**Linking**

1. **Linking to Other Web Pages**

You can add links to a web page by adding an *anchor* element <a> and including the **text of the link** in between the opening and closing tags.

Then, the actual link to the page will be included as an attribute. The *hyperlink reference* should be included in the href attribute. This is used to link to a *path*, or the address to where a file is located (whether it is on your computer or another location).

<a href="https://www.wikipedia.org/">This Is A Link To Wikipedia</a>

**Opening Links in a New Tab**

To set the link to open a new tab when clicked, we can add the target attribute to the *anchor*.

For a link to open in a new window, the target attribute requires a value of \_blank. The target attribute can be added directly to the opening tag of the anchor element, just like the href attribute.

<a href="https://en.wikipedia.org/wiki/Brown\_bear" target="\_blank">The Brown Bear</a>

1. **Linking to Internal Pages**

Now we’ve known about linking with external webpage. However, many sites also link to internal web pages like Home, About, Contact.

First, let’s **establish where our files are stored.** When making multi-page static websites, web developers often store HTML files in the ***root directory***, or a main folder where all the files for the project are stored. Normally, HTML files are stored in the same folder.

project-folder/  
|—— about.html  
|—— contact.html //all 3 files in the same folder  
|—— index.html

If the browser is currently displaying **index.html**, it also knows that **about.html** and **contact.html** are in the same folder. Therefore, we can link web pages together using a *relative path*. The path is still in href.

<a href=**"./contact.html"**>Contact</a>

In this example, the <a> tag is used with a relative path to link from the current HTML file to the **contact.html** file in the same folder. The “**./**” tells the browser to look for the file in the current folder.

1. **Linking Using any Element**

HTML allows you to turn nearly any element into a link by wrapping that element with an anchor element. With this technique, it’s possible to turn images into links by simply wrapping the <img> element with an <a> element.

<a href="https://www.wikipedia.org/"><img src="https://www.Prickly\_Pear\_Closeup.jpg" alt="A red prickly pear fruit"/></a>

Notice how the text position is now replaced with the <img/> element. This means that any element at that position will be turned into a link.

1. **Linking to Same Page**

Say we want to make it easier to jump to different portions of our page. We do this by changing the hyperlink in href to become the id of the targeted section.

<p id="top">This is the top of the page!</p>  
<h1 id="bottom">This is the bottom! </h1>

The id attribute can be added to most elements on a webpage.

Now that the targeted section/element has the id attribute, we can link to that target through href by using a string containing the # character and the target element’s id.

<ol>  
  <li><a href=**"#top"**>Top</a></li>  
  <li><a href=**"#bottom"**>Bottom</a></li>  
</ol>