



Argentina
programa



Programación FrontEnd





Antes de profundizar en los detalles, es importante comprender qué es la Programación FrontEnd y su importancia en el desarrollo web. La Programación FrontEnd es la parte del desarrollo web que se enfoca en la parte visible y tangible de un sitio web o aplicación. Esto incluye el diseño, la interacción y la funcionalidad que los usuarios experimentan directamente en la interfaz de usuario. En otras palabras, la Programación FrontEnd es responsable de todo lo que un usuario ve y con lo que interactúa en la web.

En un mundo cada vez más digital, la Programación FrontEnd es una habilidad muy valorada. Los sitios web y aplicaciones bien diseñados y fáciles de usar pueden marcar la diferencia en la forma en que los usuarios interactúan con una empresa o una marca. Una buena Programación FrontEnd puede hacer que un sitio web sea más accesible, fácil de usar y estéticamente agradable.

Para convertirse en un buen Programador FrontEnd, es necesario tener una sólida comprensión de varias habilidades. En estos próximos módulos, exploraremos algunos de los fundamentos básicos, así como algunas tecnologías clave que se utilizan comúnmente en la Programación FrontEnd. Algunos de los temas que cubriremos incluyen:

HTML: el lenguaje de marcado utilizado para crear la estructura y el contenido de un sitio web.

CSS: el lenguaje utilizado para dar estilo y diseño a un sitio web.

JavaScript: el lenguaje de programación utilizado para crear interactividad y dinamismo en un sitio web.

Frameworks y bibliotecas: herramientas populares que se utilizan para acelerar el proceso de desarrollo y mejorar la calidad del código.

Responsive Design: la práctica de diseñar sitios web que se adaptan a diferentes dispositivos y tamaños de pantalla.

A lo largo de este módulo, exploraremos cada uno de estos temas en detalle. Aprenderá cómo crear una página web desde cero, cómo aplicar estilos a su sitio web y cómo agregar interactividad a través de JavaScript. También exploraremos cómo utilizar herramientas y tecnologías populares.

En resumen, la Programación FrontEnd es una habilidad clave en el desarrollo web moderno. Con este curso, estará bien equipado para crear experiencias de usuario efectivas y atractivas en la web. ¡Comencemos!



¿Qué es y cómo se usa HTML?

HTML no se considera un lenguaje de programación, sino más bien un lenguaje de marcado utilizado para definir la estructura de contenido. Este lenguaje está compuesto por una serie de elementos que se utilizan para delimitar y controlar el aspecto y comportamiento de diferentes partes del contenido. Las etiquetas de cierre permiten transformar una palabra o imagen en un hipervínculo que dirige a otra página, aplicar formatos como cursiva, cambiar el tamaño de la letra, entre otras funciones. A modo de ejemplo, considera la siguiente línea de contenido:

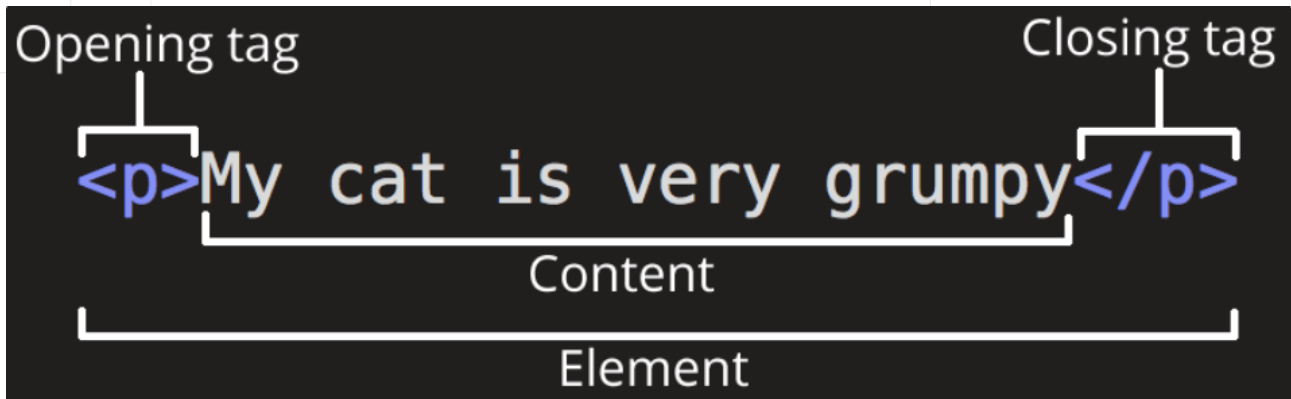
```
2  
3  Mi gato es muy gruñon  
4
```

Para indicar que un bloque de texto corresponde a un párrafo, es posible utilizar la etiqueta `<p>` para encerrar dicho texto. De esta manera, se logra definir su estructura y separarlo del resto del contenido.

```
1  
2  <p> Mi gato es muy gruñon </p>  
3  
4
```



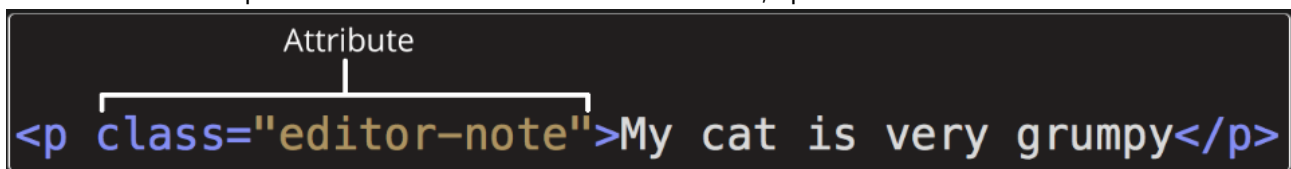
Anatomía de un elemento



Las partes principales de un elemento son las siguientes:

- La etiqueta de apertura: contiene el nombre del elemento, como por ejemplo "p", y está encerrada por paréntesis angulares (< >) de apertura y cierre. Esta etiqueta establece dónde comienza el elemento y en este caso específico, indica el inicio del párrafo.
- La etiqueta de cierre: es igual a la etiqueta de apertura, pero incluye una barra de cierre (/) antes del nombre de la etiqueta. Esta etiqueta indica el final del elemento y en el caso del párrafo, señala el término del mismo.
- El contenido: se refiere al texto que se encuentra dentro del elemento. En este caso, el contenido es simplemente el texto del párrafo.
- El elemento: se compone de la etiqueta de apertura, la etiqueta de cierre y el contenido. Estas tres partes juntas conforman el elemento completo.

Los elementos pueden también tener atributos, que se ven así:



Los atributos contienen información adicional acerca del elemento, la cual no quieres que aparezca en el contenido real del elemento. Aquí class es el nombre del atributo y editor-note el valor del atributo. En este caso, el atributo class permite darle al elemento un nombre identificativo, que se puede utilizar luego para apuntarle al elemento información de estilo y demás cosas.



Un atributo debe tener siempre:

- Un espacio entre este y el nombre del elemento (o del atributo previo, si el elemento ya posee uno o más atributos).
- El nombre del atributo, seguido por un signo de igual (=).
- Comillas de apertura y de cierre, encerrando el valor del atributo.

Los atributos siempre se incluyen en la etiqueta de apertura de un elemento, nunca en la de cierre.

Nota: el atributo con valores simples que no contengan espacios en blanco ASCII (o cualesquiera de los caracteres " ' ` = < >) pueden permanecer sin entrecomillar, pero se recomienda entrecomillar todos los valores de atributo, ya que esto hace que el código sea más consistente y comprensible.





Anidar elementos

Puedes también colocar elementos dentro de otros elementos —esto se llama anidamiento—. Si, por ejemplo, quieres resaltar una palabra del texto (en el ejemplo la palabra «muy»), podemos encerrarla en un elemento ``, que significa que dicha palabra se debe enfatizar:

```
<p>Mi gato es <strong>muy</strong> gruñon.</p>
```

Debes asegurarte que los elementos estén correctamente anidados: en el ejemplo de abajo, creaste la etiqueta de apertura del elemento `<p>` primero, luego la del elemento ``, por lo tanto, debes cerrar esta etiqueta primero, y luego la de `<p>`. Esto es incorrecto:

```
<p>Mi gato es <strong>muy gruñon.</p></strong>
```

Los elementos deben abrirse y cerrarse ordenadamente, de forma tal que se encuentren claramente dentro o fuera el uno del otro. Si estos se encuentran solapados, el navegador web tratará de adivinar lo que intentas decirle, pero puede que obtengas resultados inesperados. Así que, ¡no lo hagas!

Saltos de línea

En ocasiones puede ser conveniente introducir un punto y aparte cuando estamos escribiendo un texto. En HTML para hacer esto puede usarse la etiqueta `
`. Cuando el navegador se encuentra con esta etiqueta provoca un salto de línea y sigue mostrando el contenido de la página. Veamos un ejemplo:

*Estaba internándose en lo desconocido.

Durante todo el trayecto hasta Woodfield no dejó de pensar en los aspectos económicos, con el temor de estar cometiendo...*





En este texto podemos observar cómo tras el primer punto era necesario un cambio de línea, por lo que hemos usado `
`

Elementos vacíos

Algunos elementos no poseen contenido, y son llamados elementos vacíos. Toma, por ejemplo, el elemento `` de nuestro HTML:

```

```

Posee dos atributos, pero no hay etiqueta de cierre `` ni contenido encerrado. Esto es porque un elemento de imagen no encierra contenido al cual afectar. Su propósito es desplegar una imagen en la página HTML, en el lugar en que aparece.



Anatomía de un documento HTML

Hasta ahora has visto lo básico de elementos HTML individuales, pero estos no son muy útiles por sí solos. Ahora verás cómo los elementos individuales son combinados para formar una página HTML entera. Vuelve a visitar el código de tu ejemplo en index.html (que viste por primera vez en el artículo Manejo de archivos):

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Document</title>
6  </head>
7  <body>
8
9  </body>
10 </html>
11
```

`<!DOCTYPE html>` es el tipo de documento y se requiere como preámbulo. Anteriormente, los tipos de documento eran importantes para verificar errores y seguir reglas, pero hoy en día solo son un artefacto antiguo que debe incluirse para un correcto funcionamiento.

`<html></html>` es el elemento raíz que encierra todo el contenido de la página.

`<head></head>` es el contenedor para incluir elementos que no son visibles para los visitantes, como palabras clave, descripciones, CSS y declaraciones de características.

`<meta charset="utf-8">` establece el juego de caracteres como utf-8, capaz de soportar cualquier tipo de texto.

`<title></title>` establece el título de la página que se muestra en la pestaña o barra de título del navegador, y se utiliza como descripción en marcadores y favoritos.

`<body></body>` encierra todo el contenido visible para los visitantes, como texto, imágenes, videos, juegos y pistas de audio.



Encabezados

Los Encabezados en HTML permiten definir secciones del contenido como títulos o subtítulos. Esto es similar a como un libro tiene un título principal y capítulos con sus propios títulos y subtítulos. HTML proporciona seis niveles de encabezado, desde el más importante (h1) hasta el menos importante (h6), aunque normalmente se acostumbran a utilizar entre 3 o 4 niveles de encabezado en el código, por ejemplo:

```
1  <h1>Mi título principal</h1>
2
3  <h2>Mi título de nivel superior</h2>
4
5  <h3>Mi subtítulo</h3>
6
7  <h4>Mi sub-subtítulo</h4>
8
```



Listas

Mucho del contenido web está dado por listas, así que HTML tiene elementos especiales para ellas. El marcado de listas se realiza siempre en al menos dos elementos. Los dos tipos de listas más comunes son las listas ordenadas y las desordenadas:

Las listas desordenadas son aquellas en las que el orden de los items no es relevante, como en una lista de compras. Estas son encerradas en un elemento `` (unordered list).

Las listas ordenadas son aquellas en las que el orden sí es relevante, como en una receta. Estas son encerradas en un elemento `` (ordered list). Cada elemento de la lista se coloca dentro de un elemento `` (list item).

Por ejemplo, si quieres transformar parte del siguiente párrafo en una lista:

```
<p>En Mozilla, somos una comunidad de tecnólogos,  
pensadores, y constructores que trabajan juntos...  
</p>
```

Podrías hacer lo siguiente:

```
8  
9   <p>En Mozilla, somos una comunidad de</p>  
10  
11   <ul>  
12     <li>tecnólogos</li>  
13     <li>pensadores</li>  
14     <li>constructores</li>  
15   </ul>  
16  
17   <p>trabajando juntos... </p>  
18
```



Vínculos

Los vínculos o enlaces son muy importantes —son los que hacen de la web, la web—. Para implementar un vínculo, necesitas usar un vínculo simple — `<a>` — la a es la abreviatura de la palabra inglesa «anchor» («ancla»). Para convertir algún texto dentro de un párrafo en un vínculo, sigue estos pasos:

Elige algún texto. Nosotros elegimos «Curso PHP»

Encierra el texto en un elemento `<a>`, así:

```
<a>Curso PHP</a>
```

Proporcióname al elemento `<a>` un atributo href, así:

```
<a href="">Curso PHP</a>
```

Completa el valor de este atributo con la dirección web con la que quieras conectar al vínculo:

```
<a href="https://google.com">Curso PHP</a>
```

Podrías obtener resultados inesperados si al comienzo de la dirección web omites la parte `https://` o `http://` llamada protocolo. Así que luego del marcado del vínculo, haz clic en él para asegurarte que te dirige a la dirección deseada.

Nota: href podría parecer, en principio, una opción un tanto oscura para un nombre de atributo. Si tienes problemas para recordarla, recuerda que se refiere a **hypertext reference** (referencia de hipertexto).





Tablas

¿Qué es una tabla?

Una tabla es un conjunto estructurado de datos distribuidos en filas y columnas (datos tabulados). Una tabla permite buscar con rapidez y facilidad valores entre diferentes tipos de datos que indiquen algún tipo de conexión. Por ejemplo, una persona y su edad, o un día de la semana o el horario de una piscina municipal.

| Person | Age |
|--------|-----|
| Chris | 38 |
| Dennis | 45 |
| Sarah | 29 |
| Karen | 47 |

| Monday | Tuesday | Wednesday | Thursday | Friday | Saturday | Sunday |
|-------------------------------|-----------------------------------------|------------------------------------------------|-------------------------------|------------------------------------------------|------------------------------------------------------------|------------------------------|
| Public Swim 06:30 - 10:30 | Public Swim 06:30 - 09:00 | Public Swim 06:30 - 09:00 | Public Swim 06:30 - 11:15 | Public Swim 06:30 - 09:00 | Lane Swim 08:00 - 09:00 | Lane Swim 08:00 - 09:00 |
| Aquacise 10:30 - 11:15 | Aqua Jog 09:15 - 10:00 | Education Swimming Lessons 09:00 - 12:00 | Aquacise 11:15 - 12:00 | Education Swimming Lessons 09:00 - 12:00 | Oldham Active Kids Swimming Lessons 09:00 - 13:00 | Public Swim 09:00 - 11:00 |
| Lane Swim 11:30 - 13:00 | Parent & Baby Class 09:30 - 10:15 | Lane Swim 12:00 - 13:00 | Lane Swim 12:00 - 13:00 | Lane Swim 12:00 - 13:00 | Parent and Baby 12:00 - 12:45 | Aquacise 11:00 - 11:45 |
| Education Swimming Lessons | Public Swim 10:00 - 11:15 | Public Swim 13:00 - 14:00 | Education Swimming Lessons | Oldham Active Kids Swimming | Public Swim 13:00 - 17:00 | Public Swim 11:45 - 12:00 |



En HTML, las tablas se crean utilizando la etiqueta `<table>` y se organizan en filas y columnas usando las etiquetas `<tr>` y `<td>` respectivamente.

Por ejemplo, para crear una tabla con 2 filas y 2 columnas, se utilizaría el siguiente código HTML:

```
1  <table>
2    <tr>
3      <td>Fila 1, columna 1</td>
4      <td>Fila 1, columna 2</td>
5    </tr>
6    <tr>
7      <td>Fila 2, columna 1</td>
8      <td>Fila 2, columna 2</td>
9    </tr>
10 </table>
```

| | |
|-------------------|-------------------|
| Fila 1, columna 1 | Fila 1, columna 2 |
| Fila 2, columna 1 | Fila 2, columna 2 |



Formularios

Los formularios en HTML se utilizan para recopilar información de los usuarios y enviarla a un servidor web para su procesamiento. Los formularios están compuestos por una serie de elementos, como campos de texto, casillas de verificación, botones de radio y botones de selección, entre otros.

El proceso de creación de un formulario en HTML consta de varios pasos. Primero, se debe crear un elemento de formulario HTML utilizando la etiqueta "form". Dentro del elemento de formulario, se deben incluir los elementos de entrada que se utilizarán para recopilar información. Cada elemento de entrada se define utilizando una etiqueta HTML específica, como "input", "textarea" o "select".

```
<body>  
  <form>  
    <input type="text" name="Correo" >  
    <input type="password" name="Clave" >  
    <input type="submit" value="Enviar">  
  </form>  
</body>
```

Cada elemento de entrada tiene un atributo "name", que se utiliza para identificar el elemento en el servidor web al que se enviará el formulario. También puede tener un atributo "value", que define el valor inicial del campo de entrada. Además, se pueden utilizar otros atributos para definir el tipo de entrada que se permitirá, como "type" para especificar si se trata de un campo de texto, una casilla de verificación o un botón de radio.

Una vez que se ha definido el formulario y los elementos de entrada, se utiliza un botón de envío para enviar el formulario al servidor web. El botón de envío se define utilizando la etiqueta "input" con el atributo "type" establecido en "submit". Al hacer clic en este botón, se envía el formulario al servidor web para su procesamiento.





En resumen, los formularios en HTML son una forma útil de recopilar información de los usuarios y enviarla a un servidor web para su procesamiento. Para crear un formulario, se definen los elementos de entrada utilizando etiquetas HTML específicas y se utiliza un botón de envío para enviar el formulario al servidor web.

