



Tarea 3

Crea una nueva página web, usa la estructura recomendada y realiza los siguientes ejercicios. Crea un reporte con la imagen de los resultados obtenidos en el navegador para cada prueba realizada

Metadatos: el elemento <meta>

Los metadatos son datos que describen datos, y HTML tiene una forma «oficial» de introducir metadatos en un documento — el elemento <meta>.

a) Cambiar el tipo de codificación a ISO-8859-1 y agregar los siguientes párrafos

<meta charset="utf-8">

Este elemento simplemente especifica la codificación de caracteres del documento — es decir, el conjunto de caracteres que el documento puede usar. utf-8 es un conjunto de caracteres universal que incluye casi todos los caracteres de casi cualquier idioma humano. Esto significa que tu página web podrá gestionar la visualización de cualquier idioma; por lo tanto, resulta una buena idea configurarlo en cada página web que crees. Por ejemplo, tu página podría gestionar inglés y japonés sin problemas:

Una página web que contiene caracteres en inglés y japonés, con la codificación de caracteres establecida en universal o utf-8. Ambos idiomas se muestran bien. Si configuras tu codificación de caracteres en ISO-8859-1, por ejemplo (el juego de caracteres para el alfabeto latino), la representación de tu página puede aparecer desordenada:

<p>Ejemplo japonés: ご飯が熱い。 </p>

<p>Ejemplo español: Mañana es miércoles</p>

b) Agrega el autor y la descripción a la etiqueta meta

Dos de esos metadatos que resultan útiles de incluir en tu página definen al autor de la página y proporcionan una descripción concisa de la página.

- **name** especifica el tipo de metadato del que se trata; es decir, qué tipo de información contiene.
- **content** especifica el contenido del metadato en sí.

<meta name="author" content="Mi nombre">

<meta name="description" content="El área de aprendizaje de MDN pretende proporcionar a los recién llegados a la web todo lo que deben saber para empezar a desarrollar páginas web y aplicaciones web.">

c) Comprobar etiquetas

Ve a la página de inicio de Mozilla Developer Network.



Observa el código fuente de la página (Ctrl +clic o clic con el botón derecho en la página y selecciona la opción del menú Ver código fuente de la página).

Encuentra la etiqueta del metadato description. Se verá más o menos así (aunque puede cambiar con el tiempo):

```
<meta name="description" content="The MDN Web Docs site provides information about Open Web technologies including HTML, CSS, and APIs for both Web sites and progressive web apps.">
```

Ahora busca "Mozilla Developer Network" en tu motor de búsqueda favorito. Observarás que efectivamente merece la pena que tener el contenido de la descripción <meta> y el elemento <title> que se utiliza en la búsqueda.

Nota: Muchas características <meta> ya no se usan. Por ejemplo, los motores de búsqueda ignoran el elemento <meta> (<meta name="keywords" content="pon, tus, palabras clave, aquí">), que se supone que facilita palabras clave para motores de búsqueda de forma que se determine la relevancia de esa página según diferentes términos de búsqueda, porque los spammers rellenaban la lista de palabras clave con cientos de palabras clave que sesgaban los resultados.

d) añadir un favicon a tu página

Agregar iconos personalizados a tu sitio

Para enriquecer un poco más el diseño de tu sitio puedes añadir en tus metadatos referencias a iconos personalizados, que se mostrarán en determinados contextos. El más común de ellos es el favicon (abreviatura de favorite icon –icono «favorito», referido al uso que se le da en las listas de «favoritos» o de marcadores («bookmarks»).

Es el primer icono de este tipo: un icono cuadrado de 16 píxeles que se utiliza en varios lugares. Es posible que veas (según el navegador) favicons que se muestran en la pestaña del navegador que contiene cada página abierta y junto a las páginas marcadas en el panel de marcadores.

Guárdalo en el mismo directorio que la página index de tu sitio, en formato .ico (la mayoría de los buscadores admiten favicon en los formatos más comunes como .gif o .png, pero usar el formato ICO garantiza que funcionará hasta en Internet Explorer 6.)

<https://www.ionos.mx/tools/crear-favicon>

Añade la siguiente línea de referencia en el <head> de tu HTML:

```
<link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
```

Los navegadores modernos usan favicons en diversos lugares, como en la etiqueta de la página que está abierta y en el panel de favoritos cuando la añades a tu lista de páginas favoritas:

El panel de marcadores de Firefox, que muestra un ejemplo marcado con un favicon junto a él.



Hoy día hay un montón de otros tipos de iconos a tener presentes. Por ejemplo, los encontrarás en el código fuente de la página de inicio de MDN:

```
<!-- third-generation iPad with high-resolution Retina display: -->
<link rel="apple-touch-icon-precomposed" sizes="144x144"
href="https://developer.mozilla.org/static/img/favicon144.png">
<!-- iPhone with high-resolution Retina display: -->
<link rel="apple-touch-icon-precomposed" sizes="114x114"
href="https://developer.mozilla.org/static/img/favicon114.png">
<!-- first- and second-generation iPad: -->
<link rel="apple-touch-icon-precomposed" sizes="72x72"
href="https://developer.mozilla.org/static/img/favicon72.png">
<!-- non-Retina iPhone, iPod Touch, and Android 2.1+ devices: -->
<link rel="apple-touch-icon-precomposed"
href="https://developer.mozilla.org/static/img/favicon57.png">
<!-- basic favicon -->
<link rel="icon"
href="https://developer.mozilla.org/static/img/favicon32.png">
```

Aplicar CSS y JavaScript a HTML

Prácticamente todos los sitios web usan CSS para darles un buen aspecto y JavaScript para añadir funcionalidades interactivas, como reproductores de vídeo, mapas, juegos y demás. La manera más habitual de añadir CSS y JavaScript a una página web es con los elementos `<link>` y el elemento `<script>`, respectivamente.

El elemento `<link>` siempre debe ir dentro del `<head>` de tu documento. Este toma dos atributos, `rel="stylesheet"`, que indica que es la hoja de estilo del documento, y `href`, que contiene la ruta al archivo de la hoja de estilo:

```
<link rel="stylesheet" href="my-css-file.css">
```

El elemento `<script>` también debería ir en el `head`, y debería incluir un atributo `src` con la ruta al JavaScript que quieres cargar, y `defer`, que básicamente le dice al navegador que cargue el JavaScript al mismo tiempo que el HTML de la página. Esto es útil porque hace que todo el HTML se cargue antes de ejecutar el JavaScript, para que no haya errores porque el JavaScript ha intentado acceder a un elemento HTML que todavía no existe. De hecho hay múltiples formas de gestionar la carga del JavaScript en una página, pero esta es «a prueba de bombas» para los navegadores modernos.



```
<script src="my-js-file.js"></script>
```

Nota: El elemento `<script>` puede parecer un elemento vacío pero no lo es, y por lo tanto necesita una etiqueta de cierre. En vez de apuntar a un archivo de script externo, también puedes colocar tu código dentro del elemento `<script>`.

e) Crear archivo de javascript, copia el siguiente código y guardalo en un archivo denominado `codigo.js` usa la carpeta `scripts` para guardarlo

```
const list = document.createElement('ul');  
const info = document.createElement('p');  
const html = document.querySelector('html');
```

`info.textContent = 'A continuación se muestra una lista dinámica. Haga clic en cualquier lugar de la página para añadir un nuevo elemento de la lista. Haga clic en un elemento de la lista existente para cambiar su texto por otro.'`

```
document.body.appendChild(info);  
document.body.appendChild(list);
```

```
html.onclick = function() {  
  const listItem = document.createElement('li');  
  const listContent = prompt('¿Qué contenido quiere que tenga el elemento de la lista?');  
  listItem.textContent = listContent;  
  list.appendChild(listItem);  
  
  listItem.onclick = function(e) {  
    e.stopPropagation();  
    const listContent = prompt('Introduzca el nuevo contenido del elemento de la lista');  
    this.textContent = listContent;  
  }  
}
```

f) Crea un archivo de CSS y guardalo como `estilos.css`, usa la carpeta `styles` para guardarlo

```
html {  
  background-color: green;  
  font-size: 20px;  
}
```

```
ul {  
  background: red;
```



```
padding: 10px;  
border: 1px solid black;  
}
```

```
li {  
  margin-left: 20px;  
}
```

g) Añádelos a tu página dentro de la sección del head, debajo de las etiquetas meta

```
<link rel="stylesheet" href="styles/estilos.css">  
<script src="scripts/codigo.js"></script>
```

Establecer el idioma principal del documento

Por último, merece la pena mencionar que puedes (y realmente deberías) especificar el idioma de tu página. Esto se puede hacer añadiendo el atributo lang a la etiqueta de apertura del HTML.

```
<html lang="es-MX">
```

Esto resulta útil en muchos sentidos. Los motores de búsqueda indexarán tu documento HTML de modo más efectivo si estableces el idioma (permitiendo, por ejemplo, que aparezca correctamente en los resultados del idioma especificado), y resulta útil para que las personas con discapacidad visual que utilizan los lectores de pantalla (por ejemplo, la palabra «six» existe tanto en francés como en inglés, pero su pronunciación es diferente).

También puedes establecer que las subsecciones de tu documento se reconozcan en diferentes idiomas. Por ejemplo, podemos establecer que nuestra sección de japonés se reconozca como japonés, de la siguiente manera:

```
<p>Ejemplo Japonés: <span lang="ja">ご飯が熱い。</span>.</p>
```

h) Establecer el idioma español de México

Encabezados y párrafos

La mayor parte del texto estructurado está compuesto por encabezados y párrafos, independientemente de si lees una historia, un periódico, un libro de texto, una revista, etc. El contenido estructurado simplifica la experiencia en la lectura y se disfruta más.



i) Establece la jerarquía en el siguiente texto usa elementos de encabezado y párrafos <h> y <p>

La noche aplastante Por Chris Mills Capítulo 1: La oscura noche Era una noche oscura. En algún lugar, un búho ululó. La lluvia azotó el ... Capítulo 2: El silencio eterno Nuestro protagonista ni susurrar pudo al ver esa sombría figura ... El espectro habla Habían pasado varias horas más, cuando de repente el espectro se incorporó y exclamó: "¡Por favor, ten piedad de mi alma!"

Realmente es a tu elección lo que cada elemento involucrado represente, siempre y cuando la jerarquía tenga sentido. Solo es necesario que tengas en cuenta unas pocas buenas prácticas mientras creas la estructura:

- Preferentemente debes usar solo un <h1> por página; este es el nivel de título superior, y todos los demás se sitúan por debajo de él en la jerarquía.
- Asegúrate de que usas los títulos en el orden correcto en la jerarquía. No uses los <h3> para representar subtítulos, seguidos de los <h2> para representar los subtítulos de los subtítulos; eso no tiene sentido y provocará resultados extraños.
- De los seis niveles de títulos disponibles, debes procurar no usar más de tres por página, a menos que creas que es realmente necesario. Los documentos con muchos niveles (es decir, una jerarquía de títulos muy profunda) son de difícil manejo y navegación. En esos casos se recomienda, si es posible, separar el contenido en varias páginas.

¿Por qué necesitamos estructura?

- Los usuarios que miran una página web tienden a escanearla rápidamente para encontrar el contenido relevante que hay en ella y a menudo empiezan por leer los encabezados para



comenzar. (Solemos pasar muy poco tiempo en una página web). Si no pueden ver nada útil en unos segundos, es probable que se sientan frustrados y se vayan a otro sitio.

- Los motores de búsqueda que indizan tu página consideran el contenido de los títulos como palabras clave importantes e influyen en la puntuación de búsqueda de la página. Sin encabezados, tu página tendrá un rendimiento bajo en términos de optimización de motores de búsqueda SEO.
- Las personas con discapacidad visual severa no suelen leer páginas web: en lugar de ello, las escuchan. Lo hacen con un software llamado Lector de pantalla. Este software proporciona acceso rápido a un contenido textual dado. Entre las diversas técnicas que emplean, leen los encabezados para proporcionar un esquema del documento que permite a los usuarios encontrar rápidamente la información que quieren. Si no hay encabezados disponibles, se ven obligados a escuchar el documento entero.
- Para aplicar estilos al contenido con CSS, o para que haga cosas interesantes con JavaScript, necesitas tener elementos que delimiten el contenido relevante para que CSS/JavaScript se puedan focalizar en este efectivamente.

Por lo tanto, debemos dar a nuestro contenido una estructura definida.

Listas anidadas

Es perfectamente correcto anidar una lista dentro de otra. Posiblemente quieras tener subelementos bajo elementos de rango superior. Tomemos la segunda lista de nuestro ejemplo de la receta:

j) Incluye el siguiente código en tu página

```
<ol>
<li>Pela el ajo y picarlo en trozos gruesos.</li>
<li>Retira las semillas y el tallo del pimiento, y cortarlo en trozos gruesos.</li>
<li>Mete todos los alimentos en un procesador de alimentos.</li>
<li>Pica todos los ingredientes hasta conseguir una pasta.</li>
<li>Si deseas un hummus "grueso", procésalo corto tiempo.</li>
<li>Pica durante más tiempo si se desea obtener un hummus "suave".</li>
</ol>
```

Énfasis

Cuando queremos dar énfasis al lenguaje hablado, acentuamos ciertas palabras y así alteramos sutilmente el significado de lo que decimos. De manera similar, en el lenguaje escrito ponemos palabras en cursiva para destacarlas. En HTML usamos el elemento `` («emphasis»). El navegador, de manera predeterminada, aplica el estilo de letra itálica, pero no debes utilizar esta etiqueta solamente para establecer el estilo de letra itálica.

Resaltar Importancia



Para enfatizar palabras importantes al hablar solemos acentuarlas, y al escribir lo hacemos en estilo negrita.

En HTML usamos el elemento `` (importancia fuerte) para marcar tales expresiones. El documento resulta entonces más útil, y de nuevo los lectores de pantalla reconocen estos elementos y el tono de voz cambia a uno más fuerte. El estilo negrita es el que aplican los navegadores por omisión, pero no debes usar esta etiqueta solamente para aplicar este estilo. Para hacer eso usa el elemento `` y CSS, o un elemento ``.

k) Agrega los siguientes párrafos para enfatizar y remarcar una palabra

```
<p>Me <em>alegro</em> de que no llegues <em>tarde</em>.</p>
<p>Este líquido es <strong>altamente tóxico</strong>.</p>
<p>Cuento contigo. <strong>¡No llegues tarde!</strong></p>
<p>Este líquido es <strong>altamente tóxico</strong>
si lo bebes, <strong>podrías <em>morir</em></strong>.</p>
```

Anatomía de un enlace

Un enlace básico se crea incluyendo el texto, que queramos convertir en un enlace usando un elemento ancla `<a>`, dándole un atributo `href` (también conocido como «Hypertext Reference», «target» u objetivo) que contendrá la dirección web hacia dónde queremos que apunte el enlace. Otro atributo que posiblemente quieras agregar a tus enlaces es `title`. El título contiene información adicional sobre el enlace, como qué tipo de información contiene la página o cosas que debes tener en cuenta en el sitio web.

Como hemos mencionado anteriormente, puedes convertir cualquier contenido en un enlace, incluso Elementos de bloque y elementos en línea. Si quieres convertir una imagen en un enlace, simplemente usa el elemento `<a>` encerrando el elemento `` entre `<a>` y ``.

```
<a href="https://www.mozilla.org/es-ES/">
  
</a>
```

l) Agrega una imagen con un enlace a tu página web, usa el código del párrafo anterior como referencia

Rutas

Las URLs utilizan rutas para encontrar los archivos. Las rutas especifican dónde se encuentra el archivo que buscas dentro del sistema de archivos.

```
<p>Un enlace <a href="contactos/contacto.html">página de contactos</a>.</p>
```

m) Crea la siguiente estructura en tu proyecto



```
---index.html
  proyectos/proyecto.html
  documentos/tarea1.pdf
  personales/info/personal.html
```

- en index.html crea un enlace a proyecto.html, tarea1.pdf y personal.html
- en proyecto.html crea un enlace index.html, tarea1.pdf y personal.html
- en personal.html crea un enlace a index.html, proyecto.html y tarea1.pdf

Fragmentos de documento

Es posible apuntar hacia una parte concreta de un documento HTML en vez de a todo un documento. Para ello hay que asignar previamente un atributo id al elemento hacia el que apuntamos. Esto se debe hacer en el encabezado y quedará así:

```
<h2 id="Dirección_de_envío">Dirección de envío</h2>
```

Posteriormente para hacer referencia a este id concreto, lo añadiremos al final de la URL precedido por un #

```
<p>¿Quieres mandarnos una carta? Aquí tienes nuestra <a
href="..../index.html#Dirección_de_envío">Dirección de envío</a>.</p>
```

También podemos usar esta referencia a un fragmento de documento para apuntar hacia otra parte del mismo documento:

```
<p>La <a href="#Dirección_de_envío">Dirección de envío de la empresa</a> se encuentra al final
de esta página.</p>
```

n) agrega a un enlace a fragmentos dentro del documento proyecto.html

URLs absolutas y relativas

Dos términos que encontrarás en la Web son URL absoluta y URL relativa:

URL absoluta: Hace referencia a una dirección definida por su ubicación absoluta en la web, esta incluye el protocolo y el nombre del dominio. Por ejemplo, si subes una página de inicio index.html a un directorio llamado projects que se encuentra dentro de la raíz de un servidor web, y el dominio del sitio web es http://www.example.com, se podrá acceder a la página desde http://www.example.com/projects/index.html (o simplemente http://www.example.com/projects/, ya que la mayoría de los servidores web buscan la página de inicio index.html para cargarla por omisión si no se les especifica otra en la URL).

Una URL absoluta siempre apuntará a la misma dirección, sin importar desde dónde se utilice.



Una URL relativa: Hace referencia a una dirección que depende de la posición del archivo desde donde se utiliza — son las que vimos en la sección anterior. Por ejemplo, si desde un archivo ubicado en `http://www.example.com/projects/index.html` queremos enlazar hacia un archivo PDF ubicado en el mismo directorio, la URL sería simplemente el nombre del archivo (por ejemplo: `project-brief.pdf`) no necesitamos más información. Si el archivo PDF está situado en un subdirectorio dentro de `projects` llamado `pdfs`, la URL relativa es: `pdfs/project-brief.pdf` (la URL absoluta equivalente sería: `http://www.example.com/projects/pdfs/project-brief.pdf`).

Una URL relativa hará referencia a diferentes direcciones según dónde se encuentre el archivo desde el cual se utiliza — por ejemplo, si movemos nuestro archivo `index.html` del directorio `projects` a la raíz del sitio web (el nivel más alto, no cualquiera de los otros directorios), la URL relativa `pdfs/project-brief.pdf` ahora hará referencia a `http://www.example.com/pdfs/project-brief.pdf`, en lugar de a `http://www.example.com/projects/pdfs/project-brief.pdf`.

ñ) Investiga las siguientes etiquetas de HTML, agrega a tu página web un ejemplo de cada uno de ellos en donde se muestre su funcionamiento.

`<dl>`, `<dt>`, `<dd>`, `<blockquote>`, `<q>`, `<cite>`, `<abbr>`, `<address>`, `<sub>`, `<sup>`, `<code>`, `<pre>`, `<var>`, `<kbd>`, `<samp>`, `<time>`