



CFGS DAW

Mòdul 6: Desenvolupament web en entorn
client

INS Joan d'Àustria

DOM Niveles 2 y 3



Extraído de Wikipedia (http://es.wikipedia.org/wiki/Document_Object_Model)

La primera vez que el DOM se utilizó, fue con el navegador [Netscape Navigator](#) en su versión 2.0. Este DOM se conoce también como «modelo básico», o «**DOM nivel 0**». [Internet Explorer](#) 3.0 fue el primer navegador de [Microsoft](#) que utilizó este nivel. Netscape 3.0 empezó a utilizar *rollovers*. Microsoft empezó a usar *rollovers* en [Internet Explorer](#) 4.0. Netscape 4.0 agregó la capacidad de detectar eventos ocurridos en el ratón y el teclado. Una característica de este navegador fue el uso de capas. Sin embargo, esta capacidad se ha eliminado en los navegadores creados posteriormente.

En [Internet Explorer](#) 4.0 todos los *elementos* de una página web se empezaron a considerar *objetos* computacionales con la capacidad de ser modificados. Debido a las diferencias en estos navegadores, el [World Wide Web Consortium](#) emitió una especificación denominada «**DOM nivel 1**» en el mes de octubre de 1998 en la cual se consideraron las características y manipulación de todos los elementos existentes en los archivos [HTML](#) y [XML](#). En noviembre del año 2000 se emitió la especificación del «**DOM nivel 2**». En esta especificación se incluyó la manipulación de eventos en el navegador, la capacidad de interacción con [CSS](#), y la manipulación de partes del texto en las [páginas de la web](#). «**DOM nivel 3**» se emitió en abril de 2004; utiliza la [definición de tipo de documento](#) (DTD) y la validación de documentos.

DOM Niveles 2 y 3



ESTILOS:

En HTML se pueden definir estilos de tres formas diferentes:

- a) Mediante ficheros externos con el elemento `<link>`
- b) Mediante estilos internos con el elemento `<style>`
- c) Mediante atributos específicos de cada elemento

DOM Niveles 2 y 3



ESTILOS:

Todo elemento HTML que soporte los atributos de estilo, también dispone de esta propiedad en javascript.

El objeto Style contiene toda la información de estilos de un elemento especificada mediante los atributos, pero no de los estilos heredados ni tampoco de los introducidos por las vias a) o b) anteriores

DOM Niveles 2 y 3



ESTILOS:

Las propiedades de este objeto Style se diferencian de las CSS en cuanto que CSS utiliza guiones para separar palabras y en javascript deberemos utilizar el estilo Camel. Así por ejemplo:

CSS	Javascript
background-image	style.backgroundImage
Color	style.color
font-family	style.fontFamily
background-color	style.backgroundColor

DOM Niveles 2 y 3



ESTILOS:

La única propiedad que modifica el nombre es la propiedad `css float`, pues en javascript tiene otro significado.

Esta propiedad `float` se traduce en `cssFloat` para los navegadores actuales y en `styleFloat` para IE 8 y anteriores.

DOM Niveles 2 y 3



ESTILOS:

```
var myDiv = document.getElementById("myDiv");  
  
myDiv.style.backgroundColor = "red";  
  
myDiv.style.width = "100px";  
myDiv.style.height = "200px";  
  
myDiv.style.border = "1px solid black";
```

DOM Niveles 2 y 3



ESTILOS:

Con DOM Level 2 también se proporcionan diferentes propiedades y métodos que nos dan acceso al contenido del estilo del elemento. Entre estas destacan:

- `cssText`: string que contiene todo el estilo del elemento (tal y como se ha descrito antes)

```
myDiv.style.cssText = "width: 25px; height: 100px; background-color: green";
```


DOM Niveles 2 y 3



ESTILOS:

- length: Número de propiedades css aplicadas al elemento
- parentRule: objeto cssRule que representa la información css
- getPropertyValue: recibe el nombre de una propiedad y retorna el valor de esta
- setProperty: recibe el nombre de una propiedad, el valor y un tercer parámetro que puede ser "important" o el string vacío y la asigna al elemento
- removeProperty: recibe el nombre de una propiedad y la elimina del estilo del elemento.

```
myDiv.style.removeProperty("border");
```

DOM Niveles 2 y 3



ESTILOS CALCULADOS:

El objeto style ofrece información sobre los atributos de estilo de un elemento, pero no sobre los estilos heredados que afectan también al elemento.

En Dom level 2 se añade el método `getComputedStyle` al `document.defaultView`

Estas propiedades son de solo lectura y no podremos modificarlas.

Versiones antiguas de IE no soportan `getComputedStyle`, aunque incorporan `currentStyle`.

<http://ie.microsoft.com/testdrive/HTML5/getComputedStyle/>

DOM Niveles 2 y 3



ESTILOS CALCULADOS:

```
<!DOCTYPE html>
<html>
<head>
<title>Ejemplo</title>
<style type="text/css">
#myDiv {
    background-color: blue;
    width: 100px;
    height: 200px;
}
</style>
</head>
<body>
<div id="myDiv" style="background-color:
red; border: 1px solid black"></div>
</body>
</html>
```

```
var myDiv = document.getElementById("myDiv");
var computedStyle =
document.defaultView.getComputedStyle(myDiv, null);

alert(computedStyle.backgroundColor); //"red"
alert(computedStyle.width); //"100px"
alert(computedStyle.height); //"200px"
alert(computedStyle.border); //
```

Los estilos se mostrarán diferentes en función del navegador. Algunos mostraran los colores en formato rgb, otros como opera en formato hexadecimal...

Los estilos shorthand no se mostraran, pero podemos acceder a sus diferentes partes. Así, por ejemplo no se mostrará ningún valor para border, pero podemos acceder a borderLeftWidth, borderLeftColor, borderLeftStyle, etc