CFGS DAW

Mòdul 6: Desenvolupament web en entorn client

INS Joan d'Àustria

Eventos

Obtener información de formularios

La mayoría de aspectos de javascript y formularios pasa por la consulta y modificación de los valores de los diferentes elementos.

a) Text y textarea: El valor se consulta y se modifica directamente sobre la propiedad value:

```
<input type="text" id="texto" />
var valor = document.getElementById("texto").value;

<textarea id="parrafo"></textarea>
var valor = document.getElementById("parrafo").value;
```

Eventos

Obtener información de formulario

b) Botones radio: Cuando tenemos un grupo de radiobuttons, lo importante suele ser saber cuál de ellos es el seleccionado. La propiedad checked sobre cada uno de ellos nos mostrará si está seleccionado o no

```
<input type="radio" value="si" name="pregunta" id="pregunta_si"/> SI
<input type="radio" value="no" name="pregunta" id="pregunta_no"/> NO
<input type="radio" value="nsnc" name="pregunta" id="pregunta_nsnc"/> NS/NC
```

```
var elementos = document.getElementsByName("pregunta");
for(var i=0; i<elementos.length; i++) {
alert(" Elemento: " + elementos[i].value + "\n Seleccionado: " + elementos[i].checked);
}
```

Eventos

Obtener información de formulario

c) Checkbox: El funcionamiento es similar a los radiobuttons, pero con la particularidad que los checkbox no son excluyentes entre si.

```
<input type="checkbox" value="condiciones" name="condiciones" id="condiciones"/>
He leído y acepto las condiciones
<input type="checkbox" value="privacidad" name="privacidad" id="privacidad"/> He
leído la política de privacidad
```

```
var elemento = document.getElementById("condiciones");
alert(" Elemento: " + elemento.value + "\n Seleccionado: " + elemento.checked);
elemento = document.getElementById("privacidad");
alert(" Elemento: " + elemento.value + "\n Seleccionado: " + elemento.checked);
```

Eventos

Obtener información de formularios

d) Select: Este es el elemento más complejo. Para conocer su valor, deberemos acceder a las siguientes propiedades:

- options: vector que contiene todas las opciones de la lista (así, la primera opción sería document.getElementById("id_de_la_lista").options[0])
- selectedindex: marca el índice de la opción seleccionada por el usuario sobre el array options anterior.

```
// Obtener la referencia a la lista
var lista = document.getElementById("opciones");
// Obtener el índice de la opción que se ha seleccionado
var indiceSeleccionado = lista.selectedIndex;
// Con el índice y el array "options", obtener la opción seleccionada
var opcionSeleccionada = lista.options[indiceSeleccionado];
// Obtener el valor y el texto de la opción seleccionada
var textoSeleccionado = opcionSeleccionada.text;
var valorSeleccionado = opcionSeleccionada.value;
```

Eventos

Utilidades sobre formularios

Con lo que hemos visto, además de validar los diferentes campos, podemos aplicar una serie de utilidades:

- ✓ Evitar envío duplicado: Hay muchos usuarios que tienden a hacer doble click sobre los botones, tal y como hacen con el sistema operativo. En estos casos la información se puede enviar duplicada o generar casos de error, especialmente en aquellos casos con conexión lenta o que la respuesta del servidor se demora.

```
<form id="formulario" action="#">
...
<input type="button" value="Enviar" onclick="this.disabled=true;
this.value='Enviando...'; this.form.submit()" />
</form>
```

Eventos

Utilidades sobre formularios

- ✓ Limitar número de carácteres de un textarea: Los elementos text disponen de la propiedad maxlength, pero los textarea no. Los eventos presentan una característica importante: **Si la función que los trata retorna el valor true, el evento sigue su curso normal, mientras que si retorna false, el evento se detiene**

```
<textarea onkeypress= "return true;"> </ textarea>
```

```
<textarea onkeypress= "return false;"> </ textarea>
```

Eventos

Utilidades sobre formularios

Así, podemos construir una función que se encargue de realizar esta validación

```
<textarea id="texto" onkeypress="return limita(100);"></textarea>

function limita(maximoCaracteres) {
        var elemento = document.getElementById("texto");
        if(elemento.value.length >= maximoCaracteres ) {
                return false;
        }
        else {
                return true;
        }
}
```

Eventos

Utilidades sobre formularios

- ✓ Validación de formularios: Ya hemos visto como realizar la validación de los diferentes elementos de un formulario. ¿Cómo podemos hacer para que el formulario se envíe (submit) si no contiene errores? Podemos aprovechar lo que acabamos de ver sobre return true o false en los eventos

```
<form action="#" onsubmit="return validacion()">
...
</form>
```

```
function validacion() {
if (condicion1) {
        return false; //error
}
...
// Si el script ha llegado a este punto, todas las condiciones
// se han cumplido, por lo que se devuelve el valor true
return true;
}
```

Eventos

Utilidades sobre formularios

- ✓ Realizar validaciones utilizando expresiones regulares: Ya hemos visto que los nuevos elementos HTML5 incorporan un patrón. Desde javascript se pueden crear y validar expresiones regulares para realizar la validación de los diferentes elementos HTML. Para ello utilizaremos la función test:

  expresion_regular.test(valor)

  Esta función valida que el valor pasado cumpla con la expresión regular sobre la que se aplica y retorna true o false

  La construcción de las expresiones regulares la podemos observar por ejemplo en: http://www.w3schools.com/jsref/jsref_obj_regexp.asp