

DOM

Tema 6-Continuación
Style

Estilos

- La propiedad `style` de los elementos permite acceder a los distintos estilos **en línea de un nodo**.
- Como los estilos en línea tienen mayor prioridad que el resto dentro de CSS, permite cambiar el estilo de los elementos.

Estilos

- **style** permite acceder a todos los atributos de estilo definidos en CSS, pero solo en línea , no a los valores originales.

nodo.style.nombreAtributoCSS

El nombre de los atributos varía ligeramente respecto a CSS.

- En los atributos CSS con guiones, los guiones desaparecen.
- Cuando hay palabras compuestas, la primera va en minúsculas y las siguientes en mayúsculas.
- Por ejemplo, la propiedad color de CSS llamaría igual que la propiedad color del objeto style de DOM, la propiedad background-color de CSS se llamaría backgroundColor en DOM o la propiedad border-top-color de CSS se llamaría borderTopColor en DOM.
- La especificación de DOM del W3C da una correspondencia de todas las propiedades CSS con DOM en www.w3.org/TR/DOMLevel2_css CSS CSS2Properties

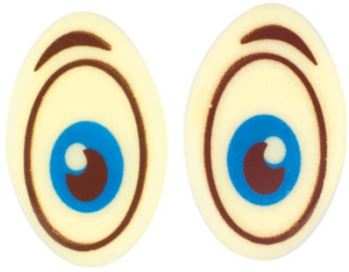
Estilos

- El objeto style es de sólo lectura por lo que no se puede modificar directamente.
- Los estilos *no* pueden establecerse asignando una cadena a la propiedad (solo lectura) style, como en `elt.style = "color: blue;"`.
- Esto es porque el atributo style devuelve un objeto del tipo CSSStyleDeclaration. En su lugar, puedes establecer las propiedades como:

```
elt.style.color = "blue";
```

Para modificar un estilo en línea hay que utilizar las propiedades del objeto style (los nombres de los atributos del estilo) que son de lectura escritura.

```
Var nodo=document.getElementById("ultimoParrafo");  
nodo.style.color = "blue"
```



- Únicamente muestra el valor de las propiedades que se establecieron utilizando el DOM. No muestra el valor de las propiedades que fueron establecidos utilizando CSS.
- Los cambios realizados no van a ser permanentes, el documento va a regresar a su estado original cuando se vuelva a cargar el documento.
- Si se utiliza un selector que devuelve muchos nodos (class, tag, name) se tiene que especificar el nodo del cual se quiere obtener el estilo.

Ejemplo: Cambiar una única propiedad css

```
<p id="intro">Supongamos que, por algún motivo,...</p>
```

Quiero cambiar el texto a otro color :

```
var intro = document.getElementById('intro');  
intro.style.backgroundColor = '#FF00FF';
```

Una vez ejecutado el html resultante quedaría :

```
<p id="intro" style="background-color: rgb(255, 0, 255);"> Supongamos que, por algún motivo,...  
</p>
```

Si queremos eliminarlo:

```
intro.style.backgroundColor = null;
```

Asignar varias propiedades

- La propiedad `style` contiene también el atributo `cssText`, que permite asignar varias propiedades CSS al mismo tiempo, asignándole una cadena (pero, ojo, porque esto sobrescribirá cualquier otro estilo en línea preexistente):

```
intro.style.cssText = 'background-color: black;  
color: white;';
```

Manipular la(s) clase(s) CSS de un elemento HTML

Si se desea cambiar varios atributos de estilo...

- Crear una regla para una clase con todos los atributos que se desea modificar.
- Cambiar el atributo class del elemento mediante *nodo.className*.

Ejemplo

Está clase la tenemos en nuestra hoja de estilos externa:

```
.negativo {  
background-color: black;  
color: white;  
}
```

- Asignamos esa clase en código:

```
var intro = document.getElementById('intro');  
intro.className = 'negativo';
```

Resultado:

```
<p id="intro" class="negativo">Supongamos  
que, por algún motivo,...</p>
```

Probando

- Creamos en nuestra hoja de estilos dos clases , una la llamaremos normal y otra recuadro.

Se trata de que empleando esta última técnica cambiemos la clase de un determinado párrafo con la clase normal asignada y cuando me sitúe encima, con el cursor cambias a recuadro y cuando esté fuera del párrafo cambia a normal.

- Avance tema7 (usar los eventos **mouseout** y **mouseover**)

Alternar hojas de estilo externas

- Existe la posibilidad de sustituir toda una hoja de estilo, por ejemplo `estilo_moderno.css` por `estilo_retro.css`.
- Esto permitiría cambiar dinámicamente todo el diseño, o «tema»

Ejemplo

```
<link id="estilos" rel="stylesheet"  
      href="css/moderno.css">
```

```
document.getElementById('estilos').href =  
    'css/retro.css';
```