



Enlaces e imágenes

Curso de Fundamentos

Introducción

Los enlaces son una de las características clave de HTML. Nos permiten vincularnos a otras páginas HTML en la web. De hecho, por eso se llama web.

En esta lección, aprenderemos cómo crear enlaces y agregar algo de estilo visual a nuestros sitios web insertando imágenes.

Resumen de la lección

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

- Cómo crear enlaces a páginas de otros sitios web en Internet.
- Cómo crear enlaces a otras páginas de sus propios sitios web.
- La diferencia entre enlaces absolutos y relativos.
- Cómo mostrar una imagen en una página web usando HTML.

Preparación

Para practicar el uso de enlaces e imágenes a lo largo de esta lección, necesitamos un proyecto HTML con el que trabajar.

1. Crea un nuevo directorio llamado `odin-links-and-images`.

2. Dentro de ese directorio, crea un nuevo archivo llamado `index.html`.
3. Abra el archivo en VS Code y complete el código HTML habitual.
4. Por último, agregue el siguiente h1 al cuerpo:

```
1 | <h1>Homepage</h1>
```

Elementos de anclaje

Para crear un enlace en HTML, utilizamos el elemento de anclaje. Un elemento de anclaje se define envolviendo el texto u otro elemento HTML que queremos que sea un enlace con una `<a>` etiqueta.

Agrega lo siguiente al cuerpo de la `index.html` página que creamos y ábrela en el navegador:

```
1 | <a>About The Odin Project</a>
```

Es posible que hayas notado que hacer clic en este enlace no hace nada. Esto se debe a que una etiqueta de ancla por sí sola no sabrá a dónde queremos vincular. Tenemos que indicarle un destino al que dirigirnos. Para ello, utilizamos un atributo HTML.

Un atributo HTML proporciona información adicional a un elemento HTML y siempre va en la etiqueta de apertura del elemento. Un atributo suele estar formado por dos partes: un nombre y un valor; sin embargo, no todos los atributos requieren un valor. En nuestro caso, necesitamos agregar un `href` atributo (referencia de hipertexto) a la etiqueta de anclaje de apertura. El valor del atributo href es el destino al que queremos que se dirija nuestro enlace.

Agrega el siguiente `href` atributo al elemento de anclaje que creamos anteriormente e intenta hacer clic en él nuevamente, no olvides actualizar el navegador para que se puedan aplicar los nuevos cambios.

```
1 | <a href="https://www.theodinproject.com/about">About The Odin
```

De forma predeterminada, cualquier texto envuelto con una etiqueta de ancla sin un `href` atributo se verá como texto sin formato. Si el `href` atributo está presente, el navegador le dará al texto un color azul y lo subrayará para indicar que es un enlace.

Vale la pena señalar que puedes usar etiquetas de anclaje para vincular a cualquier tipo de recurso en Internet, no solo a otros documentos HTML. Puedes vincular a videos, archivos PDF, imágenes, etc., pero en la mayoría de los casos, vincularás a otros documentos HTML.

Abrir enlaces en una nueva pestaña

El método que se muestra arriba abre los enlaces en la misma pestaña que la página web que los contiene. Este es el comportamiento predeterminado de la mayoría de los navegadores y se puede cambiar con relativa facilidad. Todo lo que necesitamos es otro atributo: el `target` atributo.

While `href` specifies the destination link, `target` specifies where the linked resource will be opened. If it is not present, then, by default, it will take on the `_self` value which opens the link in the current tab. To open the link in a new tab or window (depends on browser settings) you can set it to `_blank` as follows:

```
1 | <a href="https://www.theodinproject.com/about" target="_blank"
```

You may have noticed that we snuck in the `rel` attribute above. This attribute is used to describe the relation between the current page and the linked document.

noopener : The `noopener` attribute ensures that a link opened in a new tab or window cannot interact with or access the original page. Without it, the new page can use JavaScript to manipulate the original page, which poses a security risk.

For example: "[Open Example](#)"

In this code: `target="_blank"`: opens the link in a new tab. `rel="noopener"`: prevents the new tab from accessing the original page, ensuring security.

Without `noopener`, the new tab could use JavaScript to interact with the original page, which is unsafe.

noreferrer : The **noreferrer** attribute provides both privacy and security. It prevents the new page from knowing where the user came from (hiding the referrer) and also includes the behavior of **noopener** , preventing the new page from accessing the original page.

For example: "[Visit Example](#)"

In this example: `target="_blank"`: opens the link in a new tab. `rel="noreferrer"`: ensures the new page cannot see the referring page's address (privacy) and prevents it from accessing the original page (security).

By using `rel="noreferrer"`, you automatically get the benefits of both privacy and security.

Why do we need this added behaviour for opening links in new tabs? Security reasons. The prevention of access that is caused by **noopener** prevents [phishing attacks](#) where the opened link may change the original webpage to a different one to trick users. This is referred to as [tabnabbing](#). Adding the **noreferrer** value can be done if you wish to not let the opened link know that your webpage links to it.

Note that you may be fine if you forget to add `rel="noopener noreferrer"` since more recent versions of browsers [provide security](#) if only `target="_blank"` is present. Nevertheless, in line with good coding practices and to err on the side of caution, it is recommended to always pair a `target="_blank"` with a `rel="noopener noreferrer"` .

Absolute and relative links

Generally, there are two kinds of links we will create:

- Links to pages on other websites on the internet.
- Links to pages located on our own websites.

Absolute links

Links to pages on other websites on the internet are called absolute links. A typical absolute link will be made up of the following parts: `scheme://domain/path` . An absolute link will always contain the [scheme and domain](#) of the destination.

We've already seen an absolute link in action. The link we created to The Odin Project's About page earlier was an absolute link as it contains the scheme and

domain.

<https://www.theodinproject.com/about>

Relative links

Links to other pages within our own website are called relative links. Relative links do not include the domain name, since it is another page on the same site, it assumes the domain name will be the same as the page we created the link on.

Relative links only include the file path to the other page, *relative* to the page you are creating the link on. This is quite abstract, let's see this in action using an example.

Within the `odin-links-and-images` directory, create another HTML file named `about.html` and paste the following code into it:

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>Odin Links and Images</title>
6   </head>
7
8   <body>
9     <h1>About Page</h1>
10  </body>
11 </html>
```

Back in the index page, add the following anchor element to create a link to the about page:

```
1 <body>
2   <h1>Homepage</h1>
3   <a href="https://www.theodinproject.com/about">About The Odi
4
5   <a href="about.html">About</a>
6 </body>
```

Open the index file in a browser and click on the about link to make sure it is all wired together correctly. Clicking the link should go to the about page we just created.

This works because the index and about page are in the same directory. That means we can use its name (`about.html`) as the link's `href` value.

But we will usually want to organize our website directories a little better. Normally we would only have the `index.html` at the root directory and all other HTML files in their own directory.

Create a directory named `pages` within the `odin-links-and-images` directory and move the `about.html` file into this new directory.

Refresh the index page in the browser and then click on the about link. It will now be broken. This is because the location of the about page file has changed.

To fix this, we just need to update the about link `href` value to include the `pages/` directory since that is the new location of the about file *relative* to the index file.

```
1 <body>
2   <h1>Homepage</h1>
3   <a href="pages/about.html">About</a>
4 </body>
```

Refresh the index page in the browser and try clicking the about link again, it should now be back in working order.

In many cases, this will work just fine; however, you can still run into unexpected issues with this approach. Prepending `./` before the link will in most cases prevent such issues. By adding `./` you are specifying to your code that it should start looking for the file/directory *relative* to the `current` directory.

```
1 <body>
2   <h1>Homepage</h1>
3   <a href="./pages/about.html">About</a>
4 </body>
```

A metaphor

Absolute and relative links are a tricky concept to build a good mental model of, a metaphor may help:

Think of your domain name (`town.com`) as a town, the directory in which your website is located (`/museum`) as a museum, and each page on your website as a room in the museum (`/museum/movie_room.html` and `/museum/shops/coffee_shop.html`). Relative links like `./shops/coffee_shop.html` are directions from the current room (the museum movie room `/museum/movie_room.html`) to another room (the museum shop). Absolute links, on the other hand, are full directions including the protocol (`https`), domain name (`town.com`) and the path from that domain name (`/museum/shops/coffee_shop.html`):

`https://town.com/museum/shops/coffee_shop.html` .

Images

Websites would be fairly boring if they could only display text. Luckily HTML provides a wide variety of elements for displaying all sorts of different media. The most widely used of these is the image element.

To display an image in HTML we use the `` element. Unlike the other elements we have encountered, the `` element is a void element. As we have seen earlier in the course, void elements do not need a closing tag because they are naturally empty and do not contain any content.

Instead of wrapping content with an opening and closing tag, it embeds an image into the page using a `src` attribute which tells the browser where the image file is located. The `src` attribute works much like the `href` attribute for anchor tags. It can embed an image using both absolute and relative paths.

For example, using an absolute path we can display an image located on The Odin Project site:



To display images on your website that are hosted on your own web server, you can use a relative path.

► Linux, macOS, ChromeOS

► WSL2

Finally add the image to the `index.html` file:

```
1 <body>  
2   <h1>Homepage</h1>  
3   <a href="https://www.theodinproject.com/about">About The Odi  
4  
5   <a href="./pages/about.html">About</a>  
6  
7     
8 </body>
```

Save the `index.html` file and open it in a browser to view Charles in all his glory.

Parent directories

What if we want to use the dog image in the about page? We would first have to go up one level out of the pages directory into its parent directory so we could then access the images directory.

To go to the parent directory we need to use two dots in the relative filepath like this: `../`. Let's see this in action, within the body of the `about.html` file, add the following image below the heading we added earlier:

```
1 | 
```

To break this down:

1. First, we are going to the parent directory of the pages directory which is `odin-links-and-images`.
2. Then, from the parent directory, we can go into the `images` directory.
3. Finally, we can access the `dog.jpg` file.

Using the metaphor we used earlier, using `../` in a filepath is kind of like stepping out from the room you are currently in to the main hallway so you can go to another room.

Alt attribute

Besides the `src` attribute, every image element must also have an `alt` (alternative text) attribute.

The `alt` attribute is used to describe an image. It will be used in place of the image if it cannot be loaded. It is also used with screen readers to describe what the image is to visually impaired users.

This is how the The Odin Project logo example we used earlier looks with an `alt` attribute included:

As a bit of practice, add an `alt` attribute to the dog image we added to the `odin-links-and-images` project.

Image size attributes

While not strictly required, specifying height and width attributes in image tags helps the browser layout the page without causing the page to jump and flash.

It is a good habit to always specify these attributes on every image, even when the image is the correct size or you are using CSS to modify it.

Here is our Odin Project logo example with height and width attributes included:

Continúe y actualice el `odin-links-and-images` proyecto con los atributos de ancho y alto en la imagen del perro.

Asignación

1. [Vea el vídeo de enlaces HTML](#) de Kevin Powell .
2. [Vea el vídeo de imágenes HTML](#) de Kevin Powell .
3. [Vea el vídeo sobre la estructura de archivos](#) de Kevin Powell .
4. [Lea sobre los cuatro formatos de imagen principales que se pueden utilizar en la web](#) .

Comprobación de conocimientos

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

- [¿Qué elemento se utiliza para crear un enlace?](#)
- [¿Qué es un atributo?](#)
- [¿Qué atributo le dice a los enlaces a dónde dirigirse?](#)
- [¿Qué consideraciones de seguridad deben tenerse en cuenta si desea utilizar el atributo de destino para abrir enlaces en una nueva pestaña/ventana?](#)
- [¿Cuál es la diferencia entre un enlace absoluto y uno relativo?](#)
- [¿Qué elemento se utiliza para mostrar una imagen?](#)
- [¿Qué dos atributos deben tener siempre las imágenes?](#)
- [¿Cómo acceder a un directorio padre en una ruta de archivo?](#)
- [¿Cuáles son los cuatro formatos de imagen principales que puedes utilizar para las imágenes en la web?](#)

Recursos adicionales

Esta sección contiene enlaces útiles a contenido relacionado. No es obligatoria, por lo que se la puede considerar complementaria.

- [Internet es un tratamiento complicado para los enlaces e imágenes HTML](#)

- [¿Qué pasó el día que Google decidió que los enlaces que incluían \(/ \) eran malware?](#)
- [Cuándo utilizar target="_blank" de Chris Coyier en CSS-Tricks](#)
- Si desea profundizar su conocimiento sobre los distintos formatos de imagen que se utilizan en la web, [el siguiente artículo titulado: ¿Cuál es el mejor formato de imagen para su sitio web?](#) de imagekit.io es un gran recurso. Ofrece una comparación detallada de los formatos JPEG, PNG, GIF y WebP, lo que lo ayudará a elegir el adecuado para sus necesidades. Tenga en cuenta que el artículo no cubre SVG, pero sigue siendo una excelente guía para los demás formatos comunes.

[Mejorar en GitHub](#)[Informar un problema](#)[Ver registro de cambios de la lección](#)[Ver curso](#)[Marcar como completado](#)[Próxima lección](#)

¡Apóyanos!

El Proyecto Odin está financiado por la comunidad. ¡Únase a nosotros para ayudar a estudiantes de todo el mundo apoyando el Proyecto Odin!

[Más información](#)[Dona ahora](#)

**THE ODIN PROJECT**

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



Sobre nosotros

[Acerca de](#)

[Equipo](#)

[Blog](#)

[Casos de éxito](#)

Apoyo

[Preguntas frecuentes](#)

[Contribuir](#)

[Contáctenos](#)

Guías

[Guías de la comunidad](#)

[Guías de instalación](#)

Legal

[Términos](#)

[Privacidad](#)

© 2025 El Proyecto Odin. Todos los derechos reservados.