

# PROGRAMACIÓN MULTIPLATAFORMA

HTML5



# Envío de información a servidores

2

- Un formulario es parte del contenido de una página web formado por diferentes campos que permiten introducir información.
- El usuario desencadena el envío de la información, por ejemplo, mediante la pulsación de un botón.
- El navegador empaqueta la información y se la envía al servidor.



# Funcionamiento (I)

3

- Se proporciona al usuario el entorno visual del formulario.

**Regístrese**  
  
Usuario  
  
  
Contraseña  
  
  
¿Cómo nos conoció?  
☒ Internet  
☐ Prensa  
☐ Otros  
  
☒ Suscribirse al boletín de novedades



# Funcionamiento (II)

4

- El usuario rellena la información con sus datos y pulsa el botón de envío (en este caso el botón *Registrarme*)

**Regístrese**  
  
Usuario  
  
  
Contraseña  
  
  
¿Cómo nos conoció?  
☐ Internet  
☐ Prensa  
☒ Otros  
  
☐ Suscribirse al boletín de novedades



# Funcionamiento (III)

5

- Una vez que el servidor tiene los datos procesa la información.
- Informa al usuario de información relevante al respecto (registro correcto, usuario ocupado, contraseña incorrecta...).



# Formularios HTML

6

- Los elementos de los que se dispone para introducir información en un formulario se llaman controles o campos de formulario.
- A los datos en sí mismos se les llama variables.
- Para la introducción de las variables se utiliza la etiqueta **<input>**. Esta etiqueta tiene el atributo **type** que indica el tipo del control a introducir y **name** que indica el nombre que se le dará al campo.



# Accesibilidad. Label.

7

- El elemento **label** permite indicar semánticamente qué texto funciona como etiqueta de un campo de formulario.
  - `<label for="nombre">Nombre:</label>`
  - `<input type="text" name="nombre" id="nombre" />`
- Para establecer la relación entre la etiqueta y el control del formulario debemos asociarla explícitamente con el atributo **for**, cuyo valor debe ser igual al del **id** del control del formulario, identificador que, a su vez, debe ser único en la página.
- En los botones no debe utilizarse label.



# Accesibilidad. Label. Posición.

8

- Antes del campo, bien encima, bien a la izquierda, con los elementos “textarea”, “select”, y los “input” de tipo “text”, “file” y “password”.
- Detrás del campo con los “input” de tipo “radio” y “checkbox”.





# Accesibilidad. `<fieldset>` y `<legend>`

9

- **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.
- 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.



# Accesibilidad. Prevención de errores

## (I).

10

- En primer lugar debemos identificar los campos obligatorios, preferiblemente con un texto en la etiqueta del campo, y por tanto antes del mismo.
- Por ejemplo:
  - ▣ `<label for="nombre">Nombre (obligatorio):</label>`
  - ▣ `<input type="text" name="nombre" id="nombre" />`
- No podemos identificar los campos obligatorios solo con el color, por ejemplo con el color del propio campo o de su etiqueta. Sí puede ser una pista visual adicional.
- Adicionalmente podemos añadir al campo la propiedad **aria-required**, pero si ya tenemos el atributo **required**, no hace falta que lo pongamos porque si lo hacemos el lector de pantalla lo leerá dos veces.
- Por ejemplo:
  - ▣ `<input type="text" name="nombre" id="nombre" aria-required="true" />`



# Accesibilidad. Prevención de errores

## (II).

11

- Otra manera de evitar errores a los usuarios es informándoles de las restricciones en el formato de datos que deben introducir, aunque debería intentarse que los formularios fueran flexibles para introducir diferentes formatos de datos.
- La propiedad **aria-describedby** de la especificación WAI-ARIA identifica el elemento o los elementos que le proporcionan una descripción al objeto que contiene esta propiedad, es decir, que le proporcionan una información adicional.
- Por ejemplo, se puede indicar el formato requerido en la etiqueta del campo y proporcionar un ejemplo bajo el mismo:
  - `<label for="fecha">Fecha (dd/mm/aaaa):</label>`
  - `<input type="text" aria-describedby="ayuda" ... />`
  - `<p id="ayuda">Por ejemplo: 12/03/2014</p>`



# Accesibilidad. Mensajes de validación.

12

- Preferiblemente, aunque se admiten otras opciones (como ventanas de alertas o información contextual):
  - ▣ Informando mediante texto, al comienzo del formulario, de los errores existentes.
  - ▣ Describiendo el problema e identificando los campos con error (sin basarse solo en el color, como hemos visto anteriormente).
  - ▣ Si es posible, indicando o sugiriendo valores posibles.



# Secciones y grupos

13

- Las secciones son elementos esenciales en HTML.
- Combinadas con las hojas de estilo CSS ofrecen ilimitadas posibilidades para estructurar y distribuir contenidos en nuestras páginas web.
- Las capas son contenedores HTML que se utilizan para albergar a otros elementos.



# Etiqueta <div>

14

- ❑ La etiqueta por excelencia para realizar capas en HTML se llama **<div>**.
- ❑ Esta etiqueta agrupa grandes (o pequeñas) cantidades de contenido.



# Etiquetas semánticas HTML5 (I).

15

- ❑ Las etiquetas semánticas son aquellas que dan un significado a las partes del documento.
- ❑ Son etiquetas que indican qué es el contenido que contienen.
- ❑ HTML5 ha incorporado muchas etiquetas que sirven para indicar qué son y cuál es el significado de los elementos que contienen dentro.
- ❑ Google las usa para indexar mejor los contenidos de un sitio.



# Etiquetas semánticas HTML5 (II).

16

- Generalmente, en cualquier documento tenemos una cabecera, un cuerpo y un pie de página.
- De este modo, HTML5 tiene una etiqueta **header**, que debe englobar aquellos elementos correspondientes a la cabecera de la página.
- Tiene un **footer**, que debe englobar todas las etiquetas que componen el pie.
- Existirán además otras etiquetas que representan elementos típicos como la barra de navegación, etiqueta **nav**.





# Etiquetas semánticas HTML5 (III).

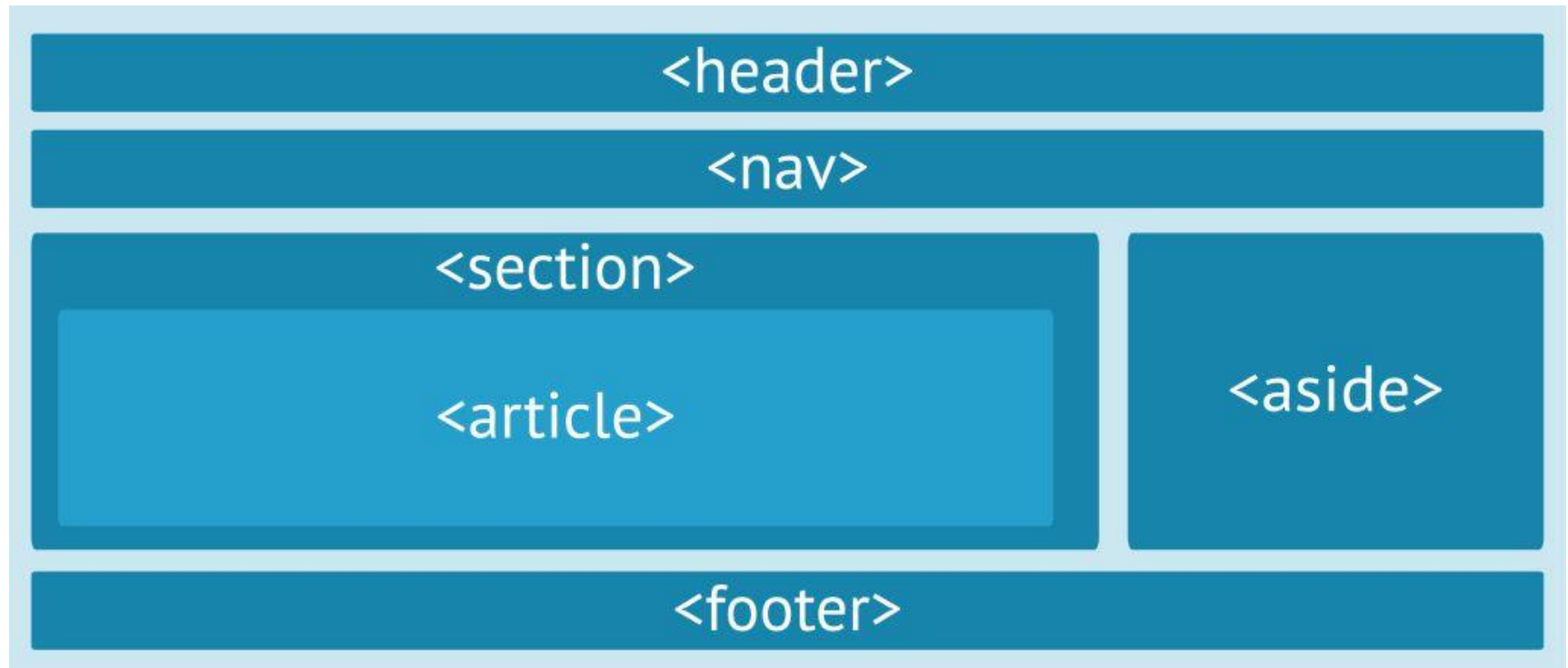
17

- Las etiquetas semánticas no tienen un estilo predeterminado que el navegador nos vaya a asignar (header no irá automáticamente arriba ni footer debajo).
- Google recomienda su uso para indexar los sitios web.
- Será capaz de leer el documento y saber qué parte es la cabecera o que parte contiene los enlaces a las principales secciones del sitio.



# Etiquetas semánticas más importantes (I)

18



# Etiquetas semánticas más importantes

## (II)

19

- **<header>**: Se usa para enmarcar un grupo de elementos de introducción. Suele envolver las etiquetas <h1>, <h2>... Ejemplos:

```
<header>
    <h1>Título del documento</h1>
    <p>Más información</p>
</header>

<article>
    <header>
        <h1>Título del artículo</h1>
        <p>Autor del artículo</p>
    </header>
    <p>Texto del artículo</p>
</article>
```



# Etiquetas semánticas más importantes

## (III)

20

- **<main>**: Se usa como contenedor general de la aplicación.
- Todo el contenido debe estar relacionado directamente.
- No debe haber más de un main en un documento y este no debe ser descendiente de un section o article.



# Etiquetas semánticas más importantes

## (IV)

21

- **<nav>**: se usa para marcar una sección del documento cuya función es la navegación por la página web.
- No debe ser usada para marcar todos los grupos de enlaces, únicamente los bloques principales de navegación por la página.
- Por ejemplo, los típicos enlaces que hay en el pie de página no se deben marcar con **<nav>**, sino como **<footer>**.

```
<nav>
  <ul>
    <li><a href="cursohtml.html">Curso de Html5</a></li>
    <li><a href="cursocss.html">Curso de CSS</a></li>
    <li><a href="cursojavascript.html">Curso de Javascript</a></li>
  </ul>
</nav>
```



# Etiquetas semánticas más importantes

## (V)

22

- **<section>**: Se utiliza para encerrar el código correspondiente a una sección genérica dentro de un documento o aplicación.
- Normalmente, un bloque de texto al que perfectamente le podríamos colocar un título o encabezado.
- Todo el contenido que engloba ha de guardar cierta relación entre sí.
- Dentro de un section grande pueden existir otros section hijo. Por ejemplo, podría haber una sección noticias con otras secciones de diferentes áreas (economía, deportes...).



# Etiquetas semánticas más importantes

## (VI)

23

- **<article>**: Pensada para separar cada uno de los elementos independientes que pudieran componer un section concreto.
- Un article podría considerarse por tanto como cada uno de los elementos en que podemos dividir un section.



# Etiquetas semánticas más importantes (VII)

24

```
<header>Título de la página</header>
<section id="news-list">
  <h2>Noticias</h2>

  <article>
    <h3>Noticia 1</h3>
    <p>Desarrollo de la noticia 1</p>
  </article>

  <article>
    <h3>Noticia 2</h3>
    <p>Desarrollo de la noticia 2</p>
  </article>
</section>
```





# Etiquetas semánticas más importantes

## (VIII)

25

- **<aside>**: Con las etiquetas `<aside>` y `</aside>`, se nos invita a rodear todo aquel contenido que no es directamente contenido principal del que estamos hablando o del que estamos tratando en esa página en concreto.
- Lo usaremos por tanto para todos aquellos elementos secundarios, como podrían ser los bloques publicitarios, enlaces externos, citas, un calendario de eventos, etc.

```
<aside>
```

```
  <h1>Publicidad</h1>
```

```
  (( código de la publicidad ))
```

```
</aside>
```



# Etiquetas semánticas más importantes

## (IX)

26

- **<footer>**: Se trata de una etiqueta para encerrar todo aquel código típico de las partes inferiores de las páginas web.
- Contendrá por tanto enlaces a otras webs relacionadas, al mapa de la web, a una página de comentarios sobre el copyright o a una política del portal.

```
<footer>
  <p>Página del manual de HTML5</p>
  <p>Contacta con nosotros: email <a
href="mailto:xxxx@xxx.com">xxxxb@xxx.com</a></p>
</footer>
```



# Elementos multimedia (I)

27

- Los elementos audiovisuales tienen unas características que deben ser compañeros de viaje indispensables en todos nuestros sitios webs.
- Entre otras cosas:
  - ▣ Ayudan enormemente al estímulo sensorial del usuario. Le ayudan a captar lo que se quiere transmitir con mucha rapidez mediante la vista y el oído.
  - ▣ Aportan un plus de calidad al resultado final.
  - ▣ Ayudan a retener al usuario en el sitio web.



# Elementos multimedia (II)

28

- Existen varios formatos con los que podemos trabajar.
- El mayor problema (cada vez menor) es que no todos los navegadores aceptan todos los formatos, haciendo necesario el uso de extensiones (*plugins*) en ellos.
- Esto representa un gran obstáculo, ya que entendemos que el usuario:
  - ▣ Tiene la capacidad de instalar en su propia máquina estos añadidos.
  - ▣ Tiene la voluntad de hacerlo.



# Elementos multimedia (III)

29

- ❑ Los navegadores modernos que soportan el nuevo estándar HTML5 no suelen requerir este tipo de extensiones.
- ❑ Antes de HTML5 no había un estándar de reproducción de video.
- ❑ Antes de HTML5 para reproducir video se necesitaba tener el plugin necesario para el formato a reproducir.
- ❑ La etiqueta HTML5 `<video>` aporta una manera única de reproducir vídeo.



# Etiqueta `<video>` (I)

30

- ❑ Para reproducir un archivo de vídeo disponemos de la etiqueta HTML5 **`<video>`**.
- ❑ Su atributo **`src`** permite especificar la ruta del archivo de vídeo a reproducir.
- ❑ Otros atributos:
  - ❑ **`controls`**: Si incluimos este atributo el navegador mostrará la barra de control del reproductor de audio/vídeo.
  - ❑ **`loop`**: Si incluimos este atributo activamos la reproducción indefinida.



# Etiqueta `<video>` (II)

31

- ▣ **autoplay:** Si incluimos este atributo la reproducción comenzará automáticamente, sin que haga falta pulsar el botón *play* para iniciar la reproducción.
- ▣ **poster:** Este atributo permite mostrar una imagen mientras el navegador está cargando el vídeo. Su valor será la ruta a la imagen que se quiera mostrar.

# Etiqueta `<audio>`

32

- La etiqueta **`<audio>`** permite incluir la inclusión en el navegador de un archivo de sonido.
- Atributos:
  - ▣ **`src`**: Origen del archivo.
  - ▣ **`autoplay`**: Permite que el archivo se inicie directamente.
  - ▣ **`controls`**: Permite ver la consola de reproducción.
  - ▣ **`loop`**: Permite la reproducción indefinida.





# Validadores

33

- ❑ **TAW:** Test de accesibilidad web. [www.tawdis.net](http://www.tawdis.net).
- ❑ **Validador de código W3C:** El concepto de validación es objeto de controversia en el ámbito del desarrollo web.
- ❑ Por una parte, la validación no es obligatoria y las páginas web se pueden ver bien sin que sean válidas.
- ❑ Por otra parte, una página válida es más correcta que otra página que no lo sea, ya que cumple con las normas y restricciones impuestas por HTML.
- ❑ Hay quien dice que las normas están para cumplirlas y quien dice que no hay que darles demasiada importancia.
- ❑ <http://validator.w3.org>

