

U4. Eventos y formularios

**Parte II. Interacción con formularios
y expresiones regulares**

Desarrollo Web en Entorno Cliente

Contenidos

- Eventos
- Validación de formularios
- Expresiones regulares
- Objeto RegExp

Eventos de formulario

- Eventos de página relacionados con **Formularios**
 - **Reset.** Este evento se produce cuando pulsamos sobre un botón de tipo reset. El manejador es `onreset`.
 - **Focus.** Este evento se produce cuando un elemento obtiene el foco. El manejador es `onfocus`.
 - **Blur.** Este evento se produce cuando un elemento pierde el foco. El manejador es `onblur`.
 - **Select.** Se acciona cuando seleccionamos texto de los cuadros de textos `<input>` y `<textarea>`. El manejador es `onselect`.
 - **Change.** Este evento se produce cuando los cuadros de texto `<input>` y `<textarea>` pierden el foco y el contenido que tenían ha variado. También se producen cuando un elemento `<select>` cambia de valor. El manejador es `onchange`.
 - **Submit.** Este evento se produce cuando pulsamos sobre un botón de tipo submit. El manejador es `onsubmit`.

Validación de formularios

- El usuario puede cometer errores al rellenar un formulario.
- Si por ejemplo se espera un código postal y se introduce el nombre de una ciudad, se producirá un error.
- Para controlar estas situaciones se pueden usar las validaciones.

Validación de formularios

- Validar un campo como obligatorio:

```
<script type="text/javascript">
  function validacion() {
    valor = document.getElementById("campo").value;

    if( valor == null || valor.length == 0 ){
      alert("El campo no puede ser vacío");
      return false;
    }
    return true;
  }
</script>
```

Validación de formularios

- Validar un campo de texto como numérico:

```
<script type="text/javascript">
  function validaNum() {
    valor = document.getElementById("telefono").value;

    if( isNaN(valor) ) {
      alert("El campo tiene que ser numérico");
      return false;
    }
    return true;
  }
</script>
```

Validación de formularios

- Validar si una fecha es correcta:

```
<script type="text/javascript">
  function validaFecha() {
    var dia = document.getElementById("dia").value;
    var mes = document.getElementById("mes").value;
    var ano = document.getElementById("ano").value;
    fecha = new Date(ano, mes, dia);
    if( !isNaN(fecha) ) {
      return false;
    }
    return true;
  }
</script>
```

Validación de formularios

- Validar un checkbox:

```
<script type="text/javascript">
function validaCheck() {
    elemento = document.getElementById("campoCondiciones");

    if( !elemento.checked ) {
        return false;
    }
    return true;
}
</script>
```


Expresiones regulares

- Las expresiones regulares describen un conjunto de elementos que siguen un patrón.
- Un ejemplo podría ser todas las palabras que comienzan por la letra 'a' minúscula.
- JavaScript implementa expresiones regulares y facilita las comprobaciones de ciertos datos que deben seguir una estructura concreta.

Expresiones regulares

- Caracteres especiales (1):
 - **^ Principio de entrada o línea.** Este carácter indica que las cadenas deberán comenzar por el siguiente carácter. Si este fuera una “a” minúscula como indicamos en el punto anterior la expresión regular sería `^a`.
 - **\$ Fin de entrada o línea.** Indica que la cadena debe terminar por el elemento precedido al dólar.

Expresiones regulares

- Caracteres especiales (2):
 - *** El carácter anterior 0 o más veces.** El asterisco indica que el carácter anterior se puede repetir en la cadena 0 o más veces.
 - **+ El carácter anterior 1 o más veces.** El símbolo más indica que el carácter anterior se puede repetir en la cadena una o más veces.
 - **? El carácter anterior una vez como máximo.** El símbolo interrogación indica que el carácter anterior se puede repetir en la cadena cero o una vez.

Expresiones regulares

- Caracteres especiales (3):
 - **.** **Cualquier carácter individual.** El símbolo punto indica que puede haber cualquier carácter individual salvo el de salto de línea.
 - **x|y** **x ó y:** La barra vertical indica que puede ser el carácter x o el y.
 - **{n}** **n veces el carácter anterior.** El carácter anterior a las llaves tiene que aparecer exactamente n veces.

Expresiones regulares

- Caracteres especiales (4):
 - **{n,m}** Entre n y m veces el carácter anterior. El carácter anterior a las llaves tiene que aparecer como mínimo n y como máximo m veces.
 - **[abc]** Cualquier carácter de los corchetes. En la cadena puede aparecer cualquier carácter que este incluido en los corchetes.

Expresiones regulares

- Caracteres especiales (5):
 - **[^abc]** Un carácter que no esté en los corchetes.
En la cadena pueden aparecer todos los caracteres que no estén incluidos en los corchetes.
 - **\b Fin de palabra.** Este símbolo indica que tiene que haber un fin de palabra o retorno de carro.
 - **\B No fin de palabra.** El símbolo \B indica cualquiera que no sea un límite de palabra.

Expresiones regulares

- Caracteres especiales (6):
 - **\d** **Cualquier carácter dígito**. Este símbolo indica que puede haber cualquier carácter numérico, de 0 a 9.
 - **\D** **Carácter que no es dígito**. Este símbolo indica que puede haber cualquier carácter siempre que no sea numérico.
 - **\f** **Salto de página**. Este símbolo indica que tiene que haber un salto de página.

Expresiones regulares

- Caracteres especiales (7):
 - **\n Salto de línea.** Este símbolo indica que tiene que haber un salto de línea.
 - **\r Retorno de carro.** Este símbolo indica que tiene que haber un retorno de carro.
 - **\s Cualquier espacio en blanco.** Este símbolo indica que tiene que haber un carácter individual de espacio en blanco: espacios, tabulaciones, saltos de página o saltos de línea.

Expresiones regulares

- Caracteres especiales (8):
 - **\S** **Carácter que no sea blanco.** Este símbolo indica que tiene que haber cualquier carácter individual que no sea un espacio en blanco.
 - **\t** **Tabulación.** Este símbolo indica que tiene que haber cualquier tabulación.
 - **\w** **Carácter alfanumérico.** Este símbolo indica que puede haber cualquier carácter alfanumérico.
 - **\W** **Carácter que no sea alfanumérico.** Este símbolo indica que puede haber cualquier carácter que no sea alfanumérico.

Expresiones regulares

- Validar un formulario con expresiones regulares:
 - Combinando las anteriores expresiones se puede abordar una infinidad de patrones para validar datos en los formularios.
 - Se pueden validar por ejemplo campos como:
 - Correo electrónico.
 - Teléfono.
 - Código postal.
 - DNI.
 - Etc.

Expresiones regulares

- Validar una dirección de correo electrónico:

```
<script type="text/javascript">
  function validaEmail() {
    valor = document.getElementById("campo").value;

    if(! (/^\w+([-+.']\w+)*@\w+([-.\]\w+)*\.\w+([-
      .]\w+)/.test(valor)) ) {
      return false;
    }
    return true;
  }
</script>
```

Expresiones regulares

- Validar un DNI:

```
<script type="text/javascript">
function validaDNI(){
    valor = document.getElementById("dni").value;
    var letras = ['T','R','W','A','G','M','Y','F','P','D','X',
        'B','N','J','Z','S','Q','V','H','L','C','K','E','T'];
    if( !(/^\d{8}[A-Z]$/.test(valor)) ){ return false; }
    if(valor.charAt(8) != letras[(valor.substring(0, 8))%23])
    { return false; }
    return true;
}
</script>
```

Expresiones regulares

- Validar un número de teléfono:

```
function validaTelefono() {  
    valor = document.getElementById("telefono").value;  
  
    if( !(/^\d{9}$/.test(valor)) ) {  
        return false;  
    }  
    return true;  
}
```

Objeto RegExp

- Una expresión regular puede crearse de cualquiera de las siguientes dos maneras:
 - Utilizando una representación literal de la expresión:

```
var re = /ab+c/;
```

 - La representación literal compila la expresión regular una vez que el script ha terminado de cargar.
 - Constructor del objeto RegExp:

```
var re = new RegExp("ab+c");
```

 - El uso del constructor compila la expresión regular en tiempo de ejecución. Su uso es recomendable en aquellos escenarios en que el patrón de la expresión regular pueda cambiar durante la ejecución del script, o bien, se desconoce el patrón (cuando es suministrado por el usuario, por ejemplo).