

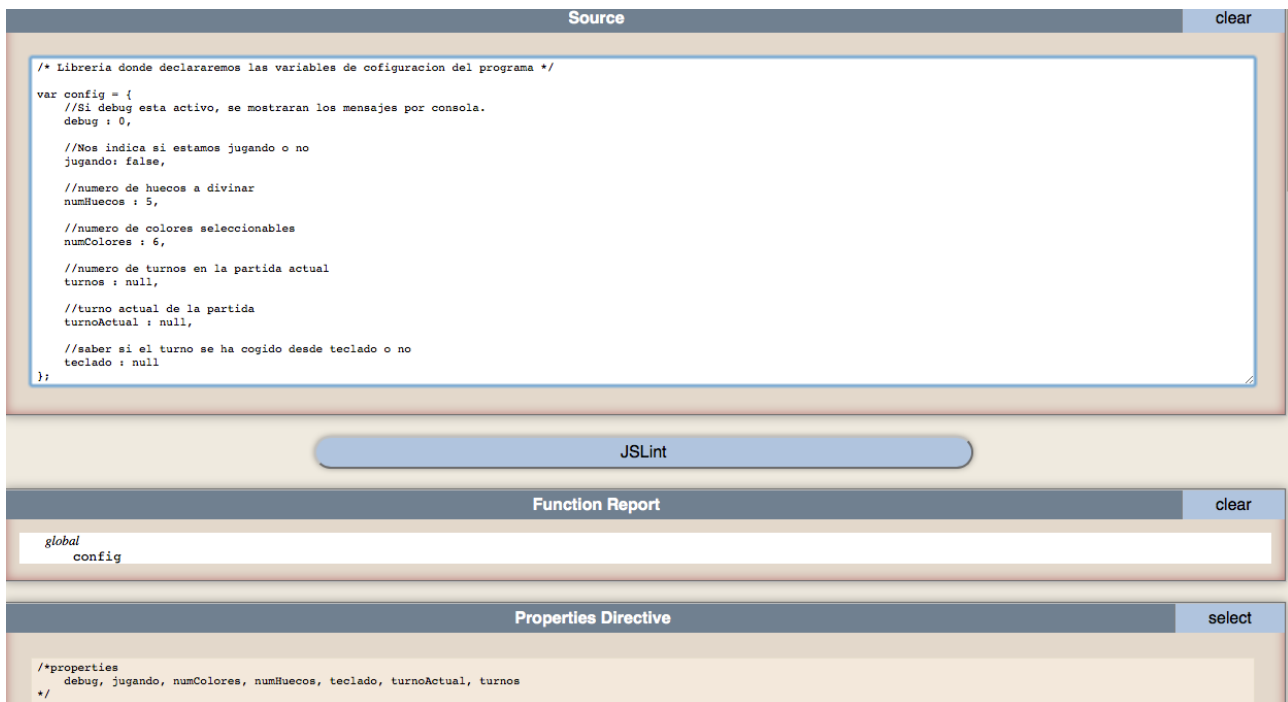
# JSLINT & JSHINT

Para aprender a utilizar las herramientas para validar código javascript he utilizado dos archivos de mi MasterMind: Config.js y Cookies.js.

## Archivo Config.js

En este caso solo pongo las capturas una vez corregido el código, ya que es un archivo con solo variables de configuración y había errores de sintaxis, casi todo espacios que faltaban.

Aquí vemos el archivo Config.js pasando el corrector de JSLint:



JSLint de Config.js

En JSLint nos indica que tenemos una variable global, config, y que tenemos 7 propiedades, que son las variables de configuración de mi MasterMind.

A continuación veremos el archivo Config.js validado en JSHint:

```

1 // Hello.
2 //
3 // This is JSHint, a tool that helps to detect errors and potential
4 // problems in your JavaScript code.
5 //
6 // To start, simply enter some JavaScript anywhere on this page. Your
7 // report will appear on the right side.
8 //
9 // Additionally, you can toggle specific options in the Configure
10 // menu.
11
12 /* Libreria donde declararemos las variables de configuracion del programa */
13
14 var config = {
15     //Si debug esta activo, se mostraran los mensajes por consola.
16     debug : 0,
17
18     //Nos indica si estamos jugando o no
19     jugando: false,
20
21     //numero de huecos adivinar
22     numHuecos : 5,
23
24     //numero de colores seleccionables
25     numColores : 6,
26
27     //numero de turnos en la partida actual
28     turnos : null,
29
30     //turno actual de la partida
31     turnoActual : null,
32
33     //saber si el turno se ha cogido desde teclado o no
34     teclado : null
35 };
36

```

CONFIGURE

One unused variable

14 config

JSHint de Config.js

Está todo correcto, solo nos indica que no se utiliza la variable config.

## Archivo Cookies.js

Aquí veremos una captura del JSLint del archivo Cookies.js que me daba muchos errores, pero casi todos de sintaxis por tema espacios:

Source

clear

```

var cookies = {
    //Funcion que crea una cookie dado un nombre, un valor y cuando expira
    setCookie : function(cname,cvalue,exdays)
    {
        var d = new Date();
        d.setTime(d.getTime()+(exdays*24*60*60*1000));
        var expires = "expires="+d.toGMTString();
        document.cookie = cname + "=" + cvalue + ";" + expires;
    },
    //funcion que coge una cookie con un nombre
    getCookie : function(cname)
    {
        var name = cname + "=";
        var ca = document.cookie.split(';');
        for(var i=0; i<ca.length; i++)
        {
            var c = ca[i].trim();
            if (c.indexOf(name)==0) return c.substring(name.length,c.length);
        }
        return "";
    },
};

```

JSLint

Errors

clear

Expected exactly one space between 'function' and '('.	line 4 character 25
setCookie : function(cname,cvalue,exdays)	
Missing space between ',' and 'cvalue'.	line 4 character 32
setCookie : function(cname,cvalue,exdays)	
Missing space between ',' and 'exdays'.	line 4 character 39
setCookie : function(cname,cvalue,exdays)	

JSHint de Cookies.js

Una vez corregidos la mayoría de los errores, solo quedan 4, que son errores de no declarar las variables “globales” de otros archivos como utils y masterUI o document:

Source

clear

```
var cookies = {  
  //Funcion que crea una cookie dado un nombre, un valor y cuando expira  
  setCookie : function (cname, cvalue, exdays) {  
    "use strict";  
    var d, expires;  
    d = new Date();  
    d.setTime(d.getTime() + (exdays * 24 * 60 * 60 * 1000));  
    expires = "expires=" + d.toGMTString();  
    document.cookie = cname + "=" + cvalue + "; " + expires;  
  },  
  
  //funcion que coge una cookie con un nombre  
  getCookie : function (cname) {  
    "use strict";  
    var name, ca, i, c;  
    name = cname + "=";  
    ca = document.cookie.split(';');  
    for (i = 0; i < ca.length; i = i + 1) {  
      c = ca[i].trim();  
      if (c.indexOf(name) === 0) {  
        return c.substring(name.length, c.length);  
      }  
    }  
  }  
}
```

JSLint

Errors

clear

'document' was used before it was defined.  
line 10 character 9  
document.cookie = cname + "=" + cvalue + "; " + expires;  
  
'document' was used before it was defined.  
line 18 character 14  
ca = document.cookie.split(';');  
  
'utils' was used before it was defined.  
line 45 character 13  
utils.alert("no hay cookie");  
  
'masterUI' was used before it was defined.  
line 49 character 13  
masterUI.ponerValorSlider(valor);

Function Report

clear

global  
Date, cookies

'setCookie'  
parameter  
cname, cvalue, exdays  
variable  
d, expires  
global  
Date  
line 4

'getCookie'  
parameter  
cname  
variable  
c, ca, i, name  
line 14

JSHint de Cookies.js

Ahora pasaremos ver el archivo Cookies.js validado en JSLint:

```

1 // Hello.
2 //
3 // This is JSHint, a tool that helps to detect errors and potential
4 // problems in your JavaScript code.
5 //
6 // To start, simply enter some JavaScript anywhere on this page. Your
7 // report will appear on the right side.
8 //
9 // Additionally, you can toggle specific options in the Configure
10 // menu.
11
12 /* Libreria donde declararemos las variables de configuracion del programa */
13
14 var cookies = {
15
16     //Funcion que crea una cookie dado un nombre, un valor y cuando expira
17     setCookie : function (cname, cvalue, exdays) {
18         "use strict";
19         var d, expires;
20         d = new Date();
21         d.setTime(d.getTime() + (exdays * 24 * 60 * 60 * 1000));
22         expires = "expires=" + d.toGMTString();
23         document.cookie = cname + "=" + cvalue + "; " + expires;
24     },
25
26     //funcion que coge una cookie con un nombre
27     getCookie : function (cname) {
28         "use strict";
29         var name, ca, i, c;
30         name = cname + "=";
31         ca = document.cookie.split(';');
32         for (i = 0; i < ca.length; i = i + 1) {
33             c = ca[i].trim();
34             if (c.indexOf(name) === 0) {
35                 return c.substring(name.length, c.length);
36             }
37         }
38         return "";
39     },
40
41     //funcion que comprueba si existe una cookie
42     checkCookie : function (cname) {
43         "use strict";
44         var cookie = cookies.getCookie(cname);
45         if (cookie !== "") {
46             //la cookie existe
47             return true;
48         }
49         //la cookie NO existe
50         return false;
51     }
52 }

```

CONFIGURE

JSLint nos hace un resumen de las funciones de Cookies.js. Sale todo correctamente, excepto que nos dice que no hemos declarado ni `utils` ni `masterUI`, como nos decía JSHint.