·        A **block-level element** wouldn't be nested inside an inline element, but it might be nested inside another block-level element.

·        **Void elements** only have a start tag; end tags must not be specified for void elements.

·        ***Self-closing tags*** *(<tag />) do not exist in HTML.*

·        Some code formatters add the trailing slash character to the start tags of void elements to make them XHTML-compatible and more readable.

·        **Boolean attributes** can only have one value, which is generally the same as the attribute name.

·        **Metadata** is data that describes data.

·        **utf-8** is a *universal character set* that includes pretty much any character from any human language.

·        Many **<meta>** elements include name and content attributes:

name specifies the type of meta element it is; what type of information it contains.

content specifies the actual meta content.

·        Sitelinks **are links from the same domain that are clustered together under a web result**.

* **favicon** (short for "favourites icon")
* If your site uses **a Content Security Policy (CSP)** to enhance its security, the policy applies to the favicon. If you encounter problems with the favicon not loading, verify that the [Content-Security-Policy](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy) header's [img-src directive](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy/img-src) is not preventing access to it.
* The [<link>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/link) element should always go inside the head of your document. This takes two attributes, **rel="stylesheet",** which indicates that it is the document's stylesheet, and **href**, which contains the path to the stylesheet file:
* The [<script>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/script) element should also go into the head, and should include a **src** attribute containing the path to the JavaScript you want to load, and **defer**, which basically instructs the browser to load the JavaScript after the page has finished parsing the HTML.
* **<html lang="en-US">** Your HTML document will be indexed more effectively by search engines if its language is set.
* A link title is only revealed on mouse hover

·        [<b>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/b), [<i>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/i), and [<u>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/u) -- elements like this, which only affect presentation and not semantics, are known as **presentational elements.**

·        A URL, **or Uniform Resource Locator** is a string of text that defines where something is located on the Web.

·         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.).

·        It's possible to link to a specific part of an HTML document, known as a **document fragment**, to do this you first have to assign an [id](https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes#id) attribute to the element you want to link to.

·        An **absolute URL** will always point to the same location, no matter where it's used.

·        When you are linking to a resource that's to be downloaded rather than opened in the browser, you can use the download attribute to provide a default save filename.

·        It is permitted to have a single term with multiple descriptions, in (description list).

·        For **block level element** quotation use **<blockquote>** element and for **inline level** element quotation us <q> element.

·        **Citations** are styled in italic font by default.

·        **Non -semantic wrapper** elements are *<div> and <span>.*

·        [<span>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/span) is an **inline non-semantic element**, which you should only use if you can't think of a better semantic text element to wrap your content, or don't want to add any specific meaning.

·      **<hr>**: the *thematic break element*

·        [void element](https://developer.mozilla.org/en-US/docs/Glossary/Void_element) (meaning, it cannot have any child content and cannot have an end tag)

·        **Search engines** also read image filenames and count them towards SEO. Therefore, you should give your image a descriptive filename; dinosaur.jpg is better than img835.png.

·        **Never** point your src attribute at an image hosted on someone else's website **without permission**. This is called "hotlinking".

·        Elements like [<img>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/img) and [<video>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/video) are sometimes referred to as **replaced elements**. This is because the element's content and size are defined by an external resource (like an image or video file), not by the contents of the element itself.

·        A **replaced element** is an element whose contents are not affected by the current document's styles. The position of the replaced element can be affected using CSS, but not the contents of the replaced element itself.

·        Some replaced elements, such as [<iframe>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/iframe) elements, may have stylesheets of their own, but they don't inherit the styles of the parent document.

·        From an accessibility viewpoint, captions and [alt](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/img#attr-alt) text have distinct roles. Captions benefit even people who can see the image, whereas [alt](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/img#attr-alt) text provides the same functionality as an absent image.

·        **CSS background images** are for decoration only, such images have no semantic meaning at all. They can't have any text equivalents, are invisible to screen readers, and so on.

·        A **MIME type** (now properly called "media type", but also sometimes "content type") is a string sent along with a file indicating the type of the file. It serves the same purpose as filename extensions traditionally do on Windows.

·        **"Transcribe"** *means "to write down spoken words as text."* The resulting text is a "transcript."

·        [**Clickjacking**](https://developer.mozilla.org/en-US/docs/Glossary/Clickjacking) is one kind of common iframe attack where hackers embed an invisible iframe into your document (or embed your document into their own malicious website) and use it to capture users' interactions.

·        On the web, you'll work with two types of images — **raster images**, and **vector images**:

·        **Raster images** are defined using a grid of pixels — include *Bitmap (.bmp), PNG (.png), JPEG (.jpg), and GIF (.gif.)*

·        **Vector images** are defined using algorithms — a vector image file contains shape and path definitions that the computer can use to work out what the image should look like when rendered on the screen.

·        **Vector image files** are much lighter than their raster equivalents, because they only need to hold a handful of algorithms, rather than information on every pixel in the image individually.

* HTML is a presentation language, whereas XML is a data-description language.
* SVG is for marking up graphics, not content.
* Multiple resolutions would be made available to the user's web browser. The browser could then determine the optimal resolution to load based on the screen size of the user's device. This is called the resolution switching problem.
* The proper layout containers (such as <header>, <section>, <article>, or <div>), their width defaults to 100% of their parent element.
* HTML has a method of defining styling information for an entire column of data all in one place — the <col> and <colgroup> elements.
* Styling columns like this is limited to a few properties: border, background, width, and visibility. To set other properties you'll have to either style every <td> or <th> in the column, or use a complex selector such as :nth-child.