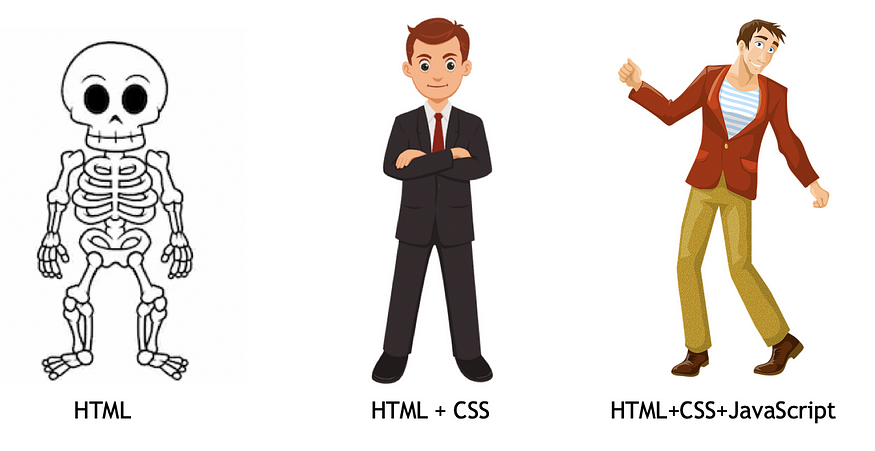
***HTML***

While HTML is a skeleton structure of a document, CSS beautifies it and Javascript makes it dynamic



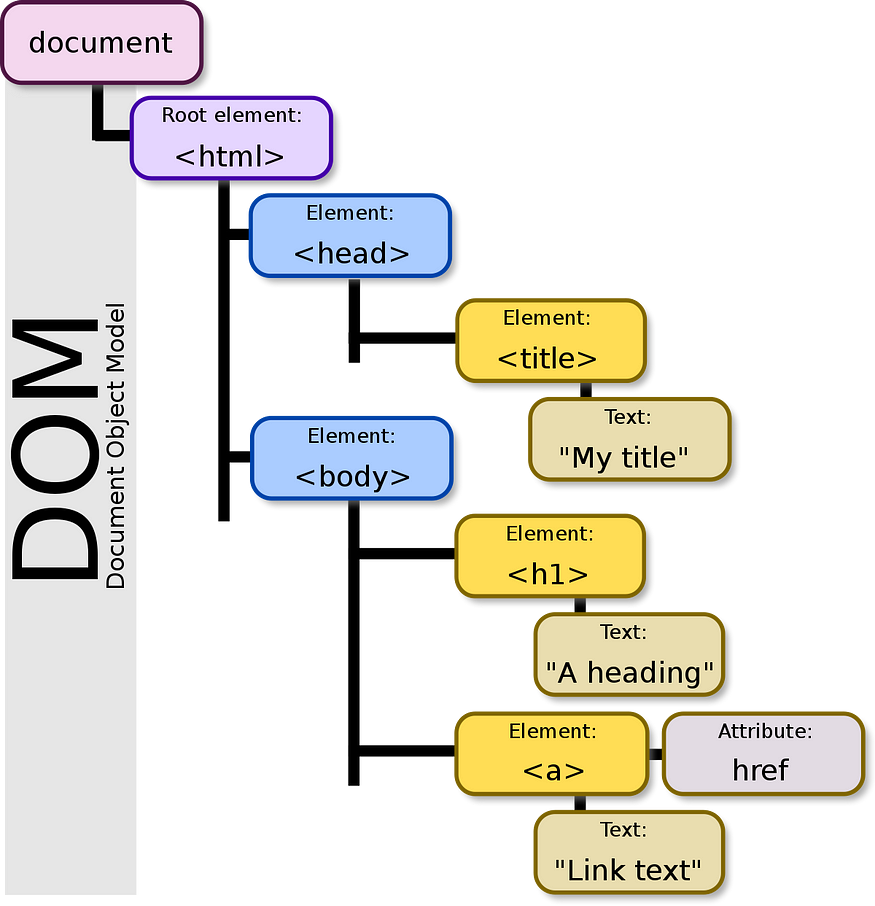
Advantages and Disadvantages of HTML:

|  |  |
| --- | --- |
| ***Advantages*** | ***Disadvantages*** |
| HTML is a light web programming language | It is difficult to read as it contains a lot of codewhich creates complexity |
| It can create static and plain web pages | Back-end languages such as PHP and JAVA are necessary to build dynamic websites and applications. |
| HTML is easy to use and analyze | We have to write a lot of code to create a webpage |
| It provides the structure of the webpage and CSS provides styling to the webpage. | HTML alone is insufficient for creating visually appealing websites. |
| All browsers support HTML. | It Takes time to load |

1. **HTML Document, DOM**

When an html document is loaded into a web browser, it becomes a document object.

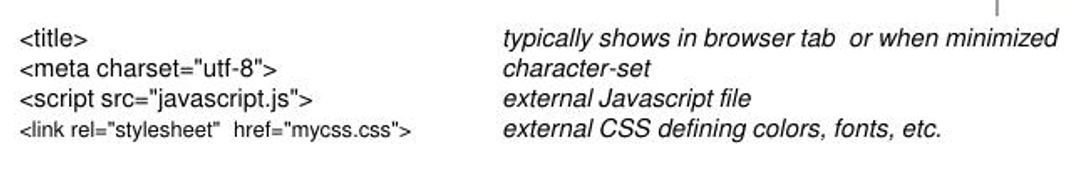
<!DOCTYPE html>  
<html>  
<head>  
<meta charset="UTF-8">  
<title>*Title of the document*</title>  
</head>  
  
<body>  
*Content of the document......*  
</body>  
  
</html>



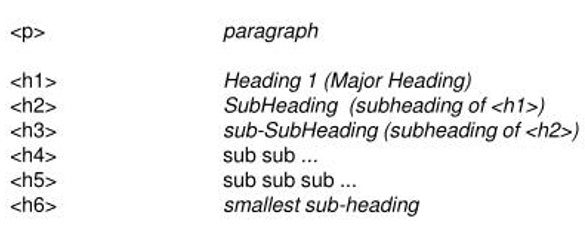
## Understanding HTML Tags:

HTML tags help web browsers convert HTML documents into web pages.

Common HTML tags in <head>:



Common HTML tags in <body>:



## Understanding HTML attributes:

* HTML attributes are special words which provide additional information about the elements.
* Each element or tag can have attributes, which defines the behaviour of that element.
* Attributes should always be applied with start tag.
* The Attribute should always be applied with its name and value pair.
* The Attributes name and values are case sensitive, and it is recommended to be written in Lowercase only.
* You can add multiple attributes in one HTML element, but need to give space between two attributes.

## Syntax:

## <element attribute\_name="value">content</element>

## <a> tag:

The <a> tag in HTML is used to create hyperlinks, which are clickable elements that navigate to another webpage, file, or location within the same webpage.

Here's how the <a> tag is structured:

<a href="URL">Link Text</a>

* href: The href attribute specifies the destination of the link. It can be a URL (web address), a file path, or an anchor within the same page.
* Link Text: The text or content that is displayed as the clickable link.

**Additional Attributes:**

The <a> tag can also have additional attributes:

target: Specifies where to open the linked document (\_self, \_blank, \_parent, \_top, or a custom frame name).

**\_self (default):** Opens the link in the same tab or window where the link was clicked.

**\_blank:** Opens the link in a new tab or window. This is commonly used when linking to external websites or documents.

**\_parent:** Opens the link in the parent frame, which is useful if you are working with nested frames.

**\_top:** Opens the link in the full body of the window, effectively breaking out of any frames.

title: Provides additional information about the link (often displayed as a tooltip).

<a href="https://www.example.com" target="\_blank" title="Visit Example Website">Example Website</a>

## Images and attributes in HTML:

* src: Specifies the URL of the image.
* alt: Specifies alternative text for the image, used by screen readers and displayed if the image fails to load.
* width: Specifies the width of the image in pixels.
* height: Specifies the height of the image in pixels.
* title: Specifies a title for the image, typically displayed as a tooltip when the user hovers over the image.

## Understanding HTML elements:

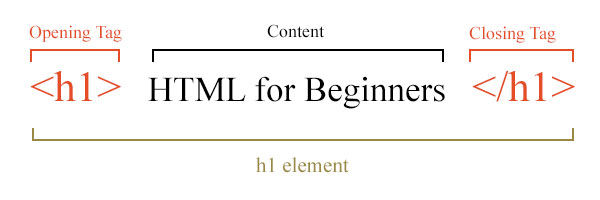
## An element is a collection of start tag, attributes, end tag, content between them.

## The diagram below displays an HTML header element. As we can see, the header element is made up of:

## An opening tag (<h1>)

## The content (“HTML for Beginners” text)

## A closing tag (</h1>)



**5.a. Nested Elements**: HTML allows to keep one element inside another element.

**5.b. Empty Elements**: Contains only opening tag. Do not require content and closing tag.

* <br>: Represents a line break.
* <hr>: Represents a thematic break or horizontal rule.
* <img>: Embeds an image into the document.
* <meta>: Provides metadata about the HTML document.

**5.c. Block level Elements**:

In HTML, block-level elements are elements that typically start on a new line and stretch out to the left and right as far as they can. They form the primary building blocks of the document's layout, and they typically contain inline elements and other block-level elements. Block-level elements are often used for major structural elements of the page, such as headings, paragraphs, lists, navigation menus, and divisions.

Here are some common block-level elements in HTML:

* <div>: The generic block-level container used for grouping elements and applying styles.
* <p>: Represents a paragraph of text.
* <hr>: Horizontal line
* <h1>, <h2>, <h3>, <h4>, <h5>, <h6>: Heading elements, with <h1> being the highest level and <h6> being the lowest.
* <ul>: Defines an unordered list.
* <ol>: Defines an ordered list.
* <li>: Represents a list item within <ul> or <ol> elements.
* <nav>: Defines a section of navigation links.
* <header>: Represents the header of a document or a section.
* <footer>: Represents the footer of a document or a section.
* <section>: Represents a thematic grouping of content.
* <article>: Represents an independent piece of content that can stand alone.
* <main>: Represents the main content of the document.

Block-level elements typically accept width, height, margin, and padding properties, and they naturally create a visual separation between content sections. By default, block-level elements have a line break before and after them in the document flow.

In contrast, inline elements do not start on a new line and only take up as much width as necessary. They are often used within block-level elements to style text or create small content elements within larger blocks.

**5.c. Inline Elements:**

Inline elements in HTML are elements that do not start on a new line and only take up as much width as necessary. Unlike block-level elements, inline elements can appear within a block-level element or other inline elements, and they typically do not create a line break before or after themselves. Inline elements are often used to style and format smaller pieces of content within larger blocks of text.

Here are some common inline elements in HTML along with their purposes:

* <span>: The generic inline container used to apply styles or manipulate text within a larger block of content.
* <a>: Defines a hyperlink, allowing users to navigate to another web page or resource.
* <strong>: Indicates strongly emphasized text, typically rendered as bold by default.
* <em>: Indicates text that is emphasized, typically rendered as italic by default.
* <img>: Embeds an image within the text flow.
* <br>: Inserts a line break within the text flow.
* <input>: Defines an input control within a form, such as text input, checkboxes, or radio buttons.
* <label>: Associates a label with a form control, providing a description or instruction for the control.
* <span>: Used for grouping inline elements or applying styles to specific parts of text.
* <code>: Renders text as computer code, typically in a monospaced font.

These elements can be combined and nested within each other to create rich and expressive content within HTML documents. Inline elements are commonly used for styling individual words or phrases, adding links and images to text, creating form elements, and highlighting specific portions of content.

Understanding the distinction between block-level and inline elements is crucial for structuring HTML documents effectively and applying appropriate styles and formatting to different parts of the content.

Top of Form

1. **HTML Lists:**

HTML Lists are used to specify lists of information. All lists may contain one or more list elements. There are three different types of HTML lists:

1. Ordered List or Numbered List (ol) - all the list items are marked with numbers by default.
2. Unordered List or Bulleted List (ul) - all the list items are marked with bullets.
3. Description List or Definition List (dl)

**Definition List:**

This type of list is typically used when you have a term and its corresponding definition .

The HTML definition list contains following three tags:

1. <dl> tag defines the start of the list.
2. <dt> tag defines a term.
3. <dd> tag defines the term definition (description).

**Semantic Elements:**

Semantic elements in HTML are elements that carry meaning. They are used to define the structure of a web page and convey the purpose of the content to both the browser and the developer. Semantic elements provide context and clarity to the content, making it more understandable for search engines, assistive technologies, and other developers who may work on the project.

Some commonly used semantic elements in HTML5 include:

<header>: Defines a header for a document or section. Typically contains introductory content, such as headings, logos, and navigation menus.

<nav>: Represents a section of navigation links. Used for navigation menus, table of contents, or other sets of links to navigate the document or related documents.

<main>: Specifies the main content of the document. It should not contain content that is repeated across multiple pages, such as headers, footers, or navigation menus.

<section>: Defines a section of related content. It groups together thematic content, such as chapters, headers, or footers.

<article>: Represents a self-contained piece of content that can be distributed or reused independently, such as blog posts, news articles, or comments.

<aside>: Defines content aside from the main content, such as sidebars, pull quotes, or tangentially related information.

<footer>: Defines the footer of a document or section. Typically contains metadata, copyright information, contact details, or links to related content.

<figure> and <figcaption>: Used together to represent self-contained content, such as images, diagrams, illustrations, or code snippets, along with captions.

<details> and <summary>: Defines additional details or disclosures that the user can view or hide. The summary represents a visible heading for the details.

Using semantic elements not only helps to structure the document logically but also improves accessibility, search engine optimization (SEO), and code maintainability. Semantic HTML communicates the meaning and purpose of content more effectively, leading to a better user experience and making the document easier to understand and maintain.