



ACCESIBILIDAD

David Naranjo Moro



9 DE FEBRERO DE 2022

2 DAW

IES RIBERA DE CASTILLA

Índice

1.0 Asociar una etiqueta con cada control de formulario	2
2.0 Incluir texto alternativo para las imágenes	2
3.0 Identificar el idioma de la página y los cambios de idioma....	3
4.0 Utilice el marcado para transmitir significado y estructura	4
5.0 Ayudar a los usuarios a evitar y corregir errores	5
6.0 Refleja el orden de lectura en el orden del código.....	7
7.0 Escribir código que se adapte a la tecnología del usuario	7
8.0 Proporcionar significado para elementos interactivos no estándar.....	8
9.0 Asegúrese de que todos los elementos interactivos sean accesibles desde el teclado.....	9
10.0 Evite CAPTCHA siempre que sea posible	9

1.0 Asociar una etiqueta con cada control de formulario

- Es muy buena práctica de maquetación asignar un atributo identificativo a nuestros elementos así facilitamos a los desarrolladores que ven nuestro código una mejor accesibilidad y mejor manejo de los datos:

Ejemplo: `<label
for="username">Username</label>
<input id="username" type="text"
name="username">`

Username

- Asignamos al label con for un nombre identificativo para el campo. Y lo mismo haremos para los input utilizando id.

2.0 Incluir texto alternativo para las imágenes

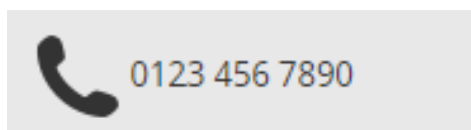
- Es muy recomendable que cuando nosotros tenemos una página web, donde mostramos información visual aplicar un método para personas que sufren discapacidad visual y utilizan algún tipo de lector de voz. Con esto conseguimos que personas con esa discapacidad puedan saber qué información están mostrando.

EJEMPLO: ``



- También podemos ya que es una buena práctica aplicar a las imágenes algo de texto para complementar esa información que se está dando, ya que aportar solo el texto para representar una imagen suele ser insuficiente así que deberíamos proporcionar algo más de información

EJEMPLO: ` 0123 456 7890`



- Otra practica muy interesante es cuando tenemos imágenes que nos redirigen a otras páginas cuando pinchamos en ellas o pasamos el botón por encima proporcionar un texto alternativo para aportar información del lugar al que nos va a redirigir

EJEMPLO

```
<a href="https://www.w3.org/">  W3C Home </a>
```

Este es un ejemplo de accesibilidad, pero se puede proporcionar información que se abrirá la página en una nueva ventana, o realizará x funcionalidad.



- Otro ejemplo claro es cuando tenemos imágenes de empresas o de productos y queremos informar al usuario como se llama esa empresa o cual es el producto a través de un texto informativo.

Ejemplo



3.0 Identificar el idioma de la página y los cambios de idioma

- En nuestros documentos HTML tendremos que indicar con el atributo Lang el idioma de la página ya que esto es una buena metodología de trabajo. Aunque nosotros en nuestra página web haya información de una lengua que no es la del idioma establecido en la página nosotros podemos aplicar a un elemento solo el atributo Lang, en casos por ejemplo donde tenemos un enlace donde indicamos que es un página con otro idioma a través de texto, imágenes etc.

Ejemplo:

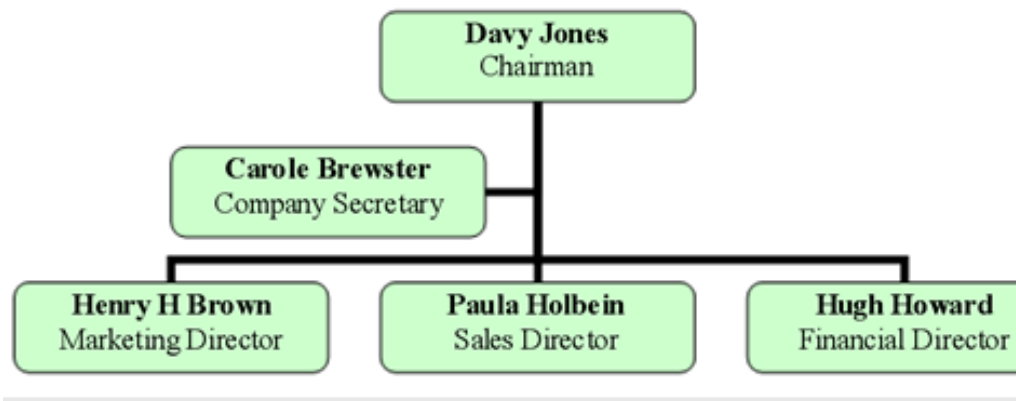
```
<html lang="fr">
<a lang="es" title="Spanish" href="qa-html-language-declarations.es">Español</a>
</html>
```

- Un último ejemplo sería a través de aportar información mediante un organigrama donde cada elemento de este tiene un texto alternativo y mediante la etiqueta área y map de html 5 podemos dividir el organigrama por elementos. Esta etiqueta aporta unos atributos muy interesantes ya que hay atributos como coords para proporcionar coordenadas escritas.

Ejemplo:

```

<map id="Map" name="Map">
<area shape="rect" coords="176,14,323,58" href="["...]"
alt="Davy Jones: Chairman"> [...]
<area shape="rect"
coords="6,138,155,182"
href="["...]"
alt="Harry H Brown: Marketing Director
(reports to chairman)"> [...] </map>
```



4.0 Utilice el marcado para transmitir significado y estructura

- Para una buena construcción de un documento tenemos que utilizar etiquetas correspondientes para los elementos adecuados. Por ejemplo, no podemos pretender poner un título de encabezado en una etiqueta <p> ya que no es una buena práctica, aunque nosotros con ccs apliquemos estilos para que parezca un encabezado, no es una buena práctica de legibilidad del html. Hay que utilizar para cada elemento su etiqueta correspondiente.

Ejemplo:

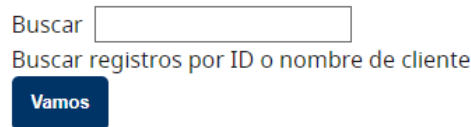
```
<section>
  <article>
    <h2>Superbear saves the day</h2>
    <time datetime="2015-08-07">7 Aug 2015</time>
    <p>The city's favorite bear yet again proves his
mettle by rescuing a young cat from a tree. Witnesses
say that Superbear's efforts were not appreciated by
```

the feline, who inflicted some minor scratch wounds on his rescuer.</p>

```
<aside>
  <h3>Related Articles</h3>
  <ul>
    <li><a href="#">Bear receives key to
city</a></li>
    <li><a href="#">Superbear stands for
mayor</a></li>
  </ul>
</aside>
</article>
</section>
```

- También podemos utilizar un atributo llamado role, donde indicaremos que acción realizara el elemento al que está asociado ese atributo, puede ser un enlace un botón un campo de texto etc....

```
<form action="#" method="post">
  <div role="search">
    <label for="search">Search for</label>
    <input type="search" id="search" aria-
describedby="search-help">
    <div id="search-help">Search records by customer
id or name</div>
    <button type="submit">Go</button>
  </div>
</form>
```

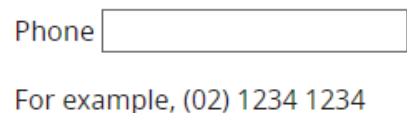


5.0 Ayudar a los usuarios a evitar y corregir errores

- Los usuarios que acceden a nuestras páginas web tenemos que proporcionarles un manejo de errores lo más optimizado y conciso posible. Esto quiere decir que, si un usuario está introduciendo un campo indicarle, que es lo que está haciendo mal o antes de que inserte los datos una breve descripción o indicar que un campo es requerido o incluso aplicarle una expresión al campo, pero siempre proporcionando información al usuario.

Ejemplo1:

```
<label for="phone">Phone</label>
<input id="phone" name="phone" type="tel"
  pattern="^\(\(?0[1-
9]{1}\)\)?[0-9 -]*$"
  aria-describedby="phone-
desc">
<p id="phone-desc">For example,
(02) 1234 1234</p>
```



Ejemplo 2:

```
<div>
<label for="email">Email:</label>
<input type="email" name="email" id="email"> </div>
<div>
<label for="url">Website: </label>
<input type="url" name="url" id="url"> </div>
<div> <label for="number">Number: </label>
<input type="number" name="number" id="number" min="0"
max="100" step="10" value="0"> </div>
<div> <label for="range">Range: </label>
<input type="range" name="range" id="range" min="0"
max="100" step="10" value="0"> </div>
<div> <label for="date">Date: </label>
<input type="date" name="date" id="date"> </div>
<div> <label for="time">Time: </label> <input type="time"
name="time" id="time"> </div>
```


EXAMPLE:


Email:

Website:

Number:

Range:

Date: 

Time: 

6.0 Refleja el orden de lectura en el orden del código.

- Hay que tener un orden correcto en el código ya que nosotros podemos tener un artículo donde tenemos una imagen y debajo un encabezado que indica información de la imagen. Pero este orden no es correcto y perjudicaría en el resultado visual de la página. Aunque nuevamente comento nosotros podamos alinear los elementos con el CSS de la forma que queramos, pero no es una buena práctica ya que debemos tener una buena estructuración del código ya que depende del cargado de los elementos que hay se puede perder información por una mala estructuración.

Ejemplo no correcto:

```

<h3>Space trainers</h3>
<p>Space...</p>
<a href="...">Add to cart</a>
```

Ejemplo correcto:


```
<h3>Space trainers</h3>

<p>Space...</p>
<a href="...">Add to cart</a>
```



Space trainers

Space trainer for a classic and stylish look.

 Add to cart

7.0 Escribir código que se adapte a la tecnología del usuario

- Una de las claras características de accesibilidad que debemos tener en nuestro sitio web es que sea responsive, ya que se tiene que adaptar para cualquier tipo de dispositivo que pueda visitar nuestro sitio web o al menos para los dispositivos más usados o para resoluciones de pantallas más actuales. Aunque es recomendable que la página web sea responsive para incluso pantallas de una resolución baja.

Ejemplo:

```
/* On narrow viewports, make the navigation full width */
@media screen and (min-width: 25em) {
  #nav {
    float: none;
    width: auto;
  }
  #main {
    margin-left: 0;
  }
}
```



```

    }
}
/* On wider viewports, put the navigation on the left
*/
@media screen and (min-width: 43em) {
    #nav {
        float: left;
        width: 24%;
    }
    #main {
        margin-left: 27%;
    }
}

```

- Obviamente tendremos que situar los elementos que interactúan en nuestra página web bien situados para aquellos dispositivos con una resolución menor como los móviles. No podemos ubicar los elementos importantes con los que se interactúan en la página en el mismo lugar de una pantalla de 1080p.

8.0 Proporcionar significado para elementos interactivos no estándar.

- Se comentó anteriormente que es muy importante asignar un rol a los elementos para indicar el cometido que realizara, ya sea buscar datos en una base de datos, enviar a un enlace externo etc.....
- También la utilización de los atributos aria ya que podemos inhabilitar un campo de texto para que cuando ocurra una determinada opción quede habilitado para el usuario o utilizar un aria expandida en el caso que tengamos una lista y cuando se abra un submenú obligue al lector de pantalla a anunciar que el elemento del menú está expandido.

Ejemplo:

```

<nav aria-label="Main Navigation" role="navigation">
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">Shop</a></li>
    <li class="has-submenu">
      <a aria-expanded="false" aria-haspopup="true"
href="#">SpaceBears</a>
      <ul>
        <li><a href="#">SpaceBear 6</a></li>
        <li><a href="#">SpaceBear 6 Plus</a></li>
      </ul>
    </li>
    <li><a href="#">MarsCars</a></li>
    <li><a href="#">Contact</a></li>
  </ul>
</nav>

```

9.0 Asegúrese de que todos los elementos interactivos sean accesibles desde el teclado

- En nuestra página web podemos tener usuarios que pueden tener alguna discapacidad de movilidad para estos casos debemos tener preparada la página a través de la utilización de scripts para poder configurar un menú de acceso a través del teclado o mediante usos de librerías externas configurar un grabador de voz para campos que son tediosos de rellenar para una persona con discapacidad como lo son los campos DNI, expedientes etc.

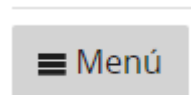
Ejemplo:

```
var buttonExample = document.getElementById('example-button');

buttonExample.addEventListener('keydown', function(e) {
  // Toggle the menu when RETURN is pressed
  if(e.keyCode && e.keyCode == 13) {

toggleMenu(document.getElementById('example-button-menu'));
  }
});

buttonExample.addEventListener('click', function(e) {
  // Toggle the menu on mouse click
  toggleMenu(document.getElementById('example-button-menu'));
});
```



10.0 Evite CAPTCHA siempre que sea posible

- Los captcha es una forma útil para verificar que el usuario que esta accediendo a la pagina web es un ser humano, pero genera problemas para personas que puedan tener algún tipo de discapacidad ya que si una persona no puede ver el captcha no puede pasarlo.

Practicas interesantes que realizar cuando con CAPTCHA:

1. Proporcionando más de dos formas de resolver los CAPTCHA
2. Proporcionar acceso a un representante humano que puede omitir CAPTCHA
3. No requiere CAPTCHA para usuarios autorizados.