



HTML Boilerplate

Curso de Fundamentos

Introducción

Todos los documentos HTML tienen la misma estructura básica o boilerplate que debe estar en su lugar antes de que se pueda hacer algo útil. En esta lección, exploraremos las diferentes partes de este boilerplate y veremos cómo encaja todo.

Resumen de la lección

Esta sección contiene una visión general de los temas que aprenderá en esta lección.

- Cómo escribir el boilerplate básico para un documento HTML.
- Cómo abrir documentos HTML en su navegador.

Crear un archivo HTML

Para demostrar un estándar HTML, primero necesitamos un archivo HTML para trabajar.

Crea una nueva carpeta en tu ordenador y llámala `html-boilerplate`. Dentro de esa carpeta, crea un nuevo archivo y nómbralo `index.html`.

Probablemente ya estés familiarizado con muchos tipos diferentes de archivos, por

ejemplo, archivos doc, pdf e imagen.

Para que el ordenador sepa que queremos crear un archivo HTML, necesitamos añadir el nombre del archivo con la extensión `.html`, como hemos hecho al crear el archivo `index.html`.

It is worth noting that we named our HTML file `index`. We should always name the HTML file that will contain the homepage of our website `index.html`. This is because web servers will by default look for an `index.html` page when users land on our websites – and not having one will cause big problems.

El DOCTYPE

Every HTML page starts with a doctype declaration. The doctype's purpose is to tell the browser what version of HTML it should use to render the document. The latest version of HTML is HTML5, and the doctype for that version is `<!DOCTYPE html>`.

Los tipos de documentos para versiones anteriores de HTML eran un poco más complicados. Por ejemplo, esta es la declaración de tipo de documento para HTML4:

```
1 | <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
```

However, we probably won't ever want to be using an older version of HTML, so we'll always use `<!DOCTYPE html>`.

Open the `index.html` file created earlier in your text editor and add `<!DOCTYPE html>` to the very first line.

Elemento HTML

Después de declarar el doctype, necesitamos proporcionar un elemento `<html>`. Esto es lo que se conoce como el elemento raíz del documento, lo que significa que todos los demás elementos del documento serán descendientes del mismo.

This becomes more important later on when we learn about manipulating HTML with

JavaScript. For now, just know that the `<html>` element should be included on every HTML document.

Back in the `index.html` file, let's add the `<html>` element by typing out its opening and closing tags, like so:

```
1 | <!DOCTYPE html>
2 | <html lang="en">
3 | </html>
```

Noticed the word `lang` here? It represents an HTML attribute which is associated with the given HTML tag i.e. `<html>` in this case. These attributes provide additional information about HTML elements. (More about `HTML attributes` in the following lesson.)

¿Cuál es el atributo de lang?

`lang` especifica el idioma del contenido del texto en ese elemento. Este atributo se utiliza principalmente para mejorar la accesibilidad de la página web. Permite que las tecnologías de asistencia, por ejemplo, lectores de pantalla, se adapten según el idioma e invoquen la pronunciación correcta.

Elemento principal

The `<head>` element is where we put important meta-information **about** our webpages, and stuff required for our webpages to render correctly in the browser. Inside the `<head>`, we **should not** use any element that displays content on the webpage.

Back in our `index.html` file, let's add a `<head>` element with a `<meta>` element and a title within it. The `<head>` element goes within the `<html>` element and should always be the first element under the opening `<html>` tag:

```
1 | <!DOCTYPE html>
2 |
```

```
3 <html lang="en">
4   <head>
5     <meta charset="UTF-8">
6     <title>My First Webpage</title>
7   </head>
8 </html>
```

Metaelemento

We should always have the `<meta>` tag with the charset encoding of the webpage in the `<head>` element: `<meta charset="utf-8">`.

Establecer la codificación es muy importante porque garantiza que la página web muestre símbolos y caracteres especiales de diferentes idiomas correctamente en el navegador.

Elemento de título

Otro elemento que siempre debemos incluir en la cabecera de un documento HTML es el elemento `<title>`:

```
<title>My First Webpage</title>
```

The `<title>` element is used to give webpages a human-readable title, which is displayed in our webpage's browser tab. For example, if you look at the current tab's name of your browser, it will read "HTML Boilerplate | The Odin Project"; this is the `<title>` of the current `.html` file.

Si no incluimos un elemento `<title>`, el título de la página web sería predeterminado con su nombre de archivo. En nuestro caso, eso sería `index.html`, que no es muy significativo para los usuarios; esto haría que fuera muy difícil encontrar nuestra página web si el usuario tiene muchas pestañas del navegador abiertas.

Hay muchos más elementos que pueden ir dentro de la cabecera de un documento HTML. Sin embargo, por ahora solo es crucial conocer los dos elementos que hemos cubierto aquí. Introduciremos más elementos que van a la cabeza durante el resto del plan de estudios.

Elemento corporal

El último elemento necesario para completar el boilerplate HTML es el elemento `<body>`. Aquí es donde irá todo el contenido que se mostrará a los usuarios: el texto, las imágenes, las listas, los enlaces, etc.

To complete the boilerplate, add a `<body>` element to the `index.html` file. The `<body>` element also goes within the `<html>` element and is always below the `<head>` element, like so:

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

Visualización de archivos HTML en el navegador

The HTML boilerplate in the `index.html` file is complete at this point, but how do you view it in the browser? There are a couple of different options:

Usar Google Chrome

Para evitar ramificar las instrucciones de nuestra lección para acomodar todas las diferencias entre los navegadores, vamos a usar Google Chrome como nuestro navegador principal durante el resto de este curso. Todas las referencias al navegador pertenecerán específicamente a Google Chrome. Le sugerimos **encarecidamente** que utilice Google Chrome para todas sus pruebas en el futuro.

1. Puede arrastrar y soltar un archivo HTML desde su editor de texto a la barra de direcciones de su navegador.
2. Puede encontrar el archivo HTML en su sistema de archivos y luego hacer doble clic en él. Esto abrirá el archivo en el navegador predeterminado que utiliza su sistema.
3. Puede usar el terminal para abrir el archivo en su navegador:
 - Ubuntu: Navegue hasta el directorio que contiene el archivo y el tipo `google-chrome index.html`
 - macOS: Navegar al directorio que contiene el archivo y el tipo `open ./index.html`
 - WSL: Navegue hasta el directorio que contiene el archivo y escriba `explorer.exe index.html`. Tenga en cuenta que esto abrirá el archivo en el navegador predeterminado que utiliza su sistema.

Using one of the methods above, open up the `index.html` file we have been working on. You'll notice the screen is blank. This is because we don't have anything in our body to display.

De vuelta en el archivo `index.html`, agreguemos un encabezado (más sobre estos más adelante) al cuerpo y guardemos el archivo:

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8">
5      <title>My First Webpage</title>
6    </head>
7
8    <body>
9      <h1>Hello World!</h1>
10   </body>
11 </html>
```

Ahora, si actualizas la página en el navegador, deberías ver que los cambios surten efecto y el encabezado "¡Hola mundo!" Se mostrará.

Acceso directo de VSCode

VSCode has a built-in shortcut you can use for generating all the boilerplate in one go. Please note that this shortcut only works while editing a file with the `.html` extension or a text file with the HTML language already selected. To trigger the shortcut, delete everything in the `index.html` file and just enter `!` on the first line. This will bring up a couple of options. Press the `Enter` key to choose the first one, and voila, you should have all the boilerplate populated for you.

Puede notar que el boilerplate producido por este atajo incluye una línea que aún no hemos mencionado:

```
1 | <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Esto no es algo que necesitemos saber hasta que discutamos el diseño receptivo, un tema avanzado que involucra diferentes tamaños de pantalla que cubriremos mucho más adelante en el plan de estudios. Por ahora, puedes dejar esa línea como está.

Todavía es bueno saber cómo escribir el boilerplate usted mismo en caso de que se encuentre usando un editor de texto como el bloc de notas (el paño prohibido), que no tiene este atajo. Trate de no usar el acceso directo en sus primeros proyectos HTML, para que pueda construir algo de memoria muscular para escribir el código estándar.

Asignación

1. Mira y sigue el brillante [vídeo](#) de Kevin Powell [Building Your First Web Page](#).
2. Build some muscle memory by deleting the contents of the `index.html` file and trying to write out all the boilerplate again from memory. Don't worry if

you have to peek at the lesson content the first few times if you get stuck. Just keep going until you can do it a couple of times from memory.

3. Ejecute su boilerplate a través del [validador HTML](#) W3. Los validadores aseguran que su marcado sea correcto y sean una excelente herramienta de aprendizaje, ya que proporcionan comentarios sobre errores de sintaxis que puede estar cometiendo a menudo y de los que no conoce, como la falta de etiquetas de cierre y espacios adicionales en su HTML.

Comprobar de conocimientos

Las siguientes preguntas son una oportunidad para reflexionar sobre temas clave de esta lección. Si no puede responder a una pregunta, haga clic en ella para revisar el material, pero tenga en cuenta que no se espera que memorice o domine este conocimiento.

- [¿Cuál es el propósito de la declaración de tipo de documento?](#)
- [¿Qué es el elemento HTML?](#)
- [¿Cuál es el propósito del elemento principal?](#)
- [¿Cuál es el propósito del elemento corporal?](#)

Recursos adicionales

Esta sección contiene enlaces útiles a contenido relacionado. No es obligatorio, así que considérelo como suplemento.

- Otra opción para abrir sus páginas HTML en el navegador es usar la [extensión Live Preview](#) para VSCode. Esto abrirá su documento HTML y lo actualizará automáticamente cada vez que guarde el documento. Sin embargo, recomendamos no usar esta extensión y, en su lugar, hacerlo a la antigua usanza, abriendo la página y actualizando la página manualmente en el navegador para sus primeros proyectos HTML. De esta manera, puedes acostumbrarte a ese proceso y no dependerás de las extensiones de inmediato.

- If you wish, you can add [the lang attribute](#) to individual elements throughout the webpage.

 [Mejorar en GitHub](#)

 [Informar de un problema](#)

[Ver registro de cambios de la lección](#)



Ver curso

Marcar completo



Próxima lección

¡Apóyanos!

El Proyecto Odin está financiado por la comunidad. ¡Únete a nosotros para empoderar a los estudiantes de todo el mundo apoyando a The Odin Project!

más información

Donar ahora



THE ODIN PROJECT

Educación de codificación de alta calidad mantenida por una comunidad de código abierto.



Sobre nosotros

[acerca de](#)

[EQUIPO](#)

[blog](#)

[Casos de éxito](#)

soporte

[preguntas frecuentes](#)

[Contribuir](#)

[Contacta con nosotros](#)

Guías

[Guías de la comunidad](#)

[Guías de instalación](#)

LEGAL

[Términos](#)

[PRIVACIDAD](#)