnh mà không cần tham chiếu đến bất cứ đối tượng cụ thể nào.
- Chuỗi có thể là một biểu thức JavaScript, một câu lệnh hoặc một nhóm câu lệnh
- Biểu thức có thể bao gồm nhiều biến và nhiều thuộc tính của một đối tượng.

```
var x = 5;
var z = 10;
document.write(eval("x + z + 5"));
```

Đối tượng string

- Đối tượng **String** được dùng để thao tác và làm việc với chuỗi văn bản.
- Chúng ta có thể tách chuỗi ra thành các chuỗi con và biến đổi chuỗi đó thành các chuỗi hoa hoặc thường trong một chương trình.
- Cú pháp tổng quát:

```
stringName.propertyName
```

hay

```
stringName.methodName
```

Tạo đối tượng string

- Có 3 phương thức khác nhau để tạo ra chuỗi.
  - Dùng lệnh var và gán cho nó một giá trị.
  - Dùng một toán tử (=) có gán với một tên biến.
  - Dùng hàm khởi tạo String (string).

Đối tượng Math

- Đối tượng Math có các thuộc tính và phương thức biểu thị các phép tính toán học nâng cao.

```
function doCalc(x) {
   var a;
   a = Math.PI * x * x;
   alert ("The area of a circle with a
   radius of " + x + " is " + a);
}
```

Đối tượng Date

- Date là một đối tượng có sẵn chứa thông tin về ngày và giờ.
- Đối tượng Date không có thuộc tính nào.
- Nó có nhiều phương thức dùng để thiết lập, lấy và xử lý các thông tin về thời gian.

Đối tượng Date (tt)

- Đối tượng Date lưu trữ thời gian theo số mili giây tính từ 1/1/1970 00:00:00
- DateObject = new Date(parameters)

Xử lý sự kiện

- Các sự kiện JavaScript hỗ trợ

  - onClick
  - onChange
  - onFocus
  - onBlur
  - onMouseOver

  - onMouseOut
  - onLoad
  - onSubmit
  - onMouseDown
  - onMouseUp

- Đặt bộ lắng nghe sự kiện:

```
<thẻ sựkiện="lệnh;">
```

Danh sách sự kiện đầy đủ

Event	Occurs when…
onabort	a user aborts page loading
onblur	a user leaves an object
onchange	a user changes the value of an object
onclick	a user clicks on an object
ondblclick	a user double-clicks on an object
onfocus	a user makes an object active
onkeydown	a keyboard key is on its way down
Onkeypress	a keyboard key is pressed
onkeyup	a keyboard key is released

Danh sách sự kiện đầy đủ (tt)

onload	a page is finished loading. **Note:** In Netscape this event occurs during the loading of a page!
onmousedown	a user presses a mouse-button
onmousemove	a cursor moves on an object
onmouseover	a cursor moves over an object
onmouseout	a cursor moves off an object
onmouseup	a user releases a mouse-button
onreset	a user resets a form
onselect	a user selects content on a page
onsubmit	a user submits a form
onunload	a user closes a page

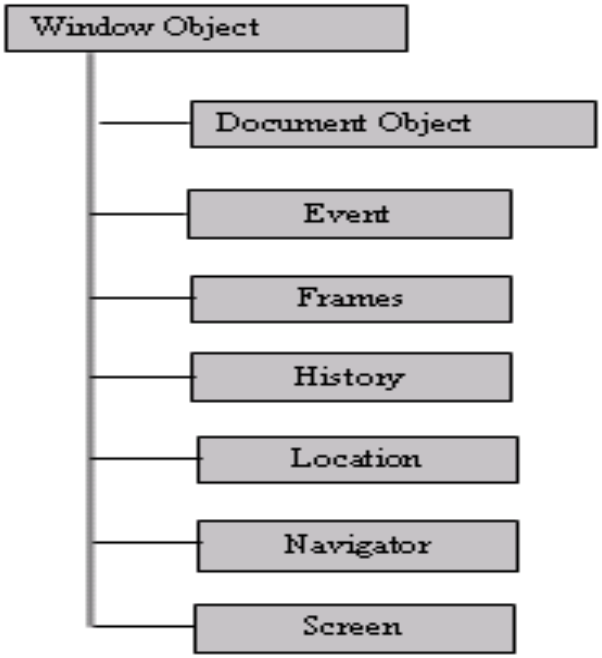
Các đối tượng của trình duyệt

DOM (Document Object Models)

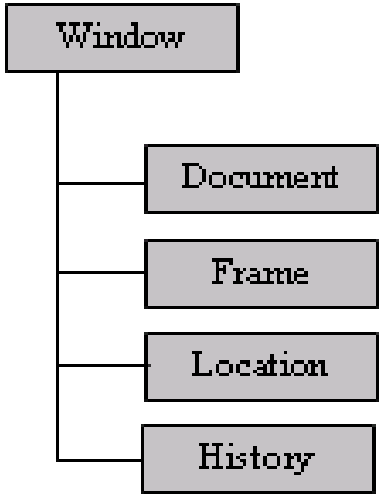
- Một tính năng quan trọng của JavaScript là ngôn ngữ dựa trên đối tượng.
- Giúp người dùng phát triển chương trình theo môđun và có thể sử dụng lại.
- Đối tượng được định nghĩa là một thực thể đơn nhất bao gồm các thuộc tính và phương thức.
- Thuộc tính là giá trị của một đối tượng.

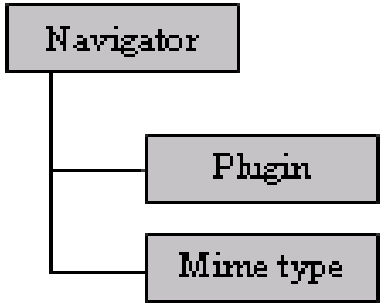
Các đối tượng trên trình duyệt

- Trình duyệt là một ứng dụng được sử dụng để hiển thị nội dung của tài liệu HTML.
- Các trình duyệt cũng đưa ra một số đối tượng có thể được truy cập và sử dụng trong script .
- Chi tiết các thuộc tính: tra tài liệu hoặc w3schools



IE Browser Objects





Netscape Browser Objects

Một vài ứng dụng

- Thay đổi thanh trạng thái, tiêu đề
- Tự động refresh
- Kiểm tra tính hợp lệ của dữ liệu form
- Hộp thoại tự tạo

Một vài ứng dụng (tiếp)

- Calendars
- Date & Time
- Document Effects
- Dynamic Content (Iframe & Ajax)
- Form Effects
- Games
- Image Effects
- Galleries, Mouseover, Slideshows

- ✓ Links & Tooltips
- ✓ Menus & Navigation
- ✓ CSS Based, Multi-levels
- ✓ Mouse and Cursor
- ✓ Scrollers
- ✓ Text Animations
- ✓ User/System Preference
- ✓ Window and Frames
- ✓ XML and RSS
- ✓ Other

Tham khảo

- http://www.javascriptkit.com
- http://www.dynamicdrive.com
- http://www.javascriptbank.com
- http://www.dhtmlcentral.com

Hiệu ứng chữ chạy trên trình thanh trạng thái của trình duyệt

Lý thuyết

- **`window`** là đối tượng quản lý cửa sổ trình duyệt.
- Đối tượng `window` có thuộc tính `status` để xác định thông báo tạm thời xuất hiện trên thanh trạng thái.
- VD: Để thể hiện dòng chữ Hello World trên thanh trạng thái ta sử dụng lệnh:

```
window.status = 'Hello World'
```

Lý thuyết (tt)

- Lệnh `setTimeout(f, n)` quy định sau khoảng thời gian n mili giây hàm f sẽ được gọi. (f là chuỗi lưu lệnh cần thực hiện)
- Giả sử `str` là một chuỗi ta có
  - `str.length`: Thuộc tính cho biết độ dài chuỗi
  - `str.substr(i, n)`: lấy ra n ký tự kể từ vị trí thứ i (Ký tự đầu tiên được đánh số là 0)

Lý thuyết (tt)

- Vài lệnh khác cùng nhóm setTimeout
  - `timeID = setTimeout(f, n)`
  - `clearTimeout(timeID)`: Hủy setTimeout
  - `intervalID = setInterval(f, n)`: Sau mỗi khoảng thời gian n ms lệnh f được gọi.
  - `clearInterval(intervalID)`: Hủy interval.

Giải thuật

- Ý tưởng giải thuật
  - Để có được cảm giác chữ chạy trên thanh trạng thái ta cần thay đổi thuộc tính status của cửa sổ bằng cách copy ký tự đầu của thanh dòng trạng thái hiện tại đưa xuống cuối cùng và lặp lại như vậy sau mỗi khoảng thời gian.
- Giải thuật: Giả sử ta có biến str đang lưu chuỗi cần chạy. Công việc thực hiện như sau:
  - B1: Thể hiện chuỗi str lên thanh trạng thái. Chuyển sang bước 2
  - B2: Chuyển ký tự đầu của str về cuối (bằng cách gán str = xâu con kể từ vị trí thứ 2 của str đến cuối + ký tự đầu tiên của str). Chuyển sang bước 3
  - B3: Trễ một khoảng thời gian rồi quay lại bước 1

Mã lệnh

```
<script language="javascript">
var str= 'Khoa CNTT-DH DNC'
//Đưa vào nhìn cho đẹp (có khoảng cách trống giữa 2 lần chạy)
for (i=str.length; i<100; i++){
  str = str + ' '
}

function ChuChay(){
  window.status = str
  str = str.substr(1,str.length-1) + str.substr(0,1);

  setTimeout(ChuChay,100)
}
ChuChay()
</script>
```

Phát triển

- Thay bằng nhiều dòng chữ chạy khác nhau (sử dụng mảng để lưu trữ)
- Chữ chạy theo nhiều cách khác nhau
- Cho chữ chạy trên thanh tiêu đề
- Cho chữ chạy trên một đối tượng khác

Các đối tượng của trình duyệt

Đối tượng window

- Tập hợp: frames[]
- Thuộc tính:
  - document
  - history
  - location
  - opener
  - status:

Đối tượng window

- Sự kiện:
  - onLoad
  - onUnload
- Phương thức
  - alert(strMessage)
  - confirm(strMessage)
  - prompt(strMessage, defaultText)
  - open(url, name, option, replace)

Đối tượng window

- Interval_ID = setInterval(strLệnh, Thời_gian)
- Timeout_ID = setTimeout(strLệnh, Thời_gian)
- clearInterval(Interval_ID)
- clearTimeout(Timeout_ID)

Đối tượng document

- Tập hợp
  - anchors[]
  - links[]
  - forms[]
  - images[]
- Thuộc tính
  - title
  - cookie
- Phương thức
  - getElementById(ID)
  - getElementByName(ten)
  - getElementByTagName(Ten_The)

Đối tượng document

- Truy xuất đến các form:
  - document.tên_form
- Truy xuất các đối tượng trong form:
  - objForm.Tên_ĐT
- Thuộc tính đối tượng:
  - value

Đối tượng document

- Truy xuất đối tượng theo Id:
  - document. getElementById(“**tênID**”)
- Thay đổi nội dung text thông thường trên trang :
  - document. getElementById(“**tênID**”).innerText
- Thay đổi nội dung hiển thị trên trang :
  - document. getElementById(“**tênID**”).innerHTML
- Xuất nội dung lên trang:
  - document. write(“**nội dung**”)

Đối tượng document

- Đặc biệt truy xuất đối tượng theo Id có dính đến thao tác chạy trên server tích hợp ASP.NET
  - document. getElementById(<%= """+**IdGet**.ClientID +"""%>)

Ví dụ

- Tạo form gồm các thông tin:
  - Tên truy cập
  - Mật khẩu
  - Nhập lại mật khẩu
- Kiểm tra dữ liệu vào có hợp lệ không?
  - Hợp lệ:
    - tên truy cập không rỗng
    - 2 mật khẩu giống nhau, khác rỗng, >4 ký tự

  Nếu hợp lệ được submit, ngược lại thông báo lỗi.

Ví dụ 2

- Web link dạng combo box: Trong combo box có tên các đơn vị liên kết. Người dùng chọn đơn vị nào thì hiện web site của đơn vị đó trong 1 cửa sổ.
- Làm đồng hồ điện tử (hiện số) trên trang web.
  - Hiển thị thời gian hiện tại
  - Hiển thị khoảng thời gian đến 1 đích nào đó