

Principales etiquetas HTML



1. Elementos en bloque y en línea

En HTML existen los **elementos inline** (*en línea*) y los **elementos block** (*en bloque*), y ambos hacen referencia a cómo se muestran en el navegador.

1.1 Elementos en línea

Los elementos de tipo inline, como su propio nombre indica, son elementos cuyo contenido se encuentra en una misma línea, de forma que si colocamos dos o más elementos en línea sin indicar expresamente un salto de línea, estos se mostrarán uno seguido del otro en una misma línea.

Algunos elementos de este tipo son: `<a>` (etiqueta para insertar enlaces), `` (etiqueta para insertar imágenes), `<code>` (etiqueta para insertar fragmentos de código), etc.

Ejemplo de elementos en línea

```
<!DOCTYPE html>
<html lang="es">
  <head> ... </head>
  <body>
    <a href="https://iessanvicente.com/"> Enlace al IES</a>
    
    <code>
      Fragmento de código
    </code>
  </body>
</html>
```

El código anterior se mostraría así en el navegador:



Como podemos observar, los elementos se encuentran en la misma línea, esto ocurre porque son elementos *inline*.

1.2 Elementos en bloque

Los elementos de tipo bloque son aquellos que se comportan como bloques, cada vez que se utiliza un elemento de este tipo se hace un salto de línea de manera predeterminada. Si colocamos varios elementos de tipo block seguidos, estos se mostrarán uno debajo del otro de forma automática.

Algunos elementos de este tipos son: `<p>` (etiqueta para empezar un párrafo), `<h1>` (etiqueta de encabezado), `` (etiqueta que indica un elemento de una lista), etc.

Ejemplo de elementos de bloque:

```
<!DOCTYPE html>
<html lang="es">
  <head> ... </head>
  <body>
    <h1> Título de la página </h1>
    <p> Esto es un párrafo en el que se explica cómo funcionan
      los elementos de bloque en HTML. Se trata de elementos
      que incorporan un salto de línea de forma automática.</p>
    <h2> Subtítulo de la página </h2>
    <ol>
      <li> etiqueta de párrafo </li>
      <li> etiqueta de encabezado </li>
      <li> ... </li>
    </ol>
  </body>
</html>
```

El código anterior se mostraría así en el navegador:



2. Etiquetas básicas de texto

Comenzaremos viendo un conjunto de etiquetas que suelen utilizarse mucho para definir el contenido de un documento HTML: párrafos, encabezados, negritas, cursivas y alguna que otra más.

2.1. Encabezados: etiquetas *h1* a *h6*

Se utilizan las etiquetas desde `<h1>` a `<h6>` para definir encabezados de mayor a menor importancia: `<h1>` define el encabezado más importante mientras que `<h6>` es el menos importante. Sólo se debe

utilizar un `<h1>` por página; éste representa el encabezado ó tema principal de toda la página. Además, no conviene omitir los niveles de encabezado: se comenzará con `<h1>`, luego `<h2>`, y así sucesivamente.

- **Tipo:** elemento en bloque

Ejemplo: ejemplo de uso de los seis encabezados, desde el más importante (el más grande) hasta el de menor relevancia (el más pequeño)

```
<!DOCTYPE html>
<html>
  <head>
    <title> Encabezados </title>
    <meta charset="utf-8">
  </head>
  <body>
    <h1> Encabezado 1 </h1>
    <h2> Encabezado 2 </h2>
    <h3> Encabezado 3 </h3>
    <h4> Encabezado 4 </h4>
    <h5> Encabezado 5 </h5>
    <h6> Encabezado 6 </h6>
  </body>
</html>
```

2.2. Párrafos: la etiqueta *p*

Esta etiqueta hace que el texto que está contenido entre su etiqueta de apertura y de cierre sea considerado un párrafo. Observa que al utilizar este elemento, se añade de forma automática un salto de línea.

- **Tipo:** elemento en bloque

Ejemplo: el siguiente ejemplo muestra el uso de un encabezado y varios párrafos.

```
<!DOCTYPE html>
<html lang="es">
  <head> ... </head>
  <body>
    <h1>Generación de texto automático con Lore Ipsum</h1>
    <p>
      Cras laoreet, ipsum ut mattis maximus, nisl augue
      tristique est, ut fringilla dolor nisi quis
      justo. Praesent at condimentum lorem, vitae auctor
      nulla. Duis vitae egestas elit, sed tincidunt
      metus. Sed vestibulum ipsum a purus bibendum, vel
      rhoncus orci venenatis. In convallis blandit
      magna in tempor.
    </p>
    <p>
      Vestibulum ante ipsum primis in faucibus orci luctus
      et ultrices posuere cubilia curae; Integer
      cursus tempor lacus. Proin tempor varius felis tempus
      ullamcorper. In lobortis massa eget odio
      iaculis, sodales ullamcorper nisi varius.
    </p>
  </body>
</html>
```

Generación de texto automático con Lore Ipsum

Cras laoreet, ipsum ut mattis maximus, nisl augue tristique est, ut fringilla dolor nisi quis justo. Praesent at condimentum lorem, vitae auctor nulla. Duis vitae egestas elit, sed tincidunt metus. Sed vestibulum ipsum a purus bibendum, vel rhoncus orci venenatis. In convallis blandit magna in tempor.

Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Integer cursus tempor lacus. Proin tempor varius felis tempus ullamcorper. In lobortis massa eget odio iaculis, sodales ullamcorper nisi varius.

Ejercicio 1:

Modifica el siguiente [código base](#) para que la página web tenga el mismo aspecto que la siguiente imagen; para eso utiliza las diferentes etiquetas de encabezado donde corresponda. Guárdalo como **ejercicio_encabezados.html**.

HTML

Introducción

HTML son las siglas de HyperText Markup Language, que puede traducirse como lenguaje de marcas o marcado de hipertexto.

El lenguaje HTML se emplea para crear las páginas web. Es muy fácil ver el código HTML de una página web, la opción exacta cambia de un navegador a otro y también puede cambiar de una versión a otra de un mismo navegador, pero suelen tener un nombre similar.

Semántica en HTML 5

En HTML 5 se incorporan nuevos elementos que permiten dotar de mayor semántica a los documentos, sobre todo a la hora de estructurar la información o los contenidos. Pasamos de un modelo gobernado por las capas `span` y los bloques `div` donde la semántica se establecía con los elementos CSS, a tener elementos específicos con su semántica.

El esquema de documento

En HTML5 encontramos elementos nuevos en el esquema del documento como `article`, `section`, `aside`, `header`, `footer`, ...

Elementos nuevos

A continuación detallan los elementos nuevos

Article

...

Section

...

2.3. Resaltado de texto: negritas y cursivas

No debemos confundir el hecho de resaltar un texto con aplicar estilos a un documento, cosa que NO debe hacerse con HTML, sino con CSS. El resaltado del texto sirve para hacer énfasis en una parte de dicho texto, atrayendo la atención del lector.

En HTML existen dos tipos principales de resaltado o enfatizado de texto: una consiste en aumentar el grosor (lo que popularmente se conoce como *tipo negrita*) y otra en modificar la inclinación del texto con respecto al que le rodea (lo que se conoce como *tipo cursiva*).

2.3.1. Negritas: etiquetas *strong* y *b*

El elemento `` pone en negrita el texto contenido en él, pero no aporta ningún significado especial, énfasis o importancia a su contenido; su uso está actualmente desaconsejado. En su lugar se recomienda usar el elemento ``, que también muestra el texto en negrita pero además sí implica que el texto es muy importante. Esta información resulta muy útil tanto para los navegadores de voz, que lo leerán dándole el énfasis correspondiente, como para los motores de búsqueda, que realizan un análisis de la página y muestran al usuario resultados de páginas que contengan determinadas palabras clave resaltadas.

- **Tipo:** elemento en línea

Ejemplo: a continuación se muestra un ejemplo sencillo del uso del elemento ``

```
<!DOCTYPE html>
<html lang="es">
  <head> ... </head>
  <body>
    <h1> Ejemplo de <code>strong</code> </h1>
    <p>
      El trasfondo tecnológico de la novela <strong>Criptonomicón</strong>
      es la <strong>criptografía</strong> y las
      <strong>matemáticas</strong> y el uso que se hacen de ellas en las
      dos tramas que se van compaginando a lo largo de la novela y que se
      centran en el marco histórico de la Segunda Guerra Mundial y en un
      presente paralelo al que vivimos actualmente. La principal aplicación
      que se da en la trama histórica será la ruptura de los códigos
      utilizados por los alemanes y japoneses sin que estos se den cuenta,
      mientras que en el presente se intenta crear la Cripta, un paraíso
      informático de datos libres del poder de los gobiernos.
    </p>
  </body>
</html>
```

Ejemplo de strong

El trasfondo tecnológico de la novela **Criptonomicón** es la **criptografía** y las **matemáticas** y el uso que se hacen de ellas en las dos tramas que se van compaginando a lo largo de la novela y que se centran en el marco histórico de la Segunda Guerra Mundial y en un presente paralelo al que vivimos actualmente. La principal aplicación que se da en la trama histórica será la ruptura de los códigos utilizados por los alemanes y japoneses sin que estos se den cuenta, mientras que en el presente se intenta crear la Cripta, un paraíso informático de datos libres del poder de los gobiernos.

2.3.2. Cursivas: etiquetas *em* e *i*

El elemento `<i>` pone en cursiva el texto contenido en él, pero no aporta ningún contenido semántico extra; al igual que el elemento ``, su uso está actualmente desaconsejado. En su lugar, se recomienda usar el elemento `` que también pone en cursiva el texto contenido en él y además permite a los lectores de pantalla pronunciar ese texto con mayor énfasis que el resto y, por tanto, hace más fácil interpretar su significado. Es importante que no utilices esta etiqueta solamente para establecer un estilo de letra cursiva. Para ello es mejor utilizar estilos CSS.

- **Tipo:** elemento en línea

Ejemplo: Ejemplo de uso del elemento ``. Observa que la primera frase suena aliviada porque la persona no llega tarde. Por el contrario, la segunda suena sarcástica y un tanto pasivo-agresiva, expresa molestia porque la persona ha llegado algo tarde.

```
<!DOCTYPE html>
<html lang="es">
  <head> ... </head>
  <body>
    <h1> Ejemplo de <code>em</code></h1>
    <p> 1. Me alegro de que no llegues tarde </p>
    <p> 2. Me <em>alegro</em> de que no llegues <em>tarde</em> </p>
  </body>
</html>
```

Ejemplo de em

1. Me **alegro** de que no llegues tarde
2. Me *alegro* de que no llegues *tarde*

Ejercicio 2:

Crea una página web llamada **curriculum.html** que tenga lo siguiente:

- Como título de la página (etiqueta *title* dentro de *head*), ponle tu nombre
- En el cuerpo (*body*), deberá tener:
 - Un encabezado principal (*h1*) con el texto "Mi curriculum".
 - Un encabezado secundario (*h2*) llamado "Datos personales", donde deberás indicar en uno o dos párrafos información sobre ti: nombre y apellidos, dirección y teléfono (te puedes inventar la dirección y el teléfono si quieres).
 - Otro encabezado secundario (*h2*) llamado "Formación", que tendrá dentro dos subapartados (*h3*): uno llamado "Idiomas" donde deberás indicar los idiomas que puedes hablar o escribir correctamente (en uno o dos párrafos), y otro llamado "Estudios" donde también en uno o dos párrafos indiques los estudios que has alcanzado hasta ahora, incluyendo tus estudios actuales (graduado de ESO, Bachillerato, Ciclos Formativos, etc.)
 - Utiliza negritas y/o cursivas para resaltar las partes del currículum que consideres como palabras clave.

2.4. La etiqueta *span*

La etiqueta `` se usa cuando queremos marcar un texto pero consideramos que ninguna de las etiquetas existentes es una buena opción. Con `` podemos aislar un fragmento de texto sin introducir cambios en la apariencia del texto marcado. La ventaja de esta etiqueta se aprecia al usarla junto con algunos atributos tales como `class` o `style`, combinados con algún código CSS. Trabajaremos en esto más adelante.

- **Tipo:** elemento en línea

Ejemplo: como hemos comentado, veremos más adelante CSS pero de momento vamos a ver un ejemplo donde se usa un elemento `` para colorear solo una parte del texto dentro de un párrafo de color azul.

```
<!DOCTYPE html>
<html lang="es">
  <head> ... </head>
  <body>
    <h1> Ejemplo de <span> </h1>

    <p>Mi madre tiene los ojos <span style="color:blue"> azules </span>.</p>
  </body>
</html>
```

Ejemplo de span

Mi madre tiene los ojos azules .

2.5. Algunos símbolos especiales

¿Cómo podríamos representar en una página web el símbolo `<`? Lo utilizamos para las etiquetas, y en el momento en que lo pongamos, el navegador va a creer que estamos empezando una etiqueta y no lo mostrará. Al igual que ese símbolo, o su contrario `>`, existen ciertos símbolos que, cuando los escribimos directamente en el documento, no se representan bien (o directamente, no se representan) en la página, porque el navegador no los reconoce como símbolos imprimibles. En su lugar, debemos escribir un código alternativo.

Aquí vemos un resumen de algunos símbolos útiles:

Símbolo	Código
<code><</code>	<code>&lt;</code>
<code>></code>	<code>&gt;</code>
<code>&</code>	<code>&amp;</code>
<code>"</code>	<code>&quot;</code>
<code>'</code>	<code>&apos;</code>
(espacio)	<code>&nbsp;</code>

Respecto a los espacios, aunque si ponemos un espacio entre palabra y palabra, se va a representar bien, en cuanto queramos utilizar el espacio como elemento indentador (por ejemplo, poner un texto cinco espacios hacia dentro) ya no va a ser posible. En estos casos tendremos que utilizar el código del espacio (` `) y repetirlo tantas veces como espacios queramos poner.

entre la etiqueta `<div>` y su correspondiente etiqueta de cierre `</div>`. En principio, los navegadores no muestran nada especial cuando se crea una sección con `<div>`, salvo que se dé formato al elemento con una hoja de estilo CSS. La forma de distinguir una sección de otra se realiza utilizando los atributos *class* ó *id*, que se explicarán más adelante en el temario.

La etiqueta `<div>`, como hemos comentado antes, se usa en HTML 4 y XHTML 1.0 para crear agrupaciones de contenidos pero este uso ha quedado obsoleto, y en su lugar se emplean unas nuevas etiquetas con significado semántico que se han introducido en HTML 5 (las estudiaremos en los siguientes apartados). No obstante, esta etiqueta se mantiene en HTML 5 pero queda reservada para agrupaciones no consideradas por las nuevas etiquetas.

- **Tipo:** elemento en bloque

Ejemplo: el siguiente ejemplo muestra una página típica HTML4 / XHTML 1.0, en la que se utilizan varias divisiones `<div>` para organizar las diferentes partes del documento.

```
<html>
  ...
  <body>
    <div id="cabecera">
      ...
    </div>

    <div id="menu">
      ...
    </div>

    <div id="contenido">
      <div class="apartado">
        ...
      </div>
      <div class="apartado">
        ...
      </div>
    </div>

    <div id="pie">
      ...
    </div>
  </body>
</html>
```

Ejercicio 3:

Piensa en el diseño de una página personal cualquiera y en las diferentes secciones de las que constaría. Posteriormente, plasma ese diseño en un documento web. A modo de ejemplo, se describe a continuación las secciones que podría tener una página personal tradicional:

- Debe tener un título con el nombre de la persona, un menú de navegación con diferentes enlaces a otras secciones o páginas: cursos, idiomas, experiencia,... (no tienes que crear el enlace, solo el texto; veremos los enlaces más adelante), un contenido principal (con una presentación básica de la persona y/o de sus aficiones), un contenido secundario con recomendaciones de compañeros, y por último un pie de página en el que se nombren sus redes sociales.

Crea con VScode un nuevo documento HTML que recoja la estructura anterior, y llámalo **pagina_divs.html**. Utiliza el elemento `<div>` para las diferentes secciones y añade el atributo `id` si esa sección es única o el atributo `class` si puede haber varias de esas secciones, seguido del nombre de la sección. Dentro de cada sección incluye un contenido de ejemplo con párrafos o encabezados. Recuerda añadir el título y las metaetiquetas vistas hasta ahora.

Guarda la página y prueba a abrirla con un navegador para comprobar cómo se ve.

3.3. El elemento *section*

Esta etiqueta se utiliza para distinguir diferentes áreas de funcionalidad o áreas temáticas. Es un elemento bastante genérico, pero aún así tiene más significado semántico que el elemento `<div>`. Normalmente las secciones empiezan con un título (`<h1>`, `<h2>`, ...) que hace referencia al tema tratado en la sección. Un elemento `<section>` puede contener a su vez otros elementos `<section>`. Las secciones interiores se entienden como subsecciones de la sección exterior. Además, un elemento `<section>` puede contener elementos `<article>` (los veremos a continuación) y viceversa.

- **Tipo:** elemento de bloque

Ejemplo: documento web con dos secciones temáticas bien diferenciadas.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Asociación astronómica</title>
    <meta charset="utf-8">
  </head>
  <body>
    <h1>Asociación astronómica de San Vicente del Raspeig</h1>

    <section>
      <h2> Noticias de actualidad </h2>
      ...
    </section>

    <section>
      <h2> Astrofotografía </h2>
      ...
    </section>
  </body>
</html>
```

3.4. El elemento *article*

Este elemento se utiliza para incluir contenido independiente; debe tener sentido por sí mismo. Por ejemplo, una noticia, las entradas de un blog o un vídeo. No debemos confundir este elemento con `<section>`. Mientras que `<section>` agrupa distintas secciones de contenido o funcionalidad, `<article>` contiene contenido independiente, con sentido para distribuirlo de forma independiente (en una fuente RSS, por ejemplo).

Como hemos comentado en el apartado anterior, también es posible anidar secciones dentro de los artículos, siempre y cuando tenga sentido hacerlo. Por ejemplo, si cada una de las entradas del blog tiene una estructura coherente de secciones distintas (por ejemplo, una introducción, un desarrollo y una conclusión), entonces se pueden poner secciones dentro de los artículos.

- **Tipo:** elemento de bloque

Ejemplo: supongamos que tenemos un blog; cada entrada del blog se puede distribuir como elemento en una fuente de noticias RSS y tendría sentido cuando se lee individualmente, por tanto, `<article>` es perfecto en este caso.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Blog de tecnología</title>
    <meta charset="utf-8">
  </head>
  <body>
    <section>
      <article>
        <!-- aquí va el contenido de la primera entrada del blog -->
      </article>

      <article>
        <!-- aquí va el contenido de la segunda entrada del blog -->
      </article>
      ...
    </section>
    ...
  </body>
</html>
```

3.5. El elemento *aside*

El elemento `<aside>` se usa para añadir contenido secundario, como por ejemplo un bloque de anuncios, un grupo de enlaces externos relacionados, una cita de texto, etc. Normalmente se suele mostrar en los laterales del documento.

Este elemento puede estar incluido directamente en el `<body>`, haciendo referencia a contenido secundario con respecto a la página en su conjunto, pero también se puede incluir en `<article>` o `<section>`, en cuyo caso se trata de contenido secundario con respecto al elemento que lo contiene.

- **Tipo:** elemento en bloque

Ejemplo: en este ejemplo vamos a añadir, en un lateral, un bloque de anuncios relacionados con la compraventa de productos de astronomía.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Asociación astronómica</title>
    <meta charset="utf-8">
  </head>
  <body>
    <h1>Asociación astronómica de San Vicente del Raspeig</h1>
    <section>
      <h2> Noticias de actualidad </h2>
      ...
    </section>

    <section>
      <h2> Astrofotografía </h2>
      ...
    </section>

    <aside>
      <!-- aquí podemos poner un bloque de anuncios de material astronómico -->
    </aside>
  </body>
</html>
```

3.6. El elemento *header*

Este elemento se utiliza en bloques de contenidos que necesiten tener su propio encabezado; puede hacer referencia a contenido introductorio de todo el documento o de una sección concreta del mismo, por lo que puede haber varios elementos `<header>` en un documento HTML. Sin embargo, no se puede colocar dentro de un `<footer>`, `<address>` u otro elemento `<header>`.

Suele contener elementos de encabezado (h1, h2, ...), así como también un logo ó un formulario de búsqueda. NO confundir con el elemento `<head>`. Mientras que el `<head>` se emplea fuera del cuerpo del documento (`<body>`) para definir los metadatos del documento, `<header>` contiene información introductoria o de navegación dentro del cuerpo del documento.

- **Tipo:** elemento en bloque

Ejemplo: se usa un elemento `<header>` para agrupar un logo y un nombre de empresa; en este caso se trata de contenido introductorio que afecta a todo el documento.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Asociación astronómica</title>
    <meta charset="utf-8">
  </head>
  <body>
    <header>
      
      <h1>Asociación astronómica de San Vicente del Raspeig</h1>
    </header>
    ...
  </body>
</html>
```

3.7. El elemento *nav*

El elemento `<nav>` se usa para agrupar los bloques de enlaces principales de navegación, bien enlaces a otras páginas o enlaces a los apartados de la propia página. Una página puede contener varios elementos `<nav>`.

- **Tipo:** elemento en bloque

Ejemplo: el siguiente ejemplo muestra una página con dos elementos `<nav>`. El `<nav>` del `<header>`, con enlaces globales a otras páginas del sitio web y el `<nav>` del `<body>`, con enlaces a las secciones de la página principal.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Asociación astronómica</title>
    <meta charset="utf-8">
  </head>
  <body>
    <header>
      
      <h1>Asociación astronómica de San Vicente del Raspeig</h1>
      <nav>
        <!-- aquí van los enlaces a otras páginas del sitio web -->
      </nav>
    </header>

    <nav>
      <!-- aquí van los enlaces a las diferentes secciones
      de la página principal -->
    </nav>

    <section>
      <h2> Noticias de actualidad </h2>
      ...
    </section>

    <section>
      <h2> Astrofotografía </h2>
      ...
    </section>
  </body>
</html>
```

3.8. El elemento *footer*

Este elemento se usa para añadir información general sobre la página: autor, dirección de contacto, licencia o condiciones de uso, etc. Se suele poner al final del documento. Puede estar incluido en `<body>` y entonces se entiende que es el pie de página de la página en su conjunto, pero también se puede incluir en `<article>`, `<section>`, `<nav>` y `<aside>` y entonces se entiende que es el pie de página del elemento que lo contiene.

- **Tipo:** elemento en bloque

Ejemplo: se añade al ejemplo anterior un elemento `<footer>` con el autor de la página y su correo electrónico.


```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Asociación astronómica</title>
    <meta charset="utf-8">
  </head>
  <body>
    <header>
      
      <h1>Asociación astronómica de San Vicente del Raspeig</h1>
    </header>

    <nav>
      <!-- aquí van los enlaces a las diferentes secciones
           de la página principal -->
    </nav>

    <section>
      <h2> Noticias de actualidad </h2>
      ...
    </section>

    <section>
      <h2> Astrofotografía </h2>
      ...
    </section>

    <aside>
      <!-- aquí podemos poner un bloque de anuncios de material astronómico -->
    </aside>

    <footer>
      <p>May Calle</p>
      <a href="mailto:maycalle@iessanvicente.com">
        maycalle@iessanvicente.com
      </a>
    </footer>
  </body>
</html>
```

3.9. El elemento *address*

Este es un elemento específico para añadir información de contacto del autor o propietario de la página ó artículo (dirección de correo, teléfono de contacto, redes sociales, etc.). Normalmente se incluye en el `<footer>` de `<body>` ó en el `<footer>` de un `<article>`. Generalmente se muestra por defecto en cursiva.

- **Tipo:** elemento en bloque

Ejemplo: en este ejemplo se modifica el footer anterior y se añade la información de contacto con la etiqueta específica `<address>`

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Asociación astronómica</title>
    <meta charset="utf-8">
  </head>
  <body>
    ...
    <footer>
      <address>
        Autor: May Calle
        Correo:
        <a href="mailto:maycalle@iessanvicente.com">
          maycalle@iessanvicente.com
        </a>
      </address>
      <p> Última modificación: 1 de septiembre de 2021</p>
    </footer>
  </body>
</html>
```

3.10. El elemento *main*

El elemento *main* se usa para identificar el bloque de contenido principal. Este elemento resulta de especial interés en el caso de usar navegadores de voz. En ese caso, el navegador puede ir directamente al contenido principal, saltando toda la información previa menos relevante.

Solo puede haber un elemento `<main>` en el documento, y solo puede estar contenido por `<body>`, `<div>` y `<form>` (etiqueta para definir formularios que veremos más adelante); no debe estar contenido en ningún `<section>`, `<article>`, `<aside>`, `<footer>`, `<header>` ó `<nav>`.

- **Tipo:** elemento en bloque

Ejemplo: en este ejemplo se muestra el uso del elemento `<main>`, en el que se incluyen las secciones principales del documento.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Asociación astronómica</title>
    <meta charset="utf-8">
  </head>
  <body>
    <header>
      
      <h1>Asociación astronómica de San Vicente del Raspeig</h1>
    </header>

    <nav>
      <!-- aquí van los enlaces a las diferentes secciones
           de la página principal -->
    </nav>

    <main>
      <section>
        <h2> Noticias de actualidad </h2>
        ...
      </section>

      <section>
        <h2> Astrofotografía </h2>
        ...
      </section>
    </main>
  </body>
</html>
```

Ejercicio 4:

Modifica el *Ejercicio 3* anterior de forma que en lugar de utilizar el elemento `<div>` se utilicen las etiquetas semánticas de estructura de HTML 5 vistas hasta ahora. Guarda el documento como **pagina_divs_html5.html**.

Como guía, se muestra a continuación cómo quedaría el ejemplo visto en el apartado 3.2 en HTML 5:

```
<html>
  ...
  <body>
    <header>
      ...
    </header>

    <nav>
      ...
    </nav>

    <main>
      <section>
        ...
      </section>
      <section>
        ...
      </section>
    </main>

    <footer>
      ...
    </footer>
  </body>
</html>
```

4. Definición de listas

En HTML existen tres tipos principales de listas:

- Listas ordenadas, donde los elementos tienen un orden, que puede venir dado por números arábigos, letras, números romanos... dependiendo de cómo configuremos el estilo de la lista.
- Listas no ordenadas, donde los elementos no muestran un orden particular, más allá del orden en que se enumeran. En este caso se emplean viñetas gráficas, tales como puntos, cuadrados, etc.
- Listas de definición, en las que aparecen un término y su definición asociada, al estilo de las entradas de un diccionario.

4.1. Listas ordenadas: etiquetas *ol* y *li*

El elemento `` se utiliza para definir listas ordenadas de items; puede ser una lista numérica o alfabética. Se utiliza el elemento `` para definir cada elemento de la lista; `` debe aparecer anidado al elemento `` (es decir, antes de la etiqueta de cierre ``)

- **Tipo:** elemento en bloque

Se puede personalizar la lista a través de los siguientes atributos:

- **reversed**: se trata de un atributo booleano (toma el valor de *verdadero* o *falso*); si está presente, indica que la lista es descendente (... 3, 2, 1) mientras que si se omite, indica que la lista es ascendente (1, 2, 3, ...).
- **start**: se utiliza para determinar el valor inicial de la lista; debe ser un número entero válido.
- **type**: indica el tipo de viñeta/marcador que se utilizará en la lista: `1` (enteros), `A` (letras mayúsculas), `a` (letras minúsculas), `I` (números romanos en mayúsculas), `i` (números romanos en minúscula), ...

Si no indicamos ninguno de los atributos anteriores, se crea una lista decimal ascendente cuyo valor inicial es 1. Veremos más adelante que también es posible usar CSS para personalizar las listas.

Ejemplo: en el siguiente ejemplo se muestran cuatro listas ordenadas; el primer ejemplo es una lista ordenada por defecto (con números enteros y ascendente), el segundo emplea el atributo `start` para empezar la lista con el número 5 en lugar del 1, el tercero emplea los números romanos para listar los items y en el último se crea una lista ordenada en orden descendente.

```
<!DOCTYPE html>
<html lang="es">
  <head> ... </head>
  <body>
    <h1> Listas ordenadas </h1>

    <p> Procedimiento para acceder a la BIOS: </p>
    <ol>
      <li> Enciende el ordenador </li>
      <li> Pulsa la tecla DEL cuando se indique </li>
      <li> Introduce la contraseña </li>
      <li> ... </li>
    </ol>

    <p> Ahora continúa con las siguientes instrucciones: </p>
    <ol start="5">
      <li> Presiona la tecla F10 </li>
      <li> Confirma la salida </li>
    </ol>

    <p> ... presta especial atención al apartado III. </p>
    <ol type="I">
      <li> Introduccción a HTML </li>
      <li> Elementos básicos de HTML </li>
      <li> Otros elementos de HTML </li>
    </ol>

    <p> Estas son mis 3 películas favoritas (ordenadas de menos a más): </p>
    <ol reversed>
      <li> First Man </li>
      <li> Contact </li>
      <li> Apollo 13 </li>
    </ol>

  </body>
</html>
```

Listas ordenadas

Procedimiento para acceder a la BIOS:

1. Enciende el ordenador
2. Pulsa la tecla DEL cuando se indique
3. Introduce la contraseña
4. ...

Ahora continua con las siguientes instrucciones:

5. Presiona la tecla F10
6. Confirma la salida

... presta especial atención al apartado III.

- I. Introducción a HTML
- II. Elementos básicos de HTML
- III. Otros elementos de HTML

Estas son mis 3 películas favoritas (ordenadas de menos a más):

3. First Man
2. Contact
1. Apollo 13

4.2. Listas no ordenadas: etiquetas *ul* y *li*

El elemento `` representa una lista no ordenada de ítems. Al igual que en el caso de las listas ordenadas, cada elemento de la lista se indica con el elemento ``, que debe aparecer antes de la etiqueta de cierre ``. Por defecto, aparece un disco negro como viñeta, pero se puede modificar utilizando el atributo `type` y alguno de los siguiente valores: `disc` (círculo relleno), `circle` (círculo sin relleno) ó `square` (cuadrado relleno). Sin embargo, este atributo se ha vuelto obsoleto en HTML5 y se recomienda usar en su lugar CSS.

- **Tipo:** elemento en bloque

Ejemplo: a continuación se muestra un ejemplo con dos listas no ordenadas; en la primera no se utiliza el atributo `type` mientras que en la segunda sí, aunque su uso no está recomendado.

```
<!DOCTYPE html>
<html lang="es">
  <head> ... </head>
  <body>
    <h1> Listas no ordenadas </h1>

    <p> Ingredientes para creps: </p>
    <ul>
      <li> 1 vaso de harina de trigo </li>
      <li> 2 vasos de leche </li>
      <li> 3 huevos </li>
      <li> Una pizca de sal </li>
    </ul>

    <p> Ingredientes para creps: </p>
    <ul type="square">
      <li> 1 vaso de harina de trigo </li>
      <li> 2 vasos de leche </li>
      <li> 3 huevos </li>
      <li> Una pizca de sal </li>
    </ul>
  </body>
</html>
```

Listas no ordenadas

Ingredientes para creps:

- 1 vaso de harina de trigo
- 2 vasos de leche
- 3 huevos
- Una pizca de sal

Ingredientes para creps:

- 1 vaso de harina de trigo
- 2 vasos de leche
- 3 huevos
- Una pizca de sal

4.3. Listas anidadas

Las listas se pueden anidar, es decir, podemos incluir una lista dentro de otra y cada una tendrá su propia numeración o viñeta. Para anidarlas, crearemos otra lista (con `` o ``) dentro de una etiqueta ``, y cerraremos `` después de cerrar previamente la definición de la sublista. Veamos un ejemplo:


```
<!DOCTYPE html>
<html lang="es">
  <head> ... </head>
  <body>
    <h1> Listas anidadas </h1>
    <ol>
      <li> Opción 1 </li>
      <li> Opción 2 <!-- La etiqueta de cierre no va aquí -->
        <ul>
          <li> Opción 2A </li>
          <li> Opción 2B </li>
          <li> Opción 2C </li>
        </ul>
      </li> <!-- Aquí está la etiqueta de cierre -->
      <li> Opción 3 </li>
      <li> Opción 4 </li>
    </ol>
  </body>
</html>
```

Listas anidadas

1. Opción 1
2. Opción 2
 - Opción 2A
 - Opción 2B
 - Opción 2C
3. Opción 3
4. Opción 4

Ejercicio 5:

Crema una página llamada **listas.html** y trata de representar un contenido como éste:

Saga Harry Potter

1. Harry Potter y la piedra filosofal
2. Harry Potter y la cámara secreta
3. Harry Potter y el prisionero de Azkaban
4. ...

Asignaturas

- Implantación de Aplicaciones Web
- Administración de S.O.
- Servicios en red
- ... etc.

Temario

1. Introducción a las Aplicaciones Web
2. Configuración del entorno
3. Diseño web con HTML y CSS
 - Estructura de un documento HTML
 - Textos básicos y encabezados
 - Estilos básicos de texto
 - El modelo de cajas...
 - ...
4. Javascript
5. ...

4.4. Listas de descripción: etiquetas *dl*, *dt* y *dd*

Las listas de descripción son listas donde se asocian términos con descripciones o valores, por ejemplo, glosarios, listas de vocabulario, listas de metadatos ó listas de preguntas y respuestas; no es adecuado para representar diálogos.

Para representar o definir estas listas utilizamos el elemento `<dl>`, donde cada término está envuelto en un elemento `<dt>` (término de descripción) y cada descripción está envuelta en un elemento `<dd>` (definición de descripción). Se puede asociar un término con múltiples descripciones y viceversa. El navegador muestra el texto contenido en `<dd>` con una sangría.

Opcionalmente, un grupo compuesto por cualquier número de términos seguidos por cualquier número de descripciones puede estar contenido en un elemento `<div>`. Esto es útil para incorporar atributos o información de estilo común al grupo.

- **Tipo:** elemento en bloque

Ejemplo: en el siguiente ejemplo, se muestra cómo se define un glosario de términos y un ejemplo de preguntas y respuestas. Observa que en el glosario de términos la tercera definición está asociada a dos términos.

```
<!DOCTYPE html>
<html lang="es">
  <head> ... </head>
  <body>
    <h1> Listas de descripción </h1>

    <p> <strong> Glosario de términos informáticos </strong> </p>
    <dl>
      <dt>HTML</dt>
      <dd>Lenguaje de marcado para la elaboración de páginas web</dd>

      <dt>Javascript</dt>
      <dd>Lenguaje de programación interpretado que se utiliza para el
        desarrollo de aplicaciones web tanto en la parte cliente como
        en la parte de servidor. </dd>

      <dt>Interfaz</dt>
      <dd>Dispositivo capaz de transformar las señales generadas por un
        aparato en señales comprensibles por otro.</dd>
      <dd>En programación orientada a objetos, elemento que proporciona
        un esqueleto común de métodos a implementar por varias clases.</dd>
    </dl>

    <p> <strong> Entrevista a Chema Alonso </strong> </p>
    <dl>
      <dt>¿Qué motivo te impulsó para comenzar con el mundo del hacking
        ético?</dt>
      <dd> Entrar en el mundo del hacking fue casi por casualidad. Con 12
        años comencé con los ordenadores a programar en BASIC porque me
        enamoré de la tecnología con la película de TRON, y luego juegos
        de guerra, y muchas otras películas de los años 80.</dd>
      <dt>¿Qué tendencias a nivel de seguridad informática marcarán los
        próximos años?</dt>
      <dd>Creo que desde el punto de vista tecnológico, la investigación más
        puntera será, sin duda, todo lo referido a la Inteligencia Artificial
        aplicada a la ciberseguridad.</dd>
    </dl>
  </body>
</html>
```

Listas de descripción

Glosario de términos informáticos

HTML

Lenguaje de marcado para la elaboración de páginas web

Javascript

Lenguaje de programación interpretado que se utiliza para el desarrollo de aplicaciones web tanto en la parte cliente como en la parte de servidor.

Interfaz

Dispositivo capaz de transformar las señales generadas por un aparato en señales comprensibles por otro.

En programación orientada a objetos, elemento que proporciona un esqueleto común de métodos a implementar por varias clases.

Entrevista a Chema Alonso

¿Qué motivo te impulsó para comenzar con el mundo del hacking ético?

Entrar en el mundo del hacking fue casi por casualidad. Con 12 años comencé con los ordenadores a programar en BASIC porque me enamoré de la tecnología con la película de TRON, y luego juegos de guerra, y muchas otras películas de los años 80.

¿Qué tendencias a nivel de seguridad informática marcarán los próximos años?

Creo que desde el punto de vista tecnológico, la investigación más puntera será, sin duda, todo lo referido a la Inteligencia Artificial aplicada a la ciberseguridad.

5. Otras etiquetas

Para terminar este primer bloque introductorio de etiquetas, aquí incluimos algunas etiquetas más que pueden tener su utilidad en ciertos tipos de documentos.

5.1. Citas: etiquetas *blockquote* y *q*

Estas dos etiquetas se utilizan para citar texto de otra fuente. La diferencia entre ellas radica en que mientras `<blockquote>` se utiliza para insertar citas largas formadas por uno ó varios párrafos ó un conjunto de otros muchos elementos incluyendo imágenes, tablas, etc., `<q>` se utiliza para citar breves líneas de texto. Otra diferencia es que, por defecto, los navegadores suelen aplicar una sangría a los elementos `<blockquote>` mientras que a los elementos `<q>` se les añaden comillas alrededor del contenido.

Adicionalmente, el recurso o URL de donde se ha extraído la cita puede ser especificado en el atributo `cite`. No confundir con el elemento HTML `<cite>`, que se utiliza para indicar el título de una obra creativa (un libro, un poema, una canción, una película, una pintura, una escultura, ...); el nombre de una persona no es el título de una obra.

- **Tipo:** `<blockquote>` es un elemento en bloque; `<q>` es un elemento en línea.

Ejemplo: a continuación se muestran un par de ejemplos de uso de los elementos `<blockquote>` y `<q>`, en los que se emplea el atributo `cite` para indicar la página web de donde se ha extraído la cita.

```

<!DOCTYPE html>
<html lang="es">
  <head> ... </head>
  <body>
    <p> Para los fans de Star Wars...
    <blockquote
      cite="https://starwars.fandom.com/es/wiki/Star_Wars_Wiki:Citas:Han_Solo">
      He recorrido esta galaxia de un extremo a otro, he visto cosas muy raras,
      pero nunca vi nada que me impulsara a creer que haya una única fuerza
      poderosa que lo controla todo. Ningún campo de energía mística controla
      mi destino. Todo eso no son más que leyendas y tonterías.
    </blockquote> - Han Solo
  </p>
</body>
</html>

```

Para los fans de Star Wars...

He recorrido esta galaxia de un extremo a otro, he visto cosas muy raras, pero nunca vi nada que me impulsara a creer que haya una única fuerza poderosa que lo controla todo. Ningún campo de energía mística controla mi destino. Todo eso no son más que leyendas y tonterías.

- Han Solo

```

<!DOCTYPE html>
<html lang="es">
  <head> ... </head>
  <body>
    <p> Según Alan Turing: <q cite="https://agenciab12.com/noticia/alan-turing-10-frases-celebres-padre-inteligencia-artificial">
    Un hombre provisto de papel, lápiz y goma, y con sujeción a una disciplina
    estricta, es, en efecto, una máquina de Turing universal.</q>
  </p>
</body>
</html>

```

Según Alan Turing: “ Un hombre provisto de papel, lápiz y goma, y con sujeción a una disciplina estricta, es, en efecto, una máquina de Turing universal ”

5.2. Texto preformateado: etiquetas *pre* y *code*

Estas etiquetas se utilizan para mostrar un texto preformateado; el texto se mostrará exactamente como está escrito en el código fuente HTML. Esto puede resultar útil en determinados casos ya que los navegadores, por

defecto, no muestran los espacios en blanco o saltos de línea que introducimos en un documento, a menos que lo indiquemos explícitamente con ` ` ó `
` respectivamente.

El elemento `<pre>` se suele utilizar para mostrar fragmentos de código, texto plano o arte ASCII. En algunos casos, es acompañado por otros elementos como `<code>`, `<samp>` o `<kbd>`, los cuales no soportan etiquetas anidadas y, por tanto, si se usa de forma combinada con el elemento `<pre>` éstos deben aparecer anidados a este elemento y no al revés.

- **Tipo:** elemento en bloque

Ejemplo: a continuación se muestran un par de ejemplos; en el primero se muestra la diferencia entre usar `<pre>` y no usarlo, y en el segundo se utilizan `<pre>` y `<code>` para mostrar un fragmento de código. Aparentemente `<pre>` y `<code>` pueden parecer iguales, pero a nivel semántico no lo son. Con `<code>` se indica de forma específica que el texto contenido en él es código en algún lenguaje de programación o marcado, y con `<pre>` se resalta que los espacios y saltos de línea son importantes.

```
<!DOCTYPE html>
<html lang="es">
  <head> ... </head>
  <body>
    <h1> Poema de Gloria Fuertes (sin usar <code>pre</code>) </h1>
    <p>
      En las noches claras,
      resuelvo el problema de la soledad del ser.
      Invito a la luna y con mi sombra somos tres.
    </p>

    <h1> Poema de Gloria Fuertes (usando <code>pre</code>) </h1>
    <pre>
      En las noches claras,
      resuelvo el problema de la soledad del ser.
      Invito a la luna y con mi sombra somos tres.
    </pre>
  </body>
</html>
```

Poema de Gloria Fuertes (sin usar pre)

En las noches claras, resuelvo el problema de la soledad del ser. Invito a la luna y con mi sombra somos tres.

Poema de Gloria Fuertes (usando pre)

```
En las noches claras,
resuelvo el problema de la soledad del ser.
Invito a la luna y con mi sombra somos tres.
```

```

<!DOCTYPE html>
<html lang="es">
  <head> ... </head>
  <body>
    <h1> Ejemplo de programa en JAVA (usando <code>p</code>) </h1>
    <p>
      public class HolaMundo {
        public static void main (String[] args) {
          System.out.println("Hola mundo");
        }
      }
    </p>

    <h1> Ejemplo de programa en JAVA (usando <code>pre</code>) </h1>
    <pre>
      <code>
        public class HolaMundo {
          public static void main (String[] args) {
            System.out.println("Hola mundo");
          }
        }
      </code>
    </pre>

    <h1> Ejemplo de programa en JAVA (usando <code>code</code>) </h1>
    <code>
      public class HolaMundo {
        public static void main (String[] args) {
          System.out.println("Hola mundo");
        }
      }
    </code>
  </body>
</html>

```

Ejemplo de programa en JAVA (usando p)

```
public class HolaMundo { public static void main (String[] args) { System.out.println("Hola mundo"); } }
```

Ejemplo de programa en JAVA (usando pre)

```

public class HolaMundo {
  public static void main (String[] args) {
    System.out.println("Hola mundo");
  }
}

```

Ejemplo de programa en JAVA (usando code)

```
public class HolaMundo { public static void main (String[] args) { System.out.println("Hola mundo"); } }
```

5.3. Subíndices y superíndices: etiquetas *sup* y *sub*

Los elementos `<sup>` y `<sub>` se utilizan para mostrar un superíndice o subíndice respectivamente. Se suele utilizar en fórmulas y ecuaciones matemáticas.

- **Tipo:** elemento en línea

Ejemplo: ejemplo de uso de los elementos `<sup>` y `<sub>`

```
<!DOCTYPE html>
<html lang="es">
  <head> ... </head>
  <body>
    <h1> Ejemplo de <code>sup</code> y <code>sub</code></h1>

    <p> Una de mis moléculas favoritas es
      C<sub>8</sub>H<sub>10</sub>N<sub>4</sub>O<sub>2</sub>,
      también conocida como "cafeína".
    </p>

    <p> El teorema de Pitágoras es: a<sup>2</sup> + b<sup>2</sup> = c<sup>2</sup>:
  </body>
</html>
```

Ejemplo de sup y sub

Una de mis moléculas favoritas es C₈H₁₀N₄O₂, también conocida como "cafeína".

El teorema de Pitágoras es: a² + b² = c²

5.4. Marcado de texto: etiqueta *mark*

Esta etiqueta se utiliza para resaltar partes de un texto como referencia o anotación, debido a su relevancia o importancia en un contexto particular; en general, los navegadores muestran el texto en negro y un color de fondo amarillo.

- **Tipo:** elemento en línea

Ejemplo:


```

<!DOCTYPE html>
<html lang="es">
  <head> ... </head>
  <body>
    <h1> Ejemplo de <code>mark</code> </h1>

    <h2> Récords de Super Mario </h2>

    <p> Mario Bros, el fontanero italiano, es el personaje de videojuegos
    más famoso del mundo. La primera vez que Mario apareció fue en
    <mark>1981</mark> en el juego virtual <mark>Donkey Kong</mark>,
    llamándose Jumpman (saltador, en español). </p>

    <p> Dentro del universo de Mario Bros, el videojuego <mark>Super Mario
    Bros para NES</mark> de 1985 es el más vendido con más de
    <mark>40,24 millones de copias vendidas</mark> a principios
    del 2019.</p>

    <p> En la categoría de speed-runners, los más rápidos en completar un
    videojuego, en octubre del 2018, el usuario <mark>"somewes"</mark>
    consiguió el menor tiempo en completar Super Mario Bros,
    fijando el récord en <mark>4 minutos y 55 segundos.</mark></p>
  </body>
</html>

```

Ejemplo de mark

Récords de Super Mario

Mario Bros, el fontanero italiano, es el personaje de videojuegos más famoso del mundo. La primera vez que Mario apareció fue en 1981 en el juego virtual Donkey Kong, llamándose Jumpman (saltador, en español).

Dentro del universo de Mario Bros, el videojuego Super Mario Bros para NES de 1985 es el más vendido con más de 40,24 millones de copias vendidas a principios del 2019.

En la categoría de speed-runners, los más rápidos en completar un videojuego, en octubre del 2018, el usuario "somewes" consiguió el menor tiempo en completar Super Mario Bros, fijando el récord en 4 minutos y 55 segundos.

5.5. Borrados e inserciones: etiquetas *del* e *ins*

El elemento `` se utiliza para definir un texto que se ha eliminado de un documento. Los motores de búsqueda a veces usan esta etiqueta para mostrar los cambios dentro de un sitio web, cuando se ha eliminado parte del texto después de que el desarrollador haya subido una nueva versión.

Por otro lado, el elemento `<ins>` se utiliza para definir un texto que se ha insertado en un documento. Los motores de búsqueda a veces usan esta etiqueta para mostrar los cambios dentro de una página web, cuando se reemplaza algún texto después de que el desarrollador haya subido una nueva versión.

Depende del navegador cómo mostrar el texto dentro de estos elementos, pero la mayoría de ellos marcarán una línea sobre el texto eliminado y subrayarán el texto insertado de forma predeterminada.

- **Tipo:** elemento en línea

Ejemplo: ejemplo que muestra varios párrafos donde anteriormente se dijeron algunas palabras o frases, pero ahora los autores han eliminado dicho texto o lo han ampliado y quieren que se vea claramente reflejado en el documento.

```
<!DOCTYPE html>
<html lang="es">
  <head> ... </head>
  <body>
    <h1> Ejemplo de <code>del</code> e <code>ins</code> </h1>

    <p> La Tierra es <del>plana</del> <ins>redonda.</ins></p>

    <p> El 31 de diciembre de 2021 <del>se acabará el mundo</del>
      <ins>no pasará nada<ins>. </p>
  </body>
</html>
```

Ejemplo de del e ins

La Tierra es ~~plana~~ redonda.

El 31 de diciembre de 2021 ~~se acabará el mundo~~ no pasará nada.

5.6. Etiquetas delimitadoras: *br* y *hr*

La etiqueta `
` es un elemento vacío (no necesita de etiqueta de cierre) que produce un salto de línea en el texto; es útil para escribir un poema o una dirección, donde la división de las líneas es significativa. No se debe utilizar `
` para incrementar el espacio entre líneas o párrafos de texto; en su lugar puedes usar propiedades CSS o el elemento `<p>`.

- **Tipo:** elemento en bloque

Ejemplo: el siguiente ejemplo muestra un uso correcto del elemento `
`

```
<!DOCTYPE html>
<html lang="es">
  <head> ... </head>
  <body>
    <h1> Ejemplo de <code>br</code> </h1>

    <p>P. Sherman<br>
    42 Wallaby Way<br>
    Sydney</p>
  </body>
</html>
```

Ejemplo de br

P. Sherman
42 Wallaby Way
Sydney

Por otra parte, la etiqueta `<hr>` es otro elemento vacío que representa un salto temático en el contenido. No debe utilizarse para indicar el inicio o final de una sección, ya que este elemento ya implica un cambio temático por sí mismo. En general, los navegadores lo muestran como una línea horizontal en color gris.

- **Tipo:** elemento en bloque

Ejemplo: en el siguiente ejemplo, se muestran un par de párrafos separados por el elemento `hr` para enfatizar el contraste entre sus temáticas.

```
<!DOCTYPE html>
<html lang="es">
  <head> ... </head>
  <body>
    <h1> Lenguajes Web </h1>

    <p> HTML es el lenguaje de marcado estándar para crear páginas web.
    HTML describe la estructura de una página web y consta de una
    serie de elementos. Los elementos HTML le dicen al navegador
    cómo mostrar el contenido. </p>

    <hr>

    <p> CSS es un lenguaje que describe cómo se mostrarán los elementos
    HTML en la pantalla del ordenador o en otros medios. CSS ahorra
    mucho trabajo porque puede controlar el diseño de varias páginas
    web a la vez. </p>

    <hr>

    <p> JavaScript es el lenguaje de programación que permite cambiar el
    contenido HTML y los valores de los atributos. </p>

  </body>
</html>
```

Lenguajes Web

HTML es el lenguaje de marcado estándar para crear páginas web. HTML describe la estructura de una página web y consta de una serie de elementos. Los elementos HTML le dicen al navegador cómo mostrar el contenido.

CSS es un lenguaje que describe cómo se mostrarán los elementos HTML en la pantalla del ordenador o en otros medios. CSS ahorra mucho trabajo porque puede controlar el diseño de varias páginas web a la vez.

JavaScript es el lenguaje de programación que permite cambiar el contenido HTML y los valores de los atributos.