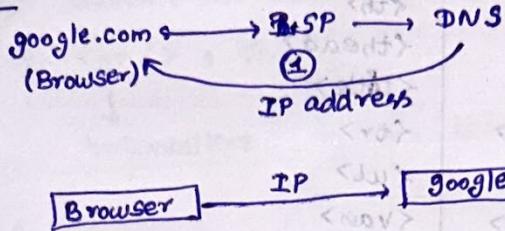


Web development

Internet:

- * Global network of networks (uses routing for communicating)
- * why route: not possible to connect everything (IP, Internet protocol)
- * Data: 0/1 (on/off)
- * server (24/7 up), client: communicates with "Server!"

* DNS:

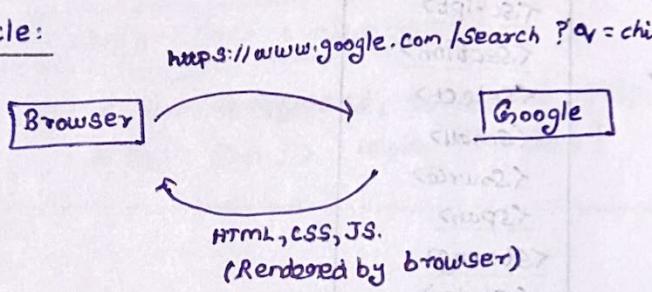


nslookup.90

web:

- * www: world wide web (Info system): share doc, resources identified by URLs
- * HTTP protocol (foundation of www): set of rules
- * others: TCP/IP protocols

Request-Response Cycle:



HTML, CSS, JS:

- * HTML: Nouns (Dino)
- * CSS: Adjective (purple)
- * JS: Verb (Danced): live notification, pop up, search, auto complete, ...

HTML:

- * Hyper Text Markup Language (link other pages /docs), markup: bold, italics, quote
- * Some elements: opening & closing tag (some don't)

Prohibited element: <password>

<input type="password">

Forbidden element: <script> — <script>

<script>

<style> — <style>

<script>

<script>

<script>

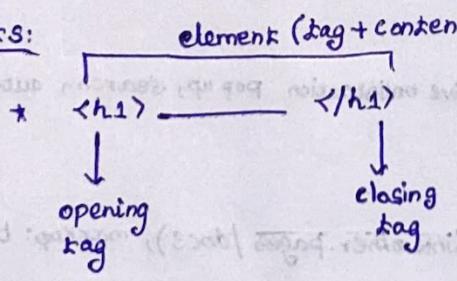
<script>

HTML tags:

2

| | | | |
|---------------------------|---------------------------------|-------------------------|--------------------------|
| <u><a></u> | <u><dt></u> | <u><meta></u> | <u><sep></u> |
| <u><abbr></u> | <u></u> | <u><meter></u> | <u><svg></u> |
| <u><address></u> | <u><embed></u> | <u><nav></u> | <u><table></u> |
| <u><area></u> | <u><fieldset></u> | <u><noscript></u> | <u><tbody></u> |
| <u><article></u> | <u><figcaption></u> | <u><object></u> | <u><td></u> |
| <u><aside></u> | <u><figure></u> | <u></u> | <u><template></u> |
| <u><audio></u> | <u><footer></u> | <u><optgroup></u> | <u><textareas></u> |
| <u></u> | <u><form></u> | <u><option></u> | <u><tfoot></u> |
| <u><base></u> | <u><head></u> | <u><output></u> | <u><th></u> |
| <u><bdi></u> | <u><header></u> | <u><p></u> | <u><thead></u> |
| <u><bdo></u> | <u><hgroup></u> | <u><pre></u> | <u><title></u> |
| <u><blockquote></u> | <u><h1> to <hb></u> | <u><picture></u> | <u><tr></u> |
| <u><body></u> | <u><hr></u> | <u><progress></u> | <u></u> |
| <u>
</u> | <u><html></u> | <u><q></u> | <u><var></u> |
| <u><button></u> | <u><i></u> | <u><rb></u> | <u><video></u> |
| <u><canvas></u> | <u><iframe></u> | <u><rp></u> | <u><wbr></u> |
| <u><caption></u> | <u></u> | <u><rt></u> | |
| <u><cite></u> | <u><input></u> | <u><ruby></u> | |
| <u><code></u> | <u><ins></u> | <u><rt></u> | |
| <u><col></u> | <u><kbd></u> | <u><samp></u> | |
| <u><colgroup></u> | <u><keygen></u> | <u><script></u> | |
| <u><data></u> | <u><label></u> | <u><section></u> | |
| <u><details></u> | <u><legend></u> | <u><select></u> | |
| <u><dd></u> | <u></u> | <u><small></u> | |
| <u></u> | <u><link></u> | <u><source></u> | |
| <u><details></u> | <u><main></u> | <u></u> | |
| <u><dfn></u> | <u><map></u> | <u></u> | |
| <u><dialog></u> | <u><mark></u> | <u><style></u> | |
| <u><div></u> | <u><menu></u> | <u><sub></u> | |
| <u><dl></u> | <u><menuitem></u> | <u><summary></u> | |

HTML Elements:



Boilerplate:

```

<!DOCTYPE html>
<html> → Top level element!
  <head>
    <title> my web page </title>
  </head>
  <body>
    </body>
  </html>
  
```

HTML 5 (using)

] Browser work enforce!

- * <html> : top most (every other: descendant (1 head, 1 body followed))
- * <head> : meta info title
- * <body> : All contents of the document!

Shortcut: ! : vscode

void elements (self closing):

- * No content : forbidden (no need for closing tag)
- * eg:
,

→ (right)
/ : Forward
\ : Backslash
(left)

↓
horizontal line

→ Break element

Do's and Don't's:

<p> — </p>
<p> — </p>

DO

<p> —

—

Not do

**
,
 : also valid,**

* HTML (preferred, as it will skip /)

* But
 more readable!

List elements:

Unordered list:

 —
 —

Ordered list (numbered):

 —
 —

can nest: Just indent for readability

Anchor tag (hyperlink)

 text

↓
attribute (additional info: where to link/ redirect)

↳ http, mail, mobile, file, sections of page, portions of text

Global attributes:

* draggable : true

→ **works with all tags!**

<Image> tag:

`` : self closing tag!

- ↳ src: URL
- ↳ alt: alternative text (screenreaders, when img breaks)

Comments:

* `<!-- -->`

Command + / (VSCode or other browsers)

HTML5:

* Latest evolution of HTML Standard (new elements, attributes, behaviours)
 ↳ multimedia, 2d/3d graphics, performance, browser storage,...

* Living Standard

* Role of browsers: do the work, Implement HTML (based on rules)

`html.spec.whatwg.org`

* `<!DOCTYPE html>` : HTML5

HTML 4.01 Strict

HTML 4.01 Transitional

HTML 1.0 Strict

* Just informs browser what markup it should expect!

* Not disables HTML5 API(s).

File paths:

* Absolute (From root of the Computer): `c:/Project/Images/cat.png`

* Relative (Root folder (say current) `Images/cat.png`)

↳ Inside Project: `index.html`

↳ Shorter, valid (even moved: hosted)

1) `.. / index.html` : Go upper level (parent folder)

2) `./ dog.png` : Current directory

Webpages:

* single/multiple pages (`index.html, about.html, contact.html`)

* `<html lang="en">` — `</html>` : Language of the content (Screen readers)

* `<meta charset="UTF-8">` : For rendering! (UTF-8: most common)

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

↳ Deprecated (IE)

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

↳ screen size (Render)!

Block (vs) Inline Elements:

* div: content division / division element!

* generic element to hold/group elements (block level)

* Span :

* generic inline element!

Other elements:

* hr (`<hr>`): thematic break b/w paragraph level elements

* <sup> : a^2

b^{sup} & c^{sup}

* <sub> : a_2

a_{sub} & c_{sub}

HTML entities

* special codes / sequences (preserved in HTML: need alternatives)

* starts with &, ends with ;

`♠`

dev.w3.org/html-author/entities

eg:

`<` → <

`>` → >

` `

Semantic markup

* relating to meaning (purpose / role of an element)

* instead of divs: main, header, section, nav, footer

main content

navigations
(links)

`Accessible : screenreaders`

similar to div

`<section>`
`<article>`
`<nav>`
`<main>`
`<header>`
`<footer>`
`<aside>`
`<summary>`
`<details>`

with meaning!

- * main: dominant & unrepeated content
Shouldn't include (sidebars, nav links, copyright info, logo, search forms)
main (not like h1, body): Doesn't affect Dom!
 - * nav: navigation links (same page, diff resources)
 - * section: generic (standalone section of a page)
 - * article: self contained composition (say weather info) : Independent to the website!
 - * aside: indirectly related (sidebar = unrelated)
 - * header: intro content / navigational content at the top (combined)
 - * footer: links, navs, about, copy right

more than 1 header/footer (article : header, footer)

- * **Time**: time/date/time content!
 - * **Figure**: (can contain img, figcaption (description))
 - * **abbr**:
 - * **data**:

Screenreaders: Identify where is main, nav content!

Table :

- * Earlier tables were used to layout websites (not now)

* <table> <caption> Sample Table </caption>
 <thead>

```
<th>  
  <th colspan="2"> The table header </th>  
</tr>  
</thead>
```

<tbody>

<tr>

<td> The table body </td>

<td> with 2 columns </td>

<Hr> (cont'd)

</tbody>

{/tables}

Ad: single cell

th: header cell

try : you

```
<tfoot>
  <tr>
    <th scope="row"> Totals </th>
    <td> 24,000 </td>
  </tr>
</tfoot>
```

| | |
|--------|--------|
| | |
| totals | 21,000 |

Scope; values; actions

| Col | Row |
|------|-------|
| Item | Total |
| 1 | 100 |
| 2 | 102 |
| 3 | |

Colspan, rowspan:

* Colspan = '2' : Take 2 columns

* Rowspan = '3' : Take 3 rows

form:

- * Shell/container: hold collection of q/p's, checkboxes, buttons
- * No visual meaning / Impact.

Form: Document section (Interactive Controls): submit info

Action: where to send the data

Method: HTTP method.

(input and form) labeling + input

Input:

- * 20+ Input elements: types: checkbox, colorpickers, date picker, password, email.
- * type: text, password, color, number, button, radio, checkbox, select
- * placeholder: Before user entering/inputting!

<label>:

- * make accessible & easy to use!
- * associate q/p & text!
- * experience: label also clickable!

Id, for: unique!

```

<div>
  <label for='cheese'> _____ </label>
  <input type='checkbox' name='cheese'
         id='cheese'>
</div>
  
```

Submit button:

* default (inside form): submits

* type = 'Submit': submits

* type = 'button': don't submit!

<input type = 'Submit' value = 'Submit form' />

<button type = 'button' value = 'Submit form: Don't' />

Name attribute:

* sent as key:value (In URL/body)

* name → Key

* user q/p → value!

checkbox

```

<div>
  <input type='checkbox' id='scales' name='scales'
         checked />
  <label for='scales'> Scales </label>
</div>
  
```

Scales
 Horns

Radio

group: Only select one! (connected)

↳ name should be same!

Value will be sent!
(attribute)

Range & Text area:

* Range: Say Dial (volume, Age, ...)

* Text area: multiline text input!

Builtin html Validations: (no JS)

* required (must!)

* min, max: numeric

* minlength, maxlength

* regex/pattern!

* url, email (inbuilt regex)

HTML: Detailed (Not case sensitive)

* <!DOCTYPE html> : using HTML 5 document

* <html> : root element

* <head> : meta info (<title>, ...)

* <body> : visible elements (h1, p)

* h1 to h6 : headings : Search engines (uses: Index)

* Paragraphs

* a, links

* Images : <image>

Recommended: lowercase (elements/tags)

<HTML> : Lowercase (as strict doctype)

use double quotes (standard)

Attributes:

* additional info about elements (specified within the start tag)

* Key = 'value'

 __

<image src='__' width='500' height='600' alt='Alternative text'>

<p style='color: red'>

↳ Don't do this (That's why CSS)

<html lang='en'>

<html lang='en-US'>

<p title='My Para'> __ </p>

↳ tooltip (when layover: show text!)

Browsers removes extra whitespaces

<p>

</p>

} Preformatted element!

- * `style = 'property: value'`
- * `background-color: tomato;`
- * `color: blue;`
- * `font-size: 100%;`
- * `text-align: center`

- * `borders: 2px solid Tomato
(thick) (solid, dotted)` ↳ color
- * `rgb(r, g, b)`
- * `#ff6347`
- * `hsl, hsla, rgba`

Formatting:

- * ` : bold`
- * ``: Important text : displayed in bold : **Semantic significance**
- * `<i>` : Italic
- * ``: emphasize → Italic (screen reader: stress)!
- * `<mark>`: marked text → mark/highlighted
- * `<small>`
- * ``: deleted text : `` : cross (line)
- * `<ins>`: Inserted text : underline
- * `<sub>`, `<sup>`

Quotation / citation:

- * `<blockquote>` : Indent! (Indent IPNS)
- * `<q>` : Short quotes →
- * `<abbr>` : Show abbr (fullform: when cursor lay over)
- * `<address>`: email, URL, number, Social media handle
- ↳ Italic, line break before & after!
- * `<cite>` : title of creative work (not author name!) : Italic
- * `<bdo>` : bi-directional override (override direction of display)
`<bdo dir="rtl"> — </bdo>`

Comments:

`!--`

→

Colors:

- * tomato, orange, dodgerblue, medium SeaGreen, gray, slateblue, violet, lightgray

- * 140+ standard color names

- * `rgb (255 x 255 x 255)`

(Black → white)

- * `rgba (alpha: 0 to 1 (fully transparent to opaque))`

- * `#ff6347` (hex: `adigit: r, g, b`)

- * `hsl` (hue, saturation, lightness)

↳ degree of color (0 to 360°) wheel

Saturation: Intensity 100% pure, (50% red, 50% gray)

Lightness: 0% (black), 100% (white)

- * `HSLA` : `hsl + Alpha`

Other styles:

- * font-family
- * font-size
- * color
- * border: 2px solid powderblue
- * padding:
- * margin:

Links:

- * `a href = "url"` → URL
- * `mailto:`

- * `target :`
 - self: default (Same window/tab)
 - blank: new window/tab
 - parent: In parent frame
 - top: Full body of the window (open the doc in)

where to open the linked doc

- * can use image, button, .. as links

`title = " " ; Show as tooltip`

* `a:link { color: green; }`

3 * `a:visited { color: pink; }`

3 * `a:hover { color: red; }`

3 * `a:active { color: blue; }`

Bookmarks:

- * add link to the part of a page. (use ID)

` Chapter title `

- * other page

` Doc 1 title `

Images:

- * ``: embed image (no closing tag)

- * `src: URL, alt: alternative text`

- * `size: width, height (or) style = "width: 100px; height: 100px;"`

- * Image as link,

formats

→ `.apng`: animated portable network graphics

→ `GIF`: Graphics Interchange format

→ `.ico, .cur`: Microsoft icon

→ `.JPG, .JPEG, .JIFF, .JPEG, .JP2`: Joint photographic expert group image

→ `.png`: Portable Network graphics

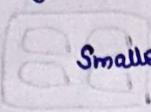
→ `.svg`: Scalable Vector graphics

Image map:

- * Clickable areas on an image
- * ``
- * `<map name='workmap'`
`<area shape='rect' coords='34,44,270,350' alt='Computer'`
`href='Computer.htm'>`
- `</map>`

rect, circle, poly, default

Background Images:

- * `<p style='background-image: url('img-girl.jpg');'>`
- 
- Smaller images repeats itself!
 \hookrightarrow `background-repeat: no-repeat; (stop!)`

- * Cover entire region:

`background-attachment: fixed;` → No Stretching (keep original proportions)
`background-size: cover;`

- * Stretch: (fit entire element)

`background-size: 100% 100%;`

Picture:

- * multiple images: choose best fit!

```
<picture>
  <source srcset="img-avatar.png" type="image/png" alt="Avatar">
  <source srcset="img-girl.jpg" type="image/jpeg" alt="Beatles">
  
</picture>
```

- 1) Small screen (attribute: use this image)

\hookrightarrow `media = '(min-width: 600px)'`
 \hookrightarrow `media = '(min-width: 465px)'`

- 2) Format support

\hookrightarrow some browsers can't recognize (use 1st format recognises)

Favicon:

- * page title icon: (simple image, high contrast): type: image/x-icon [ico format]

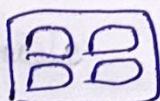
`<link rel="icon" type="image/x-icon" href="/images/favicon.ico">`

Title:

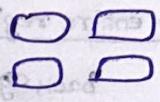
- * content & meaning of page (Important for SEO): uses this favourite/bookmark, SEO

Tables:★ `td: cells`★ `tr: rows`★ `th: headers = bold, centered`borders:★ `borders: 1px solid black;`★ avoid double border → `borders-collapse: collapse;`BgColor:★ `th, td { background-color: #96D4D4; }`Round table borders:borders: 2px solid black;] `table, th, td`

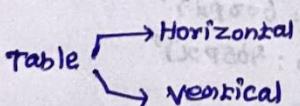
border-radius: 10px;

`th, td`

(leave table)

borders style:

border-style: dotted, dashed, solid, double, groove, ridge, inset, outset, none, hidden

border-color:Table Size:★ `<table style="width: 100%;"` → Takes full screen width!Column width:`<th style="width: 70%;"` → `</th>`Row height:`<tr style="height: 200px;"`align headers:`th { text-align: left; }`multiple cols (headers):★ `<th colspan="2" style="text-align: center;">Name` `</th>`caption: (heading of a table)★ `<table>`
`<caption> ----- </caption>``</table>`

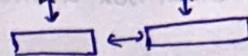
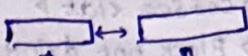
Padding (vs) Spacing:

* padding: space b/w edge and cell content (default: 0)

* spacing: space b/w 2 cells (2px: default)

[JUL] : 0px padding

[JUL] : 10px padding

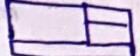
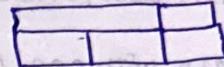


Spacing: 10px

Rowspan (vs) Colspan:

* multiple cell: Colspan

* multiple row: rowspan



Colspan

Rowspan

[rowspan] [rowspan]

Table Styling:

tr: nth-child(even) { background-color: powderblue; }

td: nth-child(odd), th: nth-child(odd) { " " ; }

Combine horizontal, vertical zebra stripes:

* tr: nth-child(even) { — }

* td: nth-child(even), th: nth-child(even) { — }



border:

* border-bottom, border-top, border-right, border-left

Hover:

* tr:hover { background-color: — ; }

HTML: Table colgroup:

* purpose: bg color for first 2 columns

* <table>

<colgroup>

<col span='2' style='background-color: tomato;'>

</colgroup>

<t>

<th>--

:

</table>

Say 1,2: no color, 3 to 5: pink

<col span='2'>

<col span='3' style='— '>

Hide Columns:

Visibility: collapse;

- * dl > dt, dd → <dl>
 - <dt> Coffee </dt>
 - <dd> Black hot drink </dd>
 - </dl>

* unordered:

markers: desc, circle, square, none (IPST-style-type)

* Horizontal bar with CSS (menubar)

* float / grid / table !

* ordered list:

- * <ol type='1'> → 1, A, a, I, ?

* Count : Stark = '50'

*dl:

- * dl : description list

* dt : term(name)

* dd : description!

Block vs Inline elements:

- * Block: new line (margin: before, after: added) : full width

| | | | | |
|-------------------------------|---------------------------------|---------------------------------|------------------------------|------------------------------|
| <code><address></code> | <code><article></code> | <code><noscript></code> | <code><dd></code> | <code><table></code> |
| <code><p></code> | <code><figcaption></code> | <code><blockquote></code> | <code><h1-h6></code> | <code><dt></code> |
| <code><div></code> | <code><nav></code> | <code><footer></code> | <code><pre></code> | <code></code> |
| <code><fieldset></code> | <code><video></code> | <code></code> | <code><hr></code> | <code><tfoot></code> |
| <code><main></code> | <code><aside></code> | <code><canvas></code> | <code><header></code> | <code><caption></code> |
| <code></code> | <code><figure></code> | <code><form></code> | <code><section></code> | <code><caption></code> |

- * Inline: ~~Starks (No!)~~ On a new line
only necessary space!

| | | | |
|-----------------------------|-------------------------------|-----------------------------|-----------------------------|
| <code><ar></code> | <code><acronym></code> | <code><bdo></code> | <code>
</code> |
| <code><code></code> | <code></code> | <code></code> | <code><Kbd></code> |
| <code><object></code> | <code><av></code> | <code><script></code> | <code><small></code> |
| <code><sub></code> | <code><textarea></code> | <code><tt></code> | <code><button></code> |
| <code><abbr></code> | <code></code> | <code><big></code> | <code><label></code> |
| <code><dfn></code> | <code><i></code> | <code><input></code> | <code></code> |
| <code><output></code> | <code><samp></code> | <code><select></code> | <code><cite></code> |
| <code><sup></code> | <code><time></code> | <code><var></code> | <code><map></code> |
| | | | <code></code> |

div:

- * Block element (full width + top, bottom linebreaks)
 - ↳ container (centre), align them side by side

* center align:

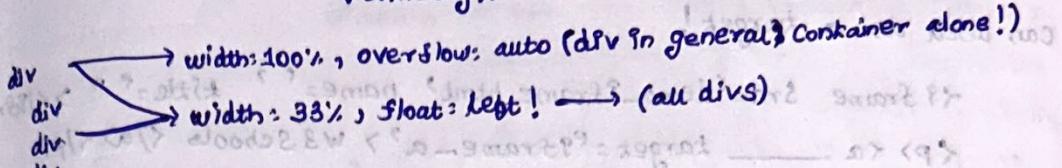
```
div {
```

```
  width: 300px  
  margin: auto
```

3

* Align Side by Side:

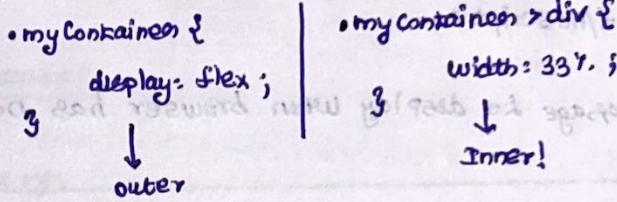
- * float: (purpose: position & format content horizontally rather than vertically).

* inline-block:

```
display: inline-block (div in general)
```

* Flex:

- * Flexible layout design without float/positioning.

* Grid:

- * grid based layout system (rows & columns)
- * like flex but define more than 1 row, position each row!

```
grid-container {
```

```
  display: grid;
```

```
  grid-template-columns: 33% 33% 33%;
```

class:

- * used by JS: manipulate/access elements with specific class

```
<div class='city'> → CSS → .class { }
```

* multiple classes:

```
<div class='city town'> .. </div>
```

```
getElementsByClassName()
```

Id:

- * ID: Unique Id of an element. (getElementById('ID'))

```
<div id='headDiv'>
```

;

```
</div>
```

```
#headDiv {
```

3 redon

- * **Class**: used by multiple elements
 * **ID**: unique for an element!

ID: use in link(anchor tag), use in JS

16

Iframes:

- * **Inline frame**: embed another doc within current HTML doc

→ <iframe src='url' title='description'></iframe>

↳ For screenreaders

→ height, width, style, border (style),

→ can used as a target!

<frame src='demo_iframe.html' name=' ' title=' '></frame>

<p>W3Schools</p>

↳ Show that link op in pframe!

when clicked!

<Script>:

- * JS in html

- * <noscript>

</noscript>

↳ message to display when browser has no JS support!

<head>:

- * **title**

- * **style**

- * **link** (relation b/w current & another doc): external style sheets

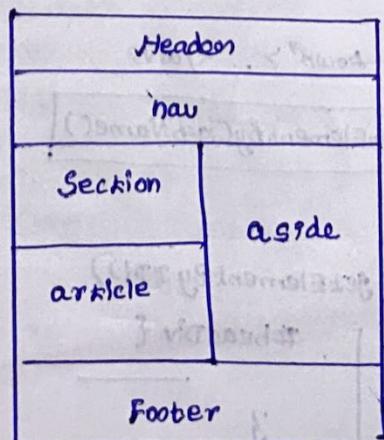
<link rel='stylesheet' href='myStyle.css'>

- * **meta**: Charset, Key words for SEO

description, author, refresh, viewport (visible region)

- * **base**: base URL for all relative paths (must be 1 base only)

<base href='https://---' target='_blank'>



Layouts: tech:

- 1) CSS framework (bootstrap)
- 2) Using float
- 3) Flexbox
- 4) Grid

Responsive:

- * Set viewpoint `<meta name='viewport' content='width=device-width, initial-scale=1.0'>`
- * Responsive Images: `width: 100%; max-width: 100%;` (will scale down but won't scale up!)
- * Picture: Show different Images based on browser width
- * Text size: `vw → view point width <h1 style='font-size: 10vw'>`
- * media queries:
 - * different styles for different browser sizes


```
@media screen and (max-width: 800px) {
            .left, .main, .right {
              width: 100%; }
```
- * responsive web page: must look good for both small & large screens.

Frameworks

- * W3.CSS (Independent of Jquery)

- * Bootstrap!

Computer code elements:

- * `<code> z = x + y; </code>`
- * `<kbd> : Keyboard I/p (ctrl+s) : monospace font`
- * `<samp> : Sample Computer program o/p : monospace font`
- * `<pre> : preserve whitespaces & line breaks`
- * `<var> : variables : italic : math expression`

Semantics:

- * meaning: browser, dev, screen readers

- * `<Section>`: Thematic group of content (chapters, intro, items, contact info)
Not so meaningful, still better than `div`

- * `<article>`: independent, self-contained content (rest of the website)
posts, blog, comment, cards, news articles

Sometimes: `sections > article, article > section`

No rule: Intuitive

- * `header`: Intro content / set of nav links (`headers: h1-h6, logo/icon, author info`)
- * `footer`: author info, copyright, contact, sitemap, back to top links, related docs
- * `nav`: not all links, only major links (screenreaders: uses where to omit)
can't read every links!
- * `aside`: sidebar (indirectly/not related)

* <figure>

 </figure>

why semantic elements:

* Data: Shared across app, enterprises, communities

HTML guide

- 1) Always use <!DOCTYPE html>
- 2) lowercase for element names
- 3) close all elements
- 4) lowercase attribute names
- 5) Always use quotes: " " : for attribute values
- 6) Images: always specify alt, width, height (Reduce flickering: Browser removes spans)
- 7) Don't use spaces around '=' signs → easy to read
- 8) Avoid long code lines
 - add blank lines : separate large/logical code blocks
 - add 2 spaces for indentation not tabs
- 9) Don't add blanklines, spaces, Indentation (without reason)
- 10) Never skip title (SEO)
- 11) Don't omit html, head, body (older browsers: crash DOM, XML software)
- 12) optional: <meta ... /> or <meta ... >
- 13) Add lang attribute, charset (meta)
- 14) Set viewports
- 15) Comments:
 - single line: <!-- -->
 - multiline:
<!--
-->

For XML, XHTML Software to access webpage: use /

StyleSheets:

* link rel='stylesheet' href='styles.css' → no need for type attribute

* short CSS rules: P { font-family: verdana; }

* Long CSS rules: multiline!

body { } → { same line, space before { }

; ;

2 spaces: indentation

Semicolon

use quotes! → when spaces!

3 closing } : no spaces!

JS:

- * <script src='myscript.js'> : no need for type
- * don't use unidy HTML codes

↳ Demo ↳ not same!
demo

* USE lowercase for filenames. (webservers: Case Sensitive, may be not)

* .html, .css, .js : extensions

* .htm (allowed, same as .html)

* default name: index.html / htm , default.htm / htm

characters:

* Reserved one ↳ <, > (<, >)

↳ (non breaking space)

↳ ‑ (non breaking hyphen)

↳ &, ¢, £, € : Case Sensitive

diacritical marks (glyph added to a symbol) ↳ à, á, ã

* math symbols, emoji, great symbols

UTF-8 : All Symbols (almost!)

other: ASCII, ANSI, ISO-8859-1, UTF-8

URL:

http://www.google.com:1773/path/filename

↓ ↓ ↓ ↓

Scheme prefix domain Port
(http, https,
mailto, file,
ftp)

<password>=asdasd
<username>=bit_galibat

URL encoded & transmitted

HTML VS XHTML:

* Extensible HTML (stricter, more XML based version. for XML apps)

* well formed, stricter error handling

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"> → DOCTYPE must!

<http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd>

xhtml xmlns="http://www.w3.org/1999/xhtml"> → html, xmlns: must!

* html, head, body, title: must

* proper nesting, closing tags, lowercase elements, attributes: lowercase

* quoted values (attribute), attribute minimization: forbidden!

 : Expected!

 : wrong!

wrong: <input type='checkbox' checked>

correct: <input checked='checked'>

Forms:

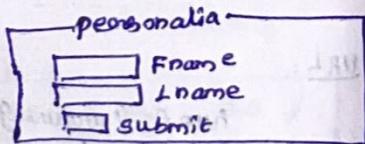
- * Container for input elements
- * <input type: text, radio, checkbox, Submit, button
- * <label for="id"> — </label>
- * Submit: action = '... .php'
- * name (input) attribute: say search query (?name=value)

Attributes:

- * action: Send data to this handler
- * target: Where to display the result (-blank, -top, -parent, -self, framename)
- * method: http method (get, post)
 - get (URL append, not for sensitive data, length < 2048 char, bookmark (complete URL), non secured data: query strings)
 - post (no size limit, body, can't bookmark)!
- * autocomplete='on' → display previously entered values!
- * <form novalidate> → can't validate when submitted

Elements:

- * input, label (screenreaders, related with input: click label, affects input element)
- * select > option → multiple (attribute): allow selecting multiple!
- * textarea (rows, cols or style='width: 200px; height: 100px')
- * button (onclick = alert('Hello world'))
- * fieldset: group related elements in a form
 - ↳ <legend> Personalia </legend>
- * datalist: predefined options for input
 - (dropdown)



```

<input list='browsers'>
<datalist id='browsers'>
  <option value='Edge'>
  <option value='Chrome'>
</datalist>
  
```

- * <output> : represents result of a calculation

```

<form action='/action-page.php' oninput='x.value = pageInt(a.value) + pageInt(b.value)'>
  
```

! assume: a → $\frac{a}{b}$
 $\frac{a}{b} \rightarrow$

$\frac{a}{b} \rightarrow$

<output name='x' for='a b'></output>

</form>

! output: $\frac{a}{b}$

Input types:

| | | | | | |
|------------|-------------------|------------|----------|----------|--------|
| * button, | * date-time-local | * image | * radio | * submit | * url |
| * checkbox | * email | * month | * range | * tel | * week |
| * color | * file | * number | * reset | * text | |
| * date | * hidden | * password | * search | * time | |

Attributes:

- * value (To send while submitting)
- * readonly (P/P is readonly)
- * disabled (unusable & unclickable)
- * size='50': visible width (characters) : default = 20
- * maxlength, minlength: String char length!
- * min, max: numbers
- * multiple (allow multiple select/files,...)
- * pattern (matches P/P with the given regex)
- * required (P/P must)
- * Step (Legal number intervals) : 3,6,....
- * autofocus: when page loads (where to focus) : In an element!
- * height, width (Image element)
- * list: refers to datalist
- * autocomplete

Formaction attribute (submit P/P):

- Input type='submit'**
- * overrides form.action attribute
 - * formenctype: encoded how while submitting (only for post)
 - * formmethod
 - * formtarget
 - * formvalidate

Graphics:

- * <canvas>: draw graphics (drawLine using JS) : script tag!
 - * <svg>: rect, ole, polygon, text,...
- (SVG: language for 2D graphics) : XML based (has DOM)
each shape: object!

Svg:

- * Resolution independent
- * Support event handling (JS)
- * Good text rendering
- * Slow render (Complex things)
- * Not suited: game app

Canvas

- * Resolution dep
- * No support
- * poor text & rendering
- * .png, .jpg (end result: saves)
- * well suited: graphic intensive game

Multimedia:

* common video: MPEG, AVI, WMV, Quicktime, rm, ram, SWF, flv, ogg
mp3, webm

* audio: mp3, realaudio, wma, aac, wav, mp3, ogg, mp4

* video

<video width='300' height='250' controls>

Source src='__' type='video/mp4'

Source src='__' type='video/ogg' → use 1st recognized format

</video>

* autoplay, muted (start playing auto but muted)

* audio controls

<source src='__' type='audio/mpeg' /> → autoplay, muted

</audio>

* plugins:

* Java applets, flash, display map, scan for virus, verify bank id.

* <object> supported by all browsers

Embed object with html element

<object data='Snippet.html'></object>

Embed src='Snippet.html' → Not supp by all!

* iframe > youtube

HTML APIs:

* geolocation (lat, long)

* drag & drop

* web storage (key/value pairs: more intuitive than Cookies)

* web workers (run BG → user can click, ... WORKER RUNS IN BG)

* server-sent events (auto update from a server)

* canvas

* 3rd party: youtube, twitter, ...

geolocation: (most) based on IP (asking user for permission, yes/no)

* supported browser + user allow

* error handle (error.PERMISSION_DENIED)

draggable:

* draggable='true' → <p>,

* specify data type eg draggable data=

* default: drop in all element (restrict)

```

1) function dragStartHandler(ev) {
    ev.dataTransfer.setData('text', ev.target.id);
}

function dragOverHandler(ev) {
    ev.preventDefault();
}

function dropHandler(ev) {
    ev.preventDefault();
    const data = ev.dataTransfer.getData('text');
    const element = document.getElementById(data);
    ev.target.appendChild(element);
}

```

Web Storage API:

- * localStorage.getItem(), setItem(), removeItem()

eg: clickCount: sessionStorage, clickCount

Web Workers:

- * HTML: unresponsive until script is finished.
- * Web Worker: external JS (run in Bg): independent of other scripts
Doesn't affect the performance of page!
- * Heavy code which can't run on the main thread (prevent unresponsiveness)
(CPU intensive tasks)

```

<script>
const el = document.getElementById('result');
if(typeof Worker != 'undefined') {
    el.innerHTML = 'Supported';
}

```

</script>

Create .js worker file:

```

if (typeof(w) == 'undefined') {
    w = new Worker('demo-ss.js');
}

```

demo-ss.js

```

var i=0;
function timedCount() {
    i += 1;
    postMessage(i);
    setTimeout('timedCount()', 500);
}
timedCount();

```

post to worker! (HTML page)

Full code:

```

<script>
let w;
function startWorker() {
    const el = document.getElementById('result');
    if(typeof(Worker) == 'undefined') {
        el.innerHTML = 'No Sup!';
    } else {
        w = new Worker('demo-ss.js');
    }
    w.onmessage = function(event) {
        el.innerHTML = event.data;
    };
}

```

```

else {
    el.innerHTML = 'No Sup!';
}
function stopWorker() {
    w.terminate();
    w = undefined;
}

```

Server-sent Event API:

84

- * automatically get message/update from a server
- * Facebook/twitter/stocks/feeds/news...

<Script>

```

const x = document.getElementById('result');
if (typeof(EventSource) != 'undefined') {
    // check supported in browser
    var source = new EventSource('demo.php');
    source.onmessage = function(event) {
        x.innerHTML += event.data + '  
';
    };
} else {
    x.innerHTML = 'Not Supported';
}

```

</Script>

demo.php:

```

<?php
header('Content-Type: text/event-stream');
header('Cache-Control: no-cache');
$time = date('r');
echo "data: the server time is: {$time}\n\n";
flush();

```

>