HTML. Unit 3. Hyperlinks.

1. What is a hyperlink

Hyperlinks are one of the most important innovations the Web has to offer. They've been a feature of the Web since the beginning, and they are what makes the Web a web. Hyperlinks allow us to link documents to other documents or resources, link to specific parts of documents, or make apps available at a web address. Almost any web content can be converted to a link so that when clicked or otherwise activated the web browser goes to another web address. These links are also known as URL's, or Uniform Resource Locators, text strings that specify where a resource (such as a web page, image, or video) can be found on the Internet.

A URL can point to HTML files, text files, images, text documents, video and audio files, or anything else that lives on the Web. If the web browser doesn't know how to display or handle the file, it will ask you if you want to open the file (in which case the duty of opening or handling the file is passed to a suitable native app on the device) or download the file (in which case you can try to deal with it later on).

For example, the BBC homepage (https://www.bbc.com/) contains many links that point not only to multiple news stories, but also different areas of the site (navigation functionality), login/registration pages (user tools), and more.

2. The <a> element

The `<a>` element (or anchor element) with its `href` attribute, creates a hyperlink to web pages, files, email addresses, locations in the same page, or anything else a URL can address. Content within each `<a>` should indicate the link's destination.

Links also provide implicitly some information about the navigation activity, as they will appear as follows in all browsers by default:

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

A basic link is created by wrapping the text or other content, inside an `<a>` element and using the `href` attribute, also known as a **Hypertext Reference**, or **target**, that contains the web address:

```
<p>
    I'm creating a link to
    <a href="https://www.mozilla.org/en-US/">the Mozilla homepage</a>.
</p>
```

The code above gives as the following result: I’m creating a link to the Mozilla homepage.

3. Adding information with the title attribute

Another attribute you may want to add to your links is `title`. The `title` contains additional information about the link, such as which kind of information the page contains, or things to be aware of on the web site:

```html
<p>
    I'm creating a link to
    <a href="https://www.mozilla.org/en-US/"
       title="The best place to find more information about Mozilla's mission and how to contribute">the Mozilla homepage</a>.
</p>
```

This gives us the following result and hovering over the link displays the title as a tooltip: I’m creating a link to the Mozilla homepage.

Keep in mind that a link title is only revealed on mouse hover, which means that people relying on keyboard controls or touchscreens to navigate web pages will have difficulty accessing title information. If a title’s information is truly important to the usability of the page, then you should present it in a manner that will be accessible to all users, for example by putting it as a regular text.

3.1. Proposed exercise: Links with information

> *Create a web page containing the information below. Use a main* `<h1>` *heading, a couple of paragraphs, and links to the pages in the list.*
>
> — Note: you must also use the `title` attribute to display information about each link on mouse hover.

- https://en.wikipedia.org/wiki/Castle_of_Santa_Bárbara (Santa Bárbara Castle is a fortification in the center of Alicante, Spain)
- https://en.wikipedia.org/wiki/Port_of_Alicante (The Port of Alicante is a seaport in Alicante, Spain on the Mediterranean Sea used for commercial and passenger traffic)
- https://en.wikipedia.org/wiki/Santa_Bárbara_Castle (Santa Bárbara Castle is a fortification in the center of Alicante, Spain)
- https://en.wikipedia.org/wiki/Mount_Benacantil (Mount Benacantil is a mount (or rather a large hill) that dominates the urban part of Alicante, and is the characteristic image of the city)

Alicante

Amongst the most notable features of the city are the Castle of Santa Bárbara, which sits high above the city, and the port of Alicante. The latter was the subject of bitter controversy in 2006-2007 as residents battled, successfully, to keep it from being changed into an industrial estate.

The Santa Bárbara castle is situated on Mount Benacantil, overlooking the city. The tower (*La Torreta*) at the top, is the oldest part of the castle, while part of the lowest zone and the walls were constructed later in the 18th century.

4. URLs and paths

To fully understand link targets, you need to understand URLs and file paths. This section gives you the information you need to achieve this.

A URL, or Uniform Resource Locator is simply a string of text that defines where something is located on the Web. For example, Mozilla's English homepage is located at `https://www.mozilla.org/en-US/`.

URLs use paths to find files. Paths specify where the file you're interested in is located in the filesystem. Let's look at an example of a directory structure, see the creating-hyperlinks directory.

The **root** of this directory structure is called `creating-hyperlinks`. When working locally with a web site, you'll have one directory that contains the entire site. Inside the **root**, we have an `index.html` file and a `contacts.html`. In a real website, `index.html` would be our home page or landing page (a web page that serves as the entry point for a website or a particular section of a website.).

There are also two directories inside our root — `pdfs` and `projects`. These each have a single file inside them — a PDF (`project-brief.pdf`) and an `index.html` file, respectively. Note that you can have two `index.html` files in one project, as long as they're in different filesystem locations. The second `index.html` would perhaps be the main landing page for project-related information.

4.1. Same directory

If you wanted to include a hyperlink inside `index.html` (the top level `index.html`) pointing to `contacts.html`, you would specify the filename that you want to link to, because it's in the same directory as the current file. The URL you would use is `contacts.html`:

```
<p>Want to contact a specific staff member?
Find details on our <a href="contacts.html">contacts page</a>.</p>
```

4.2. Moving down into subdirectories

If you wanted to include a hyperlink inside `index.html` (the top level `index.html`) pointing to `projects/index.html`, you would need to go down into the `projects` directory before indicating the file you want to link to. This is done by specifying the directory's name, then a forward slash, then the name of the file. The URL you would use is `projects/index.html`:

```
<p>Visit my <a href="projects/index.html">project homepage</a>.</p>
```

4.3. Moving back up into parent directories

If you wanted to include a hyperlink inside `projects/index.html` pointing to `pdfs/project-brief.pdf` , you'd have to go up a directory level, then back down into the `pdf` directory. To go up a directory, use two dots — `..` — so the URL you would use is `../pdfs/project-brief.pdf` :

```
<p>A link to my <a href="../pdfs/project-brief.pdf">project brief</a>.</p>
```

You can combine multiple instances of these features into complex URLs, if needed, for example: `../../../complex/path/to/my/file.html` .

4.4. Proposed exercise: Navigation menu

> *Create an «index.html» file to link your pages together with a navigation menu to create a multi-page website. This index file will link to all of your pages you have created before. Have a look at the example below so that you can get an idea of how your source code should look like. Additionally, you must also use the* `title` *attribute to insert a description of what it is inside each file linked.*
>
> — Note: From now on, each time you create a new file for any exercise, you should link it from your «index.html» file, and of course, you must validate everything each time you change something in your code or your domain.

```
<h1>HTML exercises</h1>

<h2>Headings and paragraphs</h2>
```

```
<p><a href="headings_paragraphs_1.html">Headings and paragraphs example 1</p>
<p><a href="headings_paragraphs_2.html">Headings and paragraphs example 2</p>
...

<h2>Text formatting</h2>

<p><a href="text_formatting_1.html">Text formatting example 1</p>
<p><a href="text_formatting_2.html">Text formatting example 2</p>
...
```

5. Document fragments

It's possible to link to a specific part of an HTML document, known as a document fragment, rather than just to the top of the document. To do this you first have to assign an id attribute to the element you want to link to. It normally makes sense to link to a specific heading, so this would look something like the following:

```
<h2 id="mailing_address">Mailing address</h2>
```

Then to link to that specific `id`, you'd include it at the end of the URL, preceded by a hash symbol (`#`), for example:

```
<p>Want to write us a letter? Use our <a href="contacts.html#mailing_address">mailing address</a>.</p>
```

You can even use the document fragment reference on its own to link to another part of the same document:

```
<p>The <a href="#mailing_address">company mailing address</a> can be found at the bottom of this page.</p>
```

5.1. Proposed exercise: Table of contents

> *Create a page similar to this one: https://en.wikipedia.org/wiki/Alicante (you may choose any page you like containing a large table of contents). You must start with an index, which must be followed by several sections (at least 10), each one preceded with a heading and containing several paragraphs. Also insert several links in each paragraph, as it is done in Wikipedia (fill the `title` attribute too). Have a look at this example and do something similar on your own:*

```
<h1>Alicante</h1>

<p><a href="#geography">Geography</a></p>
<p><a href="#history">History</a></p>
...

<h2 id="geography">Geography</h2>

Alicante is located in the southeast of the <a
href="https://en.wikipedia.org/wiki/Iberian_Peninsula">Iberian Peninsula</a>, on the
shores of the <a
href="https://en.wikipedia.org/wiki/Mediterranean_Sea">Mediterranean Sea</a>.
...

<h2 id="history">History</h2>

The area around Alicante has been inhabited for over 7000 years. The first tribes
of <a href="https://en.wikipedia.org/wiki/Hunter-gatherer">hunter-
gatherers</a> moved down gradually from Central Europe between 5000 and 3000 BC.
Some of the earliest settlements were made on the slopes of <a
href="https://en.wikipedia.org/wiki/Mount_Benacantil">Mount Benacantil</a>.
...
```

6. Block level links

As mentioned before, almost any content can be made into a link, even block-level elements. If you have an image you want to make into a link, use the `<a>` element and reference the image file with the `<img>` element (we will learn more about using images in a future unit):

```
<a href="https://www.mozilla.org/en-US/">
  <img src="mozilla-image.png" alt="mozilla logo that links to the mozilla homepage">
</a>
```

6.1. Proposed exercise: Image as a link

> *Using the image https://iessanvicente.com/ies_san_vicente.jpg, create a link to the url https://iessanvicente.com so that when you click on the picture, you will jump to the web page of IES San Vicente.*
>
> — Note: you must use the full url of the image as the `src` attribute of the `<img>` element.

7. E-mail links

It is possible to create links or buttons that, when clicked, open a new outgoing email message rather than linking to a resource or page. This is done using the <a> element to `mailto:`.

In its most basic and commonly used form, a `mailto:` link simply indicates the email address of the intended recipient. For example:

```
<a href="mailto:nowhere@mozilla.org">Send email to nowhere</a>
```

This code results in a link that looks like this: Send email to nowhere.

In fact, the email address is optional. If you omit it and your href is simply «mailto:», a new outgoing email window will be opened by the user's email client with no destination address. This is often useful as «Share» links that users can click to send an email to an address of their choosing.

7.1. Specifying details

In addition to the email address, you can provide other information. In fact, any standard mail header fields can be added to the mailto URL you provide. The most commonly used of these are «subject», «cc», and «body» (which is not a true header field, but allows you to specify a short content message for the new email). Each field and its value is specified as a query term.

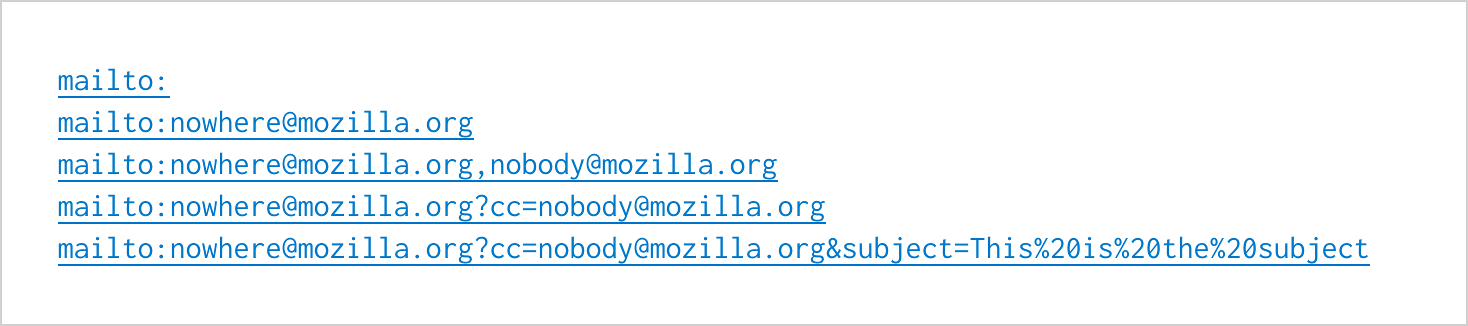
Here's an example that includes a cc, bcc, subject and body:

```
<a href="mailto:nowhere@mozilla.org?cc=name2@rapidtables.com&bcc=name3@rapidtables.com&subject=The%20subject%2
  Send mail with cc, bcc, subject and body
</a>
```

Keep in mind that the values of each field must be URL-encoded, that is with non-printing characters (invisible characters like tabs, carriage returns, and page breaks) and spaces percent-escaped. Also note the use of the question mark (`?`) to separate the main URL from the field values, and ampersands (&) to separate each field in the `mailto:` URL. This is standard URL query notation. Read The GET method to understand what URL query notation is more commonly used for.

7.2. Proposed exercise: E-mail links

> *Create a web page to display the following links (use paragraphs so that each link is shown in a different line):*



mailto:

mailto:nowhere@mozilla.org

mailto:nowhere@mozilla.org,nobody@mozilla.org

mailto:nowhere@mozilla.org?cc=nobody@mozilla.org

mailto:nowhere@mozilla.org?cc=nobody@mozilla.org&subject=This%20is%20the%20subject