**Head and Meta elements**

**Head Element:**

The head element is a container for all the meta-data about a web page that isn't actual content. It typically includes things like the page's title, links to stylesheets, and meta-data that search engines and social media platforms use to better understand and display the page.

Here's an example of how the head element might look:

<!DOCTYPE html>

<html>

<head>

<title>My Page</title>

<link rel="stylesheet" href="styles.css">

<meta name="description" content="This is my page about XYZ.">

</head>

<body>

<!-- page content goes here -->

</body>

</html>

In this example, the **head** element contains a **title** element, which specifies the page's title that appears in the browser's title bar. It also contains a **link** element that links to an external stylesheet, which is used to define the page's layout and styling. Finally, there's a **meta** element that provides a short description of the page's content, which can be used by search engines to display a summary of the page's content in search results.

**Meta Element**

The meta element is used to provide additional information about the web page that isn't directly visible on the page itself. It's typically used to provide information about the page's author, keywords that describe the page's content, and other meta-data that helps search engines and other applications understand and categorize the page.

Here's an example of how the meta element might look:

<!DOCTYPE html>

<html>

<head>

<title>My Page</title>

<meta name="author" content="John Doe">

<meta name="keywords" content="XYZ, web design, HTML">

</head>

<body>

<!-- page content goes here -->

</body>

</html>

In this example, the meta element includes two attributes: **name** and **content**.

The name attribute specifies what type of meta-data is being provided, while the content attribute provides the actual value of the meta-data. The first meta element specifies the page's author, while the second meta element provides keywords that describe the page's content.

**Let’s take a sneak peek of the content that IDE (like VS Code) generates:**

**<meta charset="UTF-8">**

**<meta http-equiv="X-UA-Compatible" content="IE=edge">**

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

These three meta elements are commonly found in the head section of an HTML document and are used ***to set character encoding***, ***define compatibility with Internet Explorer***, and ***specify the viewport of a web page.***

Here's a more detailed explanation of each element:

* **‘meta’ Charset Attribute**

The charset attribute specifies the character encoding for the HTML document. This is important because it determines how the browser interprets the text in the document. The UTF-8 character set is recommended for use on the web because it supports a wide range of characters from different languages.

* **‘meta’ Http-equiv Attribute**

The http-equiv attribute provides an HTTP header for the browser to use when rendering the page. One common use for this attribute is to set compatibility with Internet Explorer using the X-UA-Compatible value. This ensures that the page is rendered in the best way possible, based on the version of Internet Explorer being used.

* **‘meta’ Viewport Attribute**

The viewport attribute is used to specify the width and scale of the viewport of a web page, which is important for ensuring that the page is displayed properly on different devices with varying screen sizes. The width=device-width value sets the width of the viewport to the width of the device, while the initial-scale=1.0 value sets the initial zoom level to 100%.

**Creating Link with the <a> Element (or Link and Navigation)**

**Basic Links:**

* A link is specified using the <a> element. Anything between the opening <a> tag and the closing </a> tag becomes part of the link that users can click in a browser.
* As with all journeys, links have a starting point known as the *source*, and a finishing point known as the *destination;* in XHTML/HTML both points are called *anchors.*
* Each link that you see on a page that you can click is a *source anchor,* created using the <a> element. You can also use the <a> element to create markers in parts of your pages that allow you to link directly to that part of the page. These markers are called *destination anchors.*

**Linking to Other Pages:**

* To link to another web page, the opening <a> tag must carry an attribute called href; the value of the href attribute is the name of the file you are linking to.

Example:

<body>

<p> Return to the <a href = ”https://www.google.com”> home page </a>.<p>

</body>

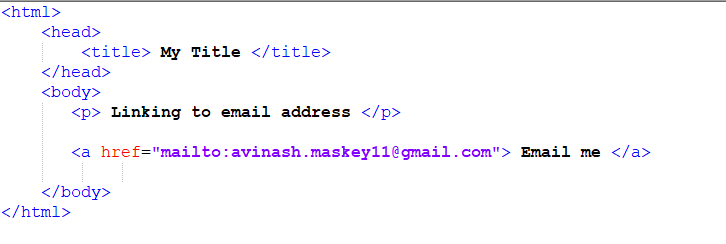
* As you can see, the value of the href attribute is what you type into a browser if you wanted to visit the google website; the full web address is often referred to as URL (Uniform Resource Locator).

Note: When creating any link, you should try to make it concise and use words that let people know what they will see if they click on the link. One reason for this is that links are usually presented in a different color than the surrounding text, which makes them stick out more than the text around them. As a result, many people *scan* pages for links when they want to go to the next page without really reading the entire page. Therefore, people are more likely to keep exploring your web site if the links are easy to read and have a better explanation than just “click here.”

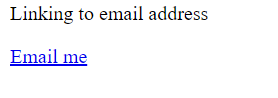
**Linking to Email Addresses:**

* You’ ve probably seen a link that shows an e - mail address, which when clicked on opens a new e - mail in your e - mail program, ready for you to send an e - mail to that address.

Example/Source Code:



Output:



Here, in the above example the value of href attribute starts with the keyword mailto, followed by a colon, and then the e - mail address you want the mail sent to. As with any other link, the content of the <a> element is the visible part of the link shown in the browser.

**Creating Source Anchor with the href Attribute:**

* The source anchor is what most people think of when talking about links on the Web — whether the link contains text or an image. It is something you can click expecting, to be taken somewhere else.
* As you have already seen, any text contained between the opening <a> tag and closing </a> tag forms part of the link that a user can click on. The URL the user should be taken to is specified as the value of the href attribute.

Note: Examples to this topic has already been discussed. Try it on your own.

**Creating a Destination Anchor Using the name and id Attributes (Linking to a Specific Part of a Page):**

* If you have a long web page, you might want to link to a specific part of that page in order to save the user from having to scroll up and down the page to find the relevant part. The *destination anchor* allows the page author to mark specific points in a page that a source anchor can point to.
* Common examples of linking to a specific part of a page that you might have seen used on web pages include:
* “Back to top” links at the bottom of a long page
* A list of contents on a page that takes the user to the relevant section of that page.
* Links within text to footnotes or definitions.

**The <a> Element’s Other Attribute:**

* The <a> element can carry several attributes that we have not yet met. While these attributes are not used as much as those covered up to this point, for completeness it is worth quickly looking at them.
* The <a> element supports all of the universal attributes, the UI event attributes, and the following attributes:

1. **The accesskey attribute:**

The accesskey attribute creates a keyboard shortcut that can be used to activate a link. For example, if you gave the accesskey attribute a value of t, when the user presses the T key along with either the Alt or Ctrl key (depending on the operating system), the link gets activated.

Example:

<a href="https://google.com" accesskey = "g" target="\_blank"> Google Page </a>

1. **The charset Attribute:**

* The charset attribute indicates the character encoding of the document the URL points to. It is typically used only when you are linking to a page in a different language that uses a different character set.
* The value must be a string that identifies the character set, such as UTF - 8 or ISO - 8859 - 1. This example links to a document in the Japanese character set:

<a href = ”http://www.amazon.co.jp/” charset=”ISO-2022-JP” > Amazon Japan < /a >

1. **The coords Attribute:**

The coords attribute is designed for use on a source anchor when it contains an image. It allows you to create something known as an *image map,* which is where different parts of the same image link to different places. The value of the coords attribute will be a set of *x* and *y* coordinates that indicates which part of the image should follow the link.

1. **The hreflang Attribute:**

The hreflang attribute indicates which language the page you are linking to is written in. It’s designed to be used when linking to a page in a different language from the current document, and the value of this attribute is a two-letter *language code.*

Example:

<a href=”http://www.amazon.co.jp/” hreflang=”JA” > Amazon Japan < /a >

1. **The rel Attribute:**

The rel attribute is used on the source anchor to indicate the relationship between the current document and the resource specified by the href attribute. The major browsers do not currently make any use of this attribute, although it is possible that automated applications could be written to use this information.

For example, the following link uses the rel attribute to indicate that its destination is a glossary of terms used in the document:

<p>For more information, please read the <a href=”#glossary” rel=”glossary”

>glossary</a>

1. **The rev Attribute:**

The rev attribute provides the same role as the rel attribute but is used on the destination anchor to describe the relation between the destination and the source. It is currently not supported by major browsers (Not supported in HTML5).

1. **The shape Attribute:**

If you want to create an image map, the shape attribute can be used to indicate the shape of an area that becomes a clickable *hotspot*.

1. **The tabindex Attribute:**

* To understand the tabindex attribute, you need to know what it means for an element to gain *focus*. Any element that a user can interact with can gain focus. If the user clicks the Tab key on his or her keyboard when a page has loaded, the browser moves focus between the parts of the page that the user can interact with. The parts of the page that can gain focus include links and some parts of forms (such as the boxes that allow you to enter text). When a link receives focus, and the user presses Enter on the keyboard, the link is activated.
* You can see focus working on the Google web site; if you repeatedly press the Tab key, you should see focus pass between links on the page. After it has passed across each link in turn, it goes onto the box

where you enter search terms, across the site’ s buttons, and usually ends up back where you typed in the URL. Then it cycles around the same elements again as you keep pressing Tab.

1. **The target Attribute:**

By default, when you use the <a> element to create a link, the document you are linking to will open in the same browser window. If you want the link to open in a new browser window, you can use the target attribute with a value of \_blank or \_parent or \_self or \_top.

Example:

<a href = ”Page2.html” target=”\_blank” > Page 2 < /a >

1. **The title Attribute:**

It is good to use a title attribute on any links that contain images. It can also help provide additional information to visitors in the form of a visual text tooltip in most browsers or an auditory clue in voice browsers for the visually impaired.

Example:

<a href=”https://google.com” title=”visit google”>Click here</a>

1. **The type Attribute:**

The type attribute specifies the MIME type of the link. MIME types can be compared to file extensions, but are more universally accepted across different operating systems. For example, an HTML page would have the MIME type text/html, whereas a JPEG image would have the MIME type img/jpeg. The following is an example of the type attribute being used to indicate that the document the link points to is an HTML document:

<a href = ”index.html” type=”text/html”>Index</a>

**Relative Vs Absolute URL**

In HTML, URLs (Uniform Resource Locators) are used to specify the location of resources such as web pages, images, or files. There are two main types of URLs: relative URLs and absolute URLs. The key difference between them lies in how they define the location of a resource.

**Relative URLs**

A relative URL specifies the location of a resource relative to the current document or the current location within the document hierarchy. It does not include the complete address of the resource. Relative URLs are typically shorter and more flexible, as they can adapt to different environments. They are commonly used for linking to internal resources within a website.

Relative URLs can take various forms:

* **Relative to the current document:** A relative URL that refers to a resource within the same directory or subdirectory as the current document. For example, if the current document is in the "articles" directory and you want to link to an image in the same directory, the relative URL could be "image.jpg".
* **Relative to the root directory**: A relative URL that starts with a forward slash ("/") and refers to a resource relative to the root directory of the website. For example, "images/image.jpg" refers to an image located in the "images" directory at the root level of the website.
* **Relative to the parent directory:** A relative URL that uses ".." to refer to a resource in the parent directory. For example, if the current document is in the "articles" directory and you want to link to a document in the parent directory, the relative URL could be "../parent.html".

Relative URLs are particularly useful when you want to create links within your website or when you need to move or migrate your website to a different domain or directory structure.

**Absolute URLs**

An absolute URL provides the complete web address or path to a resource, including the protocol (e.g., "http://" or "https://") and the domain name. Absolute URLs specify the exact location of a resource on the web. They are typically used when linking to external resources or when you need to reference a specific resource with its complete address.

Here are a few examples of absolute URLs:

* Absolute URL to an external website: "https://www.example.com"
* Absolute URL to an image on another website: "https://www.example.com/images/image.jpg"
* Absolute URL to a file on a server: "http://www.example.com/files/document.pdf"

Absolute URLs are required when linking to resources on different websites or when you want to ensure that a specific resource is always referenced, regardless of the current document's location.