

Estilos dos elementos con jQuery

Si queremos modificar o estilo dun elemento, temos dúas opcións. Podemos engadir ou eliminar unha clase CSS, provocando que o resto dos elementos que usan esa clase tamén sexan modificados. Ou ben podemos traballar directamente sobre o DOM aplicando os estilos directamente.

Imos ver como se pode facer en jQuery.

Engadir ou eliminar nomes de clase

Para engadir clases a un elemento empregaremos o método **addClass()**. Como parámetro escribiremos o nome da clase ou clases, separadas por un espacio, que desexamos engadir ó elemento:

```
$("#p:last").addClass("seleccionado");
```

Ó derradeiro párrafo do documento engadímoslle a clase "seleccionado".

Para eliminar unha clase dun elemento faise co método **removeClass()**. Como parámetro escribimos o nome da clase ou clases, separadas por un espacio que desexamos eliminar do elemento.

```
$("#p:first").removeClass("fondoazul");
```

Ó primeiro párrafo do documento sacámoslle a clase "fondoazul".

O método chamado **toggleClass()**. O que fai este método e engadir a clase indicada si non está presente ou eliminar a clase indicada si está presente.

E por último o método **hasClass()** indica si o elemento seleccionado está empregando a clase indicada como parámetro.

[Máis información e exemplos.](#)

Consulta e modificación de estilos

No apartado anterior vimos como xestionar qué clases lle aplicamos a un elemento, etc.

En moitos casos interésanos modificar directamente un estilo sin ter que activar ou desactivar unha clase.

Para facer isto dispomos do método **css()** que funciona de xeito semellante ó método `attr()`.

css(name) - Obtén o valor do propiedade name CSS, asociada ó primeiro elemento do conxunto.

```
$("#table").css("background-color")
```

Devolverá o color de fondo da primeira táboa do documento.

css(propiedades) - Nos permite asignar unha determinada propiedade ou propiedades a tódolos elementos do conxunto.

```
$("#p").css({ "backgroundColor" : "yellow", "fontWeight" : "bolder" });
```

css(name,valor) - Asigna á propiedade CSS name o valor indicado.

```
$("#p").css("color", "red")
```

Despos dispomos de accesos rápidos que nos permiten consultar ou modifica-lo ancho e a altura. Son os métodos **width()** e **height()**.

Exemplo:

```
$("#div.unhaClase").width(500)
```

é idéntica a:

```
$("#div.unhaClase").css("width", "500px")
```

Outro exemplo:

```
function informar() {  
    $('#contido').html(  
        $('#asunto').width()+'x'+$('#asunto').height()  
    );  
}
```

A función informar asignará en #contido unha mensaxe html que imprime o ancho e a altura do obxecto con ID asunto.

Diferencias entre width, width(), height e height()

Agora que estamos falando de estilos e propiedades dos obxectos tedes que ter en conta o espazo de traballo no cal nos atopamos, jQuery (instrucións que comezan por \$) ou Javascript.

Por exemplo:

\$(this).width() e iso nos devolverá o ancho do obxecto actual.

No espazo de Javascript faríamos:

this.width e nos devolverá o ancho dese obxecto.

O mesmo ocorre coa propiedade de javascript **height** e o método de jQuery **height()**.

Resumindo é moi importante decatarse do entorno no cal nos atopamos para empregar unha función ou o seu correspondente método en jQuery.

Algo semellante ocurriría co método **html()** de jQuery, que ten como propiedade asociada en Javascript a propiedade **innerHTML**.

[Máis información sobre xestión do CSS con jQuery](#)

--Veiga (discusión) 22:56 26 ene 2015 (CET)