



Tema 3, parte 5

Estilos dinámicos

Cambio de estilos

Animaciones.

Javascript, parte 5

Javascript: estilos
dinámicos

Animaciones
con Javascript
y CSS3

- Utilizando la propiedad **style** del DOM, correspondiente al atributo HTML del mismo nombre
- **style** es un objeto con propiedades cuyo nombre se corresponde con CSS pero cambiando los guiones por las iniciales en mayúscula (camelCase) p.ej: font-size pasa a ser fontSize
- ¡¡Cuidado!! Las propiedades son Strings

```
<script language="JavaScript">
  function aumentar(elemento) {
    elemento.style.fontSize =
      parseInt(elemento.style.fontSize) + 2 + "px";
  }
</script>
<p id="texto" style="font-size:12px">Hola</p>
<input type="button" value="Aumentar"
  onclick="aumentar(document.getElementById('texto'));>
```

- Idea básica
 - Asignar posición o tamaño mediante CSS
 - Usar un **temporizador** para cambiar la posición o el tamaño cada cierto nº de segundos
- Definir temporizadores
 - setInterval(código-entre-comillas, milisegundos)

```
setInterval("alert('cansino')", 3000)
```
 - setInterval(nombre-funcion, milisegundos, arg1, arg2,...)

```
function saludo(mensaje) {  
    alert(mensaje);  
}  
setInterval(saludo, 3000, "cansino")
```
- setInterval devuelve un entero que identifica al temporizador y se puede utilizar para pararlo con clearInterval(identificador)

- En CSS3 se pueden hacer animaciones simples, cambiando gradualmente el valor de alguna/s propiedad/es
 - Todavía está en proceso de estandarización. Solo funciona en versiones recientes de Chrome/Safari, Opera y Firefox (4.0)
 - (!) En cada navegador estas propiedades CSS tienen un nombre distinto

```
p {
    background-color:white;
}

/* cuando se pase el ratón por encima, el color de fondo cambiará
gradualmente a amarillo, en 2 segundos */
p:hover {
    background-color: yellow;
    -webkit-transition: background-color 2s; /* sintaxis chrome/safari */
}
```

Mientras el estándar se “estabiliza”...

- Cada navegador usa sus propios nombres de propiedades, así aunque el estándar cambiara se nos asegura que la implementación propia de cada navegador no lo hará
- Pero eso nos fuerza a usar la sintaxis “oficial” más la de cada uno de los navegadores por separado

```
p {
  background-color:white;
}

p:hover {
  background-color: yellow;
  -webkit-transition: background-color 2s; /* Chrome/safari */
  -moz-transition: background-color 2s; /* Firefox */
  -o-transition: background-color 2s; /* Opera */
  transition: background-color 2s; /* estándar */
}
```

- Usando las propiedades Javascript equivalentes a las de CSS

```
<div id="box" style="background-color:yellow; width:500px;
height:100px; overflow:hidden;">
  Esto debería encogerse hasta desaparecer al hacer click
</div>
<input type="button" value="animar" onclick="transicion()">
<script type="text/javascript">
  function transicion() {
    var box = document.getElementById('box');
    //se pueden transicionar varias propiedades
    //simultáneamente, separándolas por comas
    //(también se puede hacer en CSS "puro")
    box.style.webkitTransition = 'width 3s, height 3s';
    box.style.width = '0px';
    box.style.height = '0px';
  }
</script>
```

Javascript, parte 5

Javascript: estilos
dinámicos

Estilos dinámicos con DOM 2

Manipular los estilos CSS
que no son “inline”

DOM Level 2 Style

- El DOM 1 solo permite acceder a los estilos “inline”

```
<p style="color:red" id="saludo"> hola </p>
```

- El **DOM Style** (un módulo del DOM Level 2) permite acceder a hojas de estilo
 - En HEAD
 - En ficheros aparte
- Como era de esperar, **Explorer pre-IE9** usa un API similar pero incompatible
 - Ver http://www.quirksmode.org/dom/w3c_css.html

Hojas y reglas de estilo

- document tiene un array de hojas de estilo
 - document.styleSheets (compatible IE-estándar)
- Cada hoja tiene un array de reglas
 - cssRules (estándar)
 - rules (Explorer)

```
//obtener la primera regla de la primera hoja  
if (document.styleSheets[0].cssRules)  
    regla = document.styleSheets[0].cssRules[0]  
else  
    regla = document.styleSheets[0].rules[0]
```

- **style** (equivalente al DOM 1)

```
document.styleSheets[0].cssRules[0].style.color = "red"
```

- **selectorText**: selector (como una cadena). Solo lectura (excepto Opera)
- **cssText**: texto (como una cadena)
 - Se puede obtener el texto de toda la hoja (OK en IE), de una regla (OK en el resto) o de las propiedades de una regla (OK en todos)

```
//propiedades de la 1ª regla de la 1ª hoja  
document.styleSheets[0].cssRules[0].style.cssText
```

● Insertar

- `insertRule(textoRegla, pos)` (Estándar)
- `addRule(selector, textoRegla)` (Explorer)

```
document.styleSheets[0].insertRule("H1 {color:red}", 0)
```

● Eliminar

- `deleteRule(pos)` (Estándar)
- `removeRule(pos)` (Explorer)

```
if (document.styleSheets[0].deleteRule)
    //estándar
    document.styleSheets[0].deleteRule(0)
else
    //IE8 y anteriores
    document.styleSheets[0].removeRule(0)
```