

Formularios accesibles (2)

Esta lección pertenece al [curso “Aprende Accesibilidad Web Paso a Paso”](#) ¹

Prohibida su distribución sin permiso del autor.

Presentación

Hola, soy Olga Carreras Montoto, consultora en Experiencia de Usuario y Accesibilidad web en [“Usable y accesible”](#) ² (España).

Esta lección es la continuación de “Formularios accesibles (1)” donde aprendimos a etiquetar correctamente los controles de los formularios.

En esta lección vamos a centrarnos en los propios controles de un formulario web, en la relación entre los mismos y en los requisitos de accesibilidad que deben cumplir.

Usar controles estándar

La primera norma es utilizar controles de formulario estándar (<input>, <textarea>, <select>, <button>, etc.) siempre que sea posible y hacerlo conforme a la especificación de HTML 4 o de HTML5, según la que estemos usando en las páginas³.

Usar controles estándar de HTML nos asegura que:

¹ <http://accesibilidadweb.es>

² “Usable y accesible”, <http://www.usableyaccesible.com>

³ W3C, [HTML 4.01 Specification, “Forms”](#),

<http://www.w3.org/TR/html401/interact/forms.html>

W3C, [HTML 5, “Forms”](#), <http://www.w3.org/TR/html5/forms.html#forms>

- puedan ser operables por teclado;
- su nombre, función, estado y valor, así como los cambios en los mismos, puedan ser determinados por software y, de este modo, ser anunciados por ejemplo a los usuarios de productos de apoyo, como un lector de pantalla.

Técnica de referencia de las WCAG 2.0: [H91: Using HTML form controls and links](#)⁴.

Si no usamos controles estándar

A veces no podemos usar controles estándar, por ejemplo porque no existen, como es el caso de un slider en HTML4⁵.

A menudo simulamos o creamos un nuevo control mediante otras etiquetas estándar, -que en este caso tienen una función diferente a la suya habitual-, y programación javascript.

En estos casos será muy importante que definamos la nueva función, estado y propiedades de estos elementos, así como los cambios en los mismos, mediante WAI-ARIA, que es un tema que trataremos en detalle en lecciones posteriores.

Estructura y relaciones

También es importante informar semánticamente de la estructura y las relaciones entre los campos que se están comunicando a través de la presentación, para que puedan ser determinadas por software, es decir, que las aplicaciones (como los productos de apoyo) pueden extraer y presentar este tipo de información de distintas maneras.

⁴ WCAG 2.0, Técnica “H91: Using HTML form controls and links”,
<http://www.w3.org/TR/WCAG20-TECHS/H91.html>

⁵ Se puede consultar un ejemplo en [“Basic ARIA Slider”](#),
<http://files.paciellogroup.com/blogmisc/ARIA/slider>

Para ello, además de aplicar los requisitos que vimos en el primer vídeo en relación con el correcto etiquetado de los campos, podemos usar elementos como FIELDSET y OPTGROUP.

<fieldset> y <legend>

FIELDSET nos permite agrupar semánticamente una serie de controles de formulario y etiquetarlos en su conjunto con una descripción del grupo, que incluimos mediante el elemento LEGEND, y de esta manera comprender la relación entre los controles e interactuar con el formulario de forma más rápida y efectiva.

En el siguiente ejemplo, que ya veíamos en la lección anterior, los campos están agrupados semánticamente mediante el elemento FIELDSET:

```
<fieldset>

<legend>Fecha de nacimiento
(dd/mm/aaaa):</legend>

<input type="text" id="dia" name="dia"
title="Día (dos dígitos)" /> /

[...]

</fieldset>
```

Su primer elemento es LEGEND, que nos permite proporcionar una etiqueta o descripción del grupo.

A menudo, los productos de apoyo anunciarán esta leyenda antes de la etiqueta de cada control para proporcionarles a los usuarios esta descripción y recordarles que esos campos forman parte del mismo grupo.

El uso de FIELDSET es especialmente importante en los **grupos de radios y checks**. En estos casos la etiqueta individual de cada uno no transmite plenamente el contexto descriptivo del grupo, y es

esencial que se agrupen para facilitar que sean tratados semánticamente como un único conjunto con una descripción adicional a nivel de grupo.

Por ejemplo:

```
<fieldset>

    <legend>Estoy interesado en recibir
información sobre:</legend>

    <input type="checkbox" id="deportes"
name="deportes" value="dp">

    <label for="deportes">Deportes</label>

[...]

</fieldset>
```

Por últimos, dos recomendaciones acerca del uso de FIELDSET.

- Evita anidar elementos FIELDSET innecesariamente, pues puede crear confusión.
- Intenta conservar la agrupación visual. A veces, por razones de estilo, se elimina el borde que se presenta por defecto alrededor de los controles agrupados. Esta agrupación visual también es útil y debería intentar mantenerse, aunque sea modificando su estilo, pero que permita seguir comprendiendo visualmente la relación entre los campos y su leyenda.

Os recomiendo la lectura del artículo [“Asociación compleja de radios y labels de forma accesible”](http://olgacarreras.blogspot.com.es/2010/12/asociacion-compleja-de-radios-y-labels.html)⁶, donde podéis consultar un caso de asociación compleja entre controles de formulario, resuelta en gran medida con el elemento FIELDSET.

⁶ Carreras Montoto, Olga: “Asociación compleja de radios y labels de forma accesible”, <http://olgacarreras.blogspot.com.es/2010/12/asociacion-compleja-de-radios-y-labels.html>, 2010

Técnica de referencia de las WCAG 2.0: [H71: Providing a description for groups of form controls using fieldset and legend elements](http://www.w3.org/TR/WCAG20-TECHS/H71.html)⁷

<optgroup>

Otro elemento que nos permite informar semánticamente de la relación entre una serie de elementos es OPTGROUP.

OPTGROUP permite agrupar una serie de OPTIONS dentro de una SELECT, de selección múltiple o no, porque a menudo las listas de selección tienen grupos de opciones relacionadas.

Por ejemplo:

```
<label for="comida">Selecciona tu comida  
preferida:</label>  
  
<select id="comida" name="comida">  
  <optgroup label="Fruta">  
    <option value="1">Manzanas</option>  
    <option value="2">Plátanos</option>  
    <option value="3">Moras</option>  
  </optgroup>  
  [...]</select>
```

Estos grupos deben ser identificados semánticamente mediante OPTGROUP, como se indica en el ejemplo anterior, nunca simularlos.

Además, OPTGROUP ayuda a romper visualmente las listas largas y a que los usuarios puedan localizar más fácilmente la opción deseada.

⁷ WCAG 2.0, Técnica "H71: Providing a description for groups of form controls using fieldset and legend elements", <http://www.w3.org/TR/WCAG20-TECHS/H71.html>

Técnica de referencia de las WCAG 2.0: [H85: Using OPTGROUP to group OPTION elements inside a SELECT](#)⁸

Cambios de contexto

Por último, **los controles de formulario no pueden provocar un cambio de contexto**, -como abrir otra página o ventana, mover el foco a otro componente o enviar un formulario-, **de forma automática o sin conocimiento del usuario**, ni cuando el control coja el foco, ni al entrar datos o cambiar su estado, como explicaremos con más detalle en una lección posterior.

Por ejemplo, el formulario se debe enviar con un botón “submit”, que permite a los usuarios solicitar explícitamente enviar los datos:

```
<input type="submit" name="enviar"
value="Enviar" />
```

O por ejemplo, si tenemos una “select” que permite navegar a una página diferente en función de la opción seleccionada, no podemos hacerlo automáticamente cuando el usuario seleccione la opción, sin avisarle antes, sino que preferiblemente incluiremos junto a la “select” un botón para que sea el usuario el que lo solicite explícitamente.

Técnicas de referencia de las WCAG 2.0:

- [H32: Providing submit buttons](#)⁹
- [H84: Using a button with a select element to perform an action](#)¹⁰

⁸ WCAG 2.0, Técnica “H85: Using OPTGROUP to group OPTION elements inside a SELECT”, <http://www.w3.org/TR/WCAG20-TECHS/H85.html>

⁹ WCAG 2.0, Técnica “H32: Providing submit buttons”, <http://www.w3.org/TR/WCAG20-TECHS/H32.html>

¹⁰ WCAG 2.0, Técnica “H84: Using a button with a select element to perform an action”, <http://www.w3.org/TR/WCAG20-TECHS/H84.html>

Por último, si queréis ampliar información sobre los nuevos controles de formulario en HTML 5 y su soporte con diferentes lectores de pantalla, os recomiendo mi artículo [“HTML5 y accesibilidad: nuevos tipos de input, atributos asociados y validación nativa”](#)¹¹.

Siguiente lección

A continuación te invito a que visualices “Formularios accesibles (3)” donde aprenderemos a prevenir los errores de los usuarios en los formularios y a mejorar la accesibilidad de los mensajes de validación de los mismos.

Autor

Olga Carreras Montoto. Consultora en Experiencia de Usuario y Accesibilidad Web en [“Usable y accesible”](#) (España)

Ya para terminar, estas son las direcciones de mi blog, de mi web y de mi cuenta de Twitter. En ellas puedes encontrar información sobre mí y sobre mi trabajo, así como más artículos sobre accesibilidad.

- Blog: <http://olgacarreras.blogspot.com.es>
- Web: <http://www.usableyaccesible.com>
- Twitter: <https://twitter.com/olgacarreras>

¹¹ Carreras Montoto, Olga: “HTML5 y accesibilidad: nuevos tipos de input, atributos asociados y validación nativa”, <http://olgacarreras.blogspot.com.es/2014/03/html5-y-accesibilidad-nuevos-tipos-de.html>, 2014

Referencias

- Carreras Montoto, Olga: ["Formularios accesibles según las WCAG 2.0"](http://olgacarreras.blogspot.com.es/2009/06/formularios-accesibles-segun-las-wcag.html):
<http://olgacarreras.blogspot.com.es/2009/06/formularios-accesibles-segun-las-wcag.html>, 2009
- Carreras Montoto, Olga: ["Asociación compleja de radios y labels de forma accesible"](http://olgacarreras.blogspot.com.es/2010/12/asociacion-compleja-de-radios-y-labels.html),
<http://olgacarreras.blogspot.com.es/2010/12/asociacion-compleja-de-radios-y-labels.html>, 2010
- Carreras Montoto, Olga: ["HTML5 y accesibilidad: nuevos tipos de input, atributos asociados y validación nativa"](http://olgacarreras.blogspot.com.es/2014/03/html5-y-accesibilidad-nuevos-tipos-de-input-atributos-asociados-y-validacion-nativa.html),
<http://olgacarreras.blogspot.com.es/2014/03/html5-y-accesibilidad-nuevos-tipos-de.html>, 2014
- PACIELLO GROUP, ["Basic ARIA Slider"](http://files.paciellogroup.com/blogmisc/ARIA/slider),
<http://files.paciellogroup.com/blogmisc/ARIA/slider>
- W3C, [HTML 4.01 Specification, "Forms"](http://www.w3.org/TR/html401/interact/forms.html),
<http://www.w3.org/TR/html401/interact/forms.html>
- W3C, [HTML5, "Forms"](http://www.w3.org/TR/html5/forms.html#forms),
<http://www.w3.org/TR/html5/forms.html#forms>
- W3C, [Web Content Accessibility Guidelines \(WCAG\) 2.0](http://www.w3.org/TR/WCAG20/),
<http://www.w3.org/TR/WCAG20/>, 2008

Técnicas de las WCAG 2.0 vistas en esta lección:

- [H91: Using HTML form controls and links](http://www.w3.org/TR/WCAG20-TECHS/H91.html),
<http://www.w3.org/TR/WCAG20-TECHS/H91.html>
- [H71: Providing a description for groups of form controls using fieldset and legend elements](http://www.w3.org/TR/WCAG20-TECHS/H71.html),
<http://www.w3.org/TR/WCAG20-TECHS/H71.html>

- [H85: Using OPTGROUP to group OPTION elements inside a SELECT](http://www.w3.org/TR/WCAG20-TECHS/H85.html), <http://www.w3.org/TR/WCAG20-TECHS/H85.html>
- [H32: Providing submit buttons](http://www.w3.org/TR/WCAG20-TECHS/H32.html), <http://www.w3.org/TR/WCAG20-TECHS/H32.html>
- [H84: Using a button with a select element to perform an action](http://www.w3.org/TR/WCAG20-TECHS/H84.html), <http://www.w3.org/TR/WCAG20-TECHS/H84.html>

Mejorar la accesibilidad para los usuarios con discapacidad casi siempre mejora la usabilidad para todos los usuarios. J.Nielsen