Manejo de objetos y eventos con librería JavaScript

Módulo: **Desarrollo Web en entorno cliente**

CFGS Desarrollo de Aplicaciones Web

¿$ vs $()?

- **$.fn**: **métodos del espacio de nombres**, son métodos del objeto jQuery y trabajan con selecciones.

$(“h1”).text();

- **$**: métodos que no trabajan con selecciones y no se les pasa ningún argumento. A estos se les denomina **métodos del núcleo** de jQuery.

$.each

- Existen métodos del objeto y del núcleo que poseen el mismo nombre y hay que tener cuidado de cómo usarlos. Por ejemplo:

$.each y $.fn.each

Métodos del núcleo

$.trim: quita los espacios en blanco del principio y final de una cadena.

```
    $.trim('    varios espacios en blanco   ');
```

$.each: itera sobre objetos y arrays.

```
$.each([ 'foo', 'bar', 'baz' ], function(idx, val) {
        console.log('elemento ' + idx + 'es ' + val);
    });
```

$.inArray: devuelve el índice de un valor en un array o -1 si no lo encuentra.

```
var myArray = [ 1, 2, 3, 5 ];
 if ($.inArray(4, myArray) !== -1) {
    console.log('valor encontrado');
 }
```

$.proxy: establece el significado de this incluido dentro de la función como el segundo argumento.

```
var myFunction = function() { console.log(this); };
var myObject = { foo : 'bar' };

myFunction(); // devuelve el objeto window

var myProxyFunction = $.proxy(myFunction, myObject);
myProxyFunction(); // devuelve el objeto myObject
```

Comprobando tipos

1. Tipos primitivos: Utilizar typeof de JS

   Typeof miVar;

2. Valores nulos: Utilizar el operador de igualdad estricta

   miVar===null;

3. Tipos no primitivos: Utilizar jQuery

```
jQuery.isFunction(miVar);
jQuery.isPlainObject(miVar);
jQuery.isArray(miVar);
jQuery.isNumeric(16);
```

Detectar características y navegador

$.support: detecta las características del navegador

```
$(document).ready(function(){
    $("p").html("This browser can create XMLHttpRequest object: " +
jQuery.support.ajax);
});
```

Características:

- ajax
- boxModel
- changeBubbles
- checkClone
- checkOn
- cors
- cssFloat
- hrefNormalized
- htmlSerialize
- leadingWhitespace
- noCloneChecked
- noCloneEvent
- opacity
- optDisabled
- optSelected
- scriptEval()
- style
- submitBubbles
- tbody

Nota: $.browser: detecta el tipo de navegador y su versión. Esta característica está en desuso y no se recomienda su uso. Usar en su lugar el objeto $.support.

Evitar conflictos con otras bibliotecas

Cuando usamos otros framework de JS que usan la variable $ puede ocurrir que se presenten conflictos. Para evitar estos errores podemos:

1. **Poner jQuery en modo "no conflicto". ¿Cómo?** Inmediatamente después de que jQuery se cargue en la página y antes del código que se va a ejecutar.

```
<script src="prototype.js"></script>  // prototype usa $
<script src="jquery.js"></script>     // se carga jquery
<script> $.noConflict();</script>   // modo "no conflicto"
```

2. Se puede usar jQuery o declarar una variable que haga referencia a jQuery.

```
$.noConlict();
jQuery("h1").text();
```

```
var jq = $.noConflict();
jq("h1").text();
```

3. Si tenemos un bloque de código y no queremos cambiarlo podemos pasar como parámetro $ al método ready.

```
$.noConlict();
jQuery(document).ready(function($) {
    $("h1").text();
});
```