UT 9 - FORMULARIOS Y CONTROLES (RESUMEN)

Propiedades y eventos especiales para formularios (<form>) y controles (<input>, <select> y otros).

Contenido

1. Propiedades y métodos de formularios

Navegación: formularios y elementos

Los formularios del documento son miembros de una colección especial `document.forms`.

También se llama "colección nombrada": se puede acceder por el nombre del formulario (propiedad `name`) y por un índice.

Propiedad	Descripción
`document.forms`	Colección nombrada de formularios del documento
`document.forms.name` *`formulario`*`.elements["name"]`	Puntero al formulario con ese "name"
`document.forms[0]`	Puntero al primer formulario del documento

Para cada formulario, se crea una "colección nombrada" de sus controles.

Si hay varios elementos con el mismo name, en lu

Propiedad	Descripción
`formulario``.elements`	Colección nombrada de controles del formulario
`formulario``.elements.name` *`formulario`*`.elements["name"]`	Puntero al control del formulario con ese "name"(*)
`formulario``.elements[0]`	Puntero al primer control del formulario (*)

(*) Si hay varios elementos con el mismo `name` (`checkbox`, `radio`), cada elemento de la colección `elements`, será a su vez una colección.

Fieldset como "subformularios"

Los elementos `<fieldset>` son como sub-formularios, ya que tienen una propiedad `elements` con todos los elementos que están dentro del formulario.

Notación corta: *formulario*.name

En lugar de *`formulario`*`.elements.name`, podemos escribir *`formulario`*`.name`.

Propiedad	Descripción
`formulario``.name` *`formulario`*`["name"]`	Puntero al control del formulario con ese "name"(*)

Referencia inversa: element.form

Todos los controles de un formulario tienen una propiedad `form`, que apunta al formulario.

Propiedad	Descripción
`control``.form`	Puntero al formulario al que pertenece ese control.

Elementos del formulario (controles)

<input> y <textarea>

Propiedad	Descripción
`input.value`	Cadena de texto con la información del `input`
`input.checked`	Boolean que indica si el `input` está seleccionado o no. Para `<input type="checkbox">` o `<input type="radio">`
`textarea.value`	Cadena de texto con la información del `textarea`. (*) No utilizar `innerHTML`, porque guarda el HTML que había inicialmente en la página, no su valor actual

<select> y <option>

Propiedad	Descripción
`select.options`	Colección de subelementos `<option>`
`select.value`	Valor del `<option>` seleccionado actualmente
`select.selectedIndex`	Número del `<option>` seleccionado actualmente.
`option.selected`	Boolean que indica si un `<option>` está seleccionado.
`option.index`	Número del `option` respecto a los demás en su `select`.
`option.text`	Contenido del `option`.

Cómo asignar un valor a un select: tres métodos

- select.value = value del option que queremos seleccionar
- select.selectedIndex = índice del option que queremos seleccionar
- option.selected = true

(*) Si el `<select>` tiene activado el atributo multiple, permite seleccionar varios `<option>`.
En ese caso, habrá que utilizar el último método.

new Option

Permite crear fácilmente elementos <option>.

```
option = new Option(text, value, defaultSelected, selected)
```

Ejemplo:

```
let option = new Option("Text", "value");

// crea <option value="value">Text</option>
```

2. Foco: focus / blur

Un elemento se puede "enfocar" de varios modos: haciendo click sobre el con Tab, etc.

Cuando un elemento se enfoca, la entrada desde teclado (o desde un paste) irá dirigida a ese elemento.

Cuando un elemento pierde el foco (se "desenfoca", "blur"), la entrada desde teclado ya no irá dirigida a ese elemento. Pero además, indica que los datos ya han sido introducidos. Puede ser el momento de ejecutar código que valide los datos.

Eventos focus y blur

Evento	Descripción
`focus`	Un elemento obtiene el foco. Este evento no se propaga.
`blur`	Un elemento pierde el foco. Este evento no se propaga

Métodos focus y blur

Método	Descripción
`focus()`	Pone el foco sobre el elemento
`blur()`	Quita el foco del elemento

(*) El foco se puede perder cuando el usuario hace clic en otro lado. Pero el propio JavaScript puede causarlo:

- Un `alert` traslada el foco hacia sí mismo. El elemento que lo tenía, lo pierde.
- Si un elemento se elimina, pierde el foco.

~~Permitir enfocado sobre cualquier elemento~~

Por defecto, no todos los elementos permiten el enfoque. Lo permiten aquellos con los que el visitante puede interactuar `<button>`, `<input>`, `<select>`, `<a>`, etc.

Pero este comportamiento por defecto se puede modificar. Si un elemento html tiene el atributo `tabIndex`, puede tener el foco. Los valores de este atributo son 1, 2, 3... indican que un elemento puede tener el foco y en qué orden se asigna. Si un elemento tiene `tabIndex`=0, puede tener el foco, pero obtiene el foco en el orden que aparece en el código.

Se puede asignar a un elemento el atributo `tabIndex` con `elemento.setAttribute("tabIndex")` o con `elemento.tabIndex=.`

~~Delegación: focusin/focusout~~

Los eventos `focus` y `blur` no se propagan, por lo que no se puede utilizar delegación de eventos. Existen dos eventos `focusin` y `focusout` que son equivalentes a `focus` y `blur`, pero se propagan.

Evento	Descripción
`focusin`	Un elemento obtiene el foco. Este evento se propaga.
`focusout`	Un elemento pierde el foco. Este evento se propaga.

3. Eventos: change, input, cut, copy, paste

Evento: change

Se activa cuando el elemento finaliza un cambio.

- input de tipo texto: cuando pierde el foco.
- input de tipo `boolean` (`type`=`checkbox`/`radio`): cuando se selecciona una opción.

Evento:input

Se dispara cada vez que un valor es modificado por el usuario. No espera a que finalice el cambio. Ejemplo, cuando escribimos o borramos una letra dentro del `input`.

Eventos:cut, copy, paste

Se disparan al cortar/copiar/pegar un valor.

Se puede acceder a los datos cortados/copiados/pegados.

Se puede impedir su funcionamiento con `event`.`preventDefault`().

Método	Descripción
`event.clipboardDate(getData("text/plain"))`	Contenido del portapapeles
`document.getSelection()`	Contenido del portapapeles

4. Formularios: evento y método submit

Evento: submit

Es un evento del formulario.

Se activa cuando el formulario es enviado, normalmente haciendo clic en un botón de `type=submit`, o pulsando `enter` en un campo de texto del formulario.

Se suele utilizar para validar el formulario y permitir el envío al servidor, o impedirlo.

Método: submit()

Es un método del formulario.

Permite enviar el formulario al servidor manualmente. Si se llama a este método, no se dispara en evento `submit`.