

Apartado 1. Verificación de valores introducidos.

Campo COUNTRY:

```
48 function comparar(input) {  
49     var op = document.getElementById("country");  
50     for (var i = 0; i < op.options.length; i++)  
51     {  
52         if ( op.options[i].value.toUpperCase() == input.toUpperCase())  
53             return true;  
54     }  
55     alert('País erroneo');  
56     return false;  
57 }
```

A partir del método onblur, comparo el input introducido por el usuario con la lista de países que tengo en mi datalist. La validación se hace en mayúsculas (toUpperCase) para poder comparar correctamente evitando así algún error debido a que el usuario haya introducido el país de una forma no esperada. En caso de que el input no coincida con ningún elemento de la lista salta un mensaje de error.

Verificación de la temperatura:

```
61 function Temperatura() {  
62     var elx = parseInt(document.getElementById("a").value);  
63     var ely = parseInt(document.getElementById("b").value);  
64     var elz = document.getElementById("c");  
65  
66     elz.value = elx - ely;  
67     //document.getElementById("c").innerHTML.value = elx.value + ely.value;  
68     if(elz.value > 26 ||  
69        elz.value < 22)  
70     {  
71         elz.style.color = 'red';  
72         alert('Valor de la temperatura fuera de rango, debe estar entre 22 y 26 grados')  
73     }  
74     else  
75     {  
76         elz.style.color = '#2F4F4F';  
77     }  
78 }  
79 }
```

La función es llamada a partir del método onchange de ambos selectores de temperatura utilizando su conversión correspondiente con "parseInt". Se indica la diferencia entre ambos y en caso de que el valor no se encuentre dentro del rango especificado salta un mensaje de error y el color de la diferencia se cambia a rojo mediante "style.color".

Validar contraseña:

Para la validación de la contraseña he utilizado la función `functionDatos()` de la práctica anterior, que responde al método `onclick` del botón correspondiente a la página de configuración.

Mediante una expresión regular que dicta las condiciones de mi contraseña, verifico (mediante el método `match()`) si el input introducido es válido. A parte del número de caracteres, he implementado que la expresión regular verifique si la contraseña tiene al menos una letra minúscula, una letra mayúscula y un número. Si obtengo todas las ocurrencias de la expresión procedo a verificar la longitud de la contraseña (mediante la propiedad `length`) y guardo en un string el nivel de seguridad según su tamaño para luego mostrarlo por pantalla.

En caso de que la contraseña sea inválida, compruebo que condiciones han fallado concretamente utilizando una expresión regular para cada condición. Al encontrar el fallo guardo en un string la condición que falta, en caso contrario el string se queda en blanco, de forma que al final pueda mostrar por pantalla todos los strings concatenados para que solo se vean los errores.

```
2  function functionDatos(input) {
3      //alert('Esta funcion no esta implementada')
4      var passw = /^(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{6,}$/;
5
6      if(input.value.match(passw))
7      {
8          //alert('a')
9          var seg = "";
10         if(input.value.length < 7)
11             seg = "acceptable";
12         else if(input.value.length >= 7 && input.value.length <= 12)
13             seg = "media";
14         else if(input.value.length > 12)
15             seg = "alta";
16         alert('Contrase\u00F1a v\u00E1lida, nivel de seguridad: ' + seg);
17         return true;
18     }
19     else
20     {
21         var minuscula = /[a-z]/g;
22         var mayuscula = /[A-Z]/g;
23         var num = /[0-9]/g;
24         var letra = "";
25         var Letra = "";
26         var numero = "";
27         var caracteres = "";
28         if(!input.value.match(minuscula))
29             //alert('Debe contener una letra minuscula' + letra)
30             letra = " una letra min\u00FAscula,";
31         if(!input.value.match(mayuscula))
32             //alert('Debe contener una letra mayuscula')
33             Letra = " una letra may\u00FAscula,";
34         if(!input.value.match(num))
35             //alert('Debe contener al menos un numero')
36             numero = " un n\u00FAmero,";
37         if(input.value.length < 6)
38             //alert('Debe contener al menos 6 caracteres')
39             caracteres = " 6 caracteres.";
40         alert('Le falta al menos:' + letra + Letra + numero + caracteres);
41         //alert('La contrasena debe contener al menos 6 caracteres, un numero, u
42         return false;
43     }
44 }
```

```
80 function seguridad(){
81     var elx = document.getElementById("pwd");
82     var passw = /^(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{6,}$/;
83
84     if(elx.value.match(passw))
85         elx.style.color = 'green';
86     else
87         elx.style.color = 'red';
88 }
```

En este apartado haré uso de nuevo de las expresiones regulares, ahora para determinar si el campo input de las coordenadas está recibiendo coordenadas en formato decimal. Dicha expresión reconoce si los datos de latitud y longitud son válidos, si se utiliza el “punto” como separador decimal y la “coma” como separador entre latitud y longitud, y también si los datos se han proporcionado juntos. En caso afirmativo se hace visible un enlace que tenía escondido en la página de configuración y se modifica de forma que se “peguen” las coordenadas proporcionadas al final de un enlace a Google maps. Si la expresión no se cumple oculto el enlace, ya que las coordenadas no están bien introducidas.

```

90 function maps(){
91 //var valido = /^[^d]+,[^d]*$/;
92 //var valido = /\d{2}\.d{6}/gm;
93 //var valido = /^[^d+,\d+$/; //este solo acepta enteros, enteros
94 var valido = /^[+]?([1-8]?\d(\.d+)?|90(\.0+)?|\s*[-+]??(180(\.0+)?|((1[0-7]\d)|([1-9]?\d))(\.d+)?))$/;
95 var a = document.getElementById('url');
96 var elx = document.getElementById("gps1");
97 var url = "http://maps.google.com/maps?q=";
98 if(elx.value.match(valido))
99 {
100     alert('Coordenadas v\u00E9lidas, creando enlace...')
101     a.href = url + elx.value;
102     if(a.style.display == "none")
103         a.style.display = "block";
104 }
105 else
106 {
107     alert('Coordenadas inv\u00E9lidas')
108     if(a.style.display == "block")
109         a.style.display = "none";
110 }
111 }

```

El procedimiento para este apartado es similar a los anteriores. He buscado una expresión para contraseñas lo más genérica posible y la comparo con el input mediante el método `match()`.

```
function validar() {  
    x = document.getElementById('b').value;  
    //var valido = /^[a-zA-Z0-9_!#$%&'*/+=/^\~]+@[a-zA-Z0-9-]+\.(?=[a-zA-Z0-9-])+$;/;  
    var valido = /^(("[\w-\.\@"]+)|([\w-\.@]+))@([\w-]+\.[\w-]+\.[\w-]+)$$/i;  
    if(x.match(valido))  
    {  
        alert("Dirección de correo válida");  
        return true;  
    }  
    else  
    {  
        alert("Error, introduzca una dirección de correo válida");  
        return false;  
    }  
}
```