

Hyper text mark up language.

Released in 1993.

Standard text formatting language. (tells what and how to display)
that will be rendered.

HTML can only make the structure of the data that will be rendered. CSS, JS are needed to make it visually appealing.

1) Element: `<h1> heading1 </h1>`

Tag: $\langle h_1 \rangle, \langle 1/h_1 \rangle \dots$

2) Tags are the primary comp. of HTML defines how the content will be structured but attributes are like characteristics of the element.

`<p align = 'center'>` para `</p>`

↓ attribute

↓ tags.

↓ attribute

↓ tags

Void elements are those with no closing tags

Example: `
` `` `<hr/>`

→ A blank sequence of white space characters are treated as single
The Browser collapses multiple spaces helping reading easier.

HTML entities :

We can not represent few characters directly. To use them, HTML entities are helpful.

< f1t V#60

✓ Lamp ✓ #38

> Rgt 27#62

nbsp nbsp F# 160.

List



ordered unordered

 $\angle u$ or $\angle o$ $\angle l$ \angle /l
$$\langle u^2 \rangle \quad \langle 1/u \rangle \quad \langle 1/\sigma \rangle \text{ or } \langle u \rangle$$

Div - Block level element. (Starts in a new line)

span - inline element. Takes width which is only necessary.

class attribute is used to specify the class name for HTML element. Multiple elements can have same class name. This is basically used to add stylesheets written.

- class value can be same for multiple elements but id of one HTML element cannot be associated with another.

Enctype: This attribute specifies how data should be encoded while submitting it to the server.

This can be used only if method = "post"

This attribute can have 3 values,

1) application/x-www-form-urlencoded :

This is default one. All characters are encoded before sent.

spaces → '+' symbols and special characters are → Hex ASCII.

2) multipart/form-data:

This is used to send the file data to server side for processing.

Must attribute if user has to upload a file through form.

3) text/plain :

Sends data without any encoding. Not recommended.

HTML Layout

<header> starting info.

<footer> last section

<nav> Nav menu

<article> set of info.

<section>

<aside>

used inside article to define basic structure of page side bar content.

optimize website assets loading.

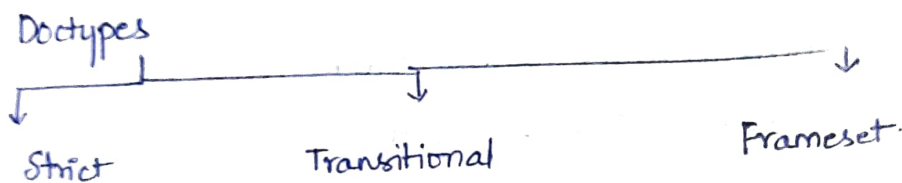
- 1) CDN Hosting : Content delivery Network Reduces latency.
- 2) File compression: reduces size of asset to reduce data transfer.
- 3) file concatenation: reduces num of HTTP calls.
- 4) Minify scripts: reduces overall size of css and js files.
- 5) Parallel downloads.
- 6) Lazy loading: instead of loading

Formatting tags: `` `<i>` `` → makes text italic but with added Semantics Importance.

`<big>` `<small>` `<sub>` `<sup>`

`` → strike out text

`` `<mark>` `<ins>` → added text.



character set is defined in `<meta>` `charset = "UTF-8"`

↓ specifies about the character encoding for HTML.

`` and `` have semantic meaning.

- head should always be inside HTML doc. It contains about the metadata of the page like meta, link, style, script (Not displayed) only 1 head tag is allowed, always before body tag.

- In body we will have all the possible content to be displayed on the webpage.

```
<html>
  <head> </head>
  <body> </body>
</html>
```

Nesting of webpages is allowed. `<iframe>` helps in achieving that.

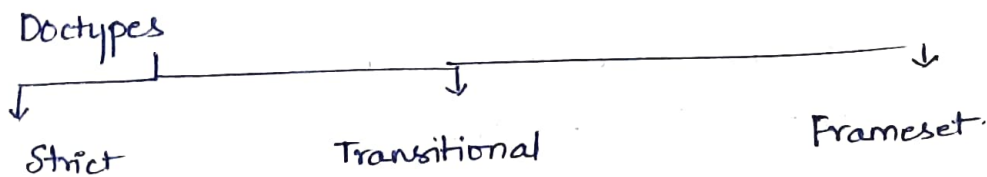
optimize website assets loading.

- 1) CDN Hosting : Content delivery Network. Reduces latency.
- 2) File compression : reduces size of asset to reduce data transfer.
- 3) file concatenation : reduces num of HTTP calls.
- 4) Minify scripts : reduces overall size of css and js files.
- 5) Parallel downloads :
- 6) Lazy loading : instead of loading

Formatting tags : `` `<i>` `` → makes text italic but with added Semantics importance.

`<big>` `<small>` `<sub>` `<sup>`

`` → strike out text `` `<mark>` `<ins>` → added text.



character set is defined in `<meta>` `charset = "UTF-8"`

↓ specifies about the character encoding for HTML.

`` and `` have semantic meaning.

- head should always be inside HTML doc. It contains about the metadata of the page. like meta, link, style, script (Not displayed) only 1 head tag is allowed, always before body tag.

- In body we will have all the possible content to be displayed on the webpage.

```
<html>
  <head> </head>
  <body> </body>
</html>
```

Nesting of webpages is allowed. `<iframe>` helps in achieving that.

Cell padding is about the space between text in the cell and edge/ border of the cell. and cell spacing is about the distance between two cells.

Rowspan, colspan helps in clubbing the rows.

— We can change an inline element to block by adding display attribute as block to the element.

Positioning an Element in HTML

- 1) Static - default. positions according to flow of doc.
- 2) Absolute - positioned relative to parent element.
- 3) fixed - Same as abs but w.r.t html element.
- 4) Relative - according to normal flow + relative to original pos.
- 5) Initial - Resets property to its default value.
- 6) Inherit - Inherits property from its parent.

Displaying an Element

- 1) inline
- 2) Block
- 3) inline Block - can format using height and width.
- 4) flex - flexbox
- 5) inline flex
- 6) grid - grid container
- 7) None. - hide the element.

— When the element has attribute Visibility. "hidden" this takes up space but display. "none" does not take any space on the webpage and hides the element.

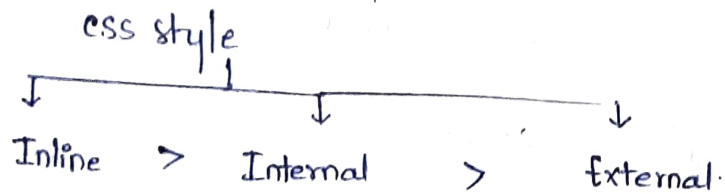
specifying link

` `

target: used to specify, where the linked document has to be opened.

target =

- self : default, opens in same tab
- blank : New tab
- parent : opens in parent frame
- top : opens doc in a full body window.



Anchor tag is used to create a hyper link to another web page or a part of a document. Link is used to define a link between an external resource and current doc which are not clickable.

Rules where to place script tags:

- 1) Lib/event scripts in the head section.
- 2) Normal scripts that do not write anything on page - head.
- 3) Scripts that render/write something are to be placed in bottom of the body section.

Forms: `<form>` used basically to collect user inputs.

`<input>` `<label>` `<checkbox>` `<radio>` `<submit>`

Events in HTML can be handled using JS and JQuery, like `onclick`, `onchange`, `ondrag`, etc.

Advantages of HTML5:

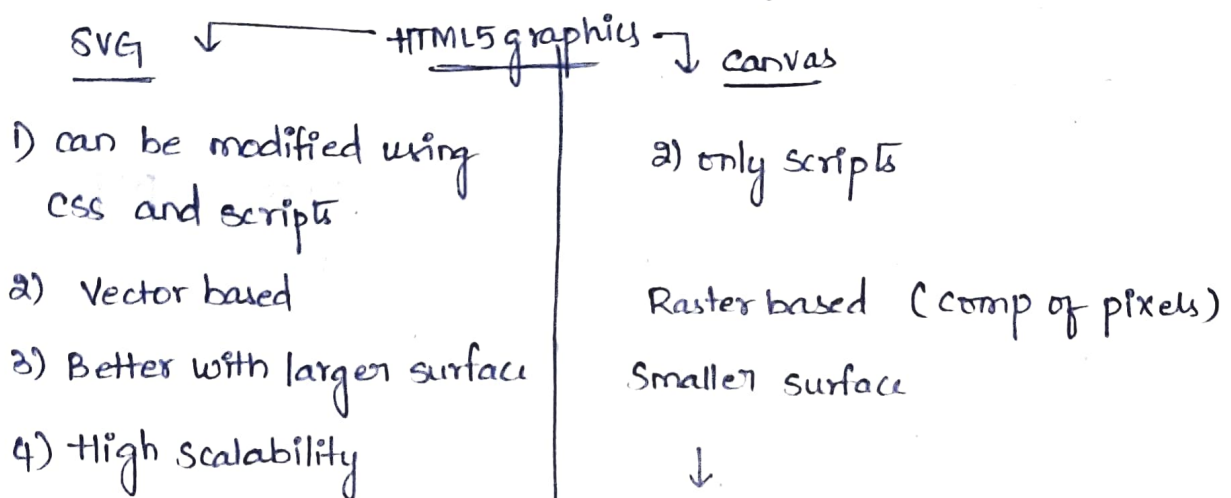
- 1) JS can run in background.
 - 2) Multimedia support.
 - 3) allows users to draw various shapes.
 - 4) New Semantics.
 - 5) capability to store offline data using SQL db.
- 1) Better structuring, new tags, cleaner
 - 2) Interactive elements. Video instead of flash plugin.

Inline: <a> <input> <label> etc.

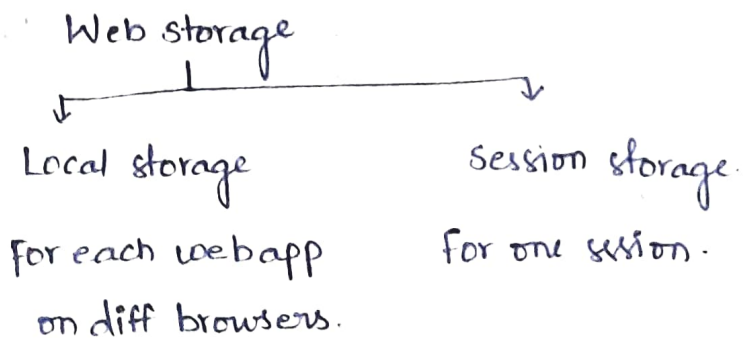
Block: <div> <p> <header> <footer> <form> <table>

Semantic elements: describes particular meaning to the browser and developer. Like <form> <table> <article> etc.

- progress tag should be used when we want to show the whole progress of a task. Flex meter tag can be used.



- Audio types allowed are, mp3, wav, ogg.



Web storage helps in storing some of the static data in the local storage of browser so that we do not need to fetch the data every time from server which decreases load time and achieves smooth user experience.

output tag is used for representing the result of a calculation.

According to HTML5 specification, each `<header>` element must have one atleast `ch1>` tag. In previous HTML version, there was only `ch1>` for entire webpage

Input types : date week month time datetime datetime-local color email number search tel range url.

Media tags: audio video source embed track.
(audio + subtitles)
Video + ...)

MathML - mathematical mark up language.

used for displaying mathematical expressions