

HTML

Subham mohanty

HTML - Hyper text

↓
Text that have links to another text/data

Markup

↓
Annotation to contains as header footer etc.

Language

↓
HTML is having certain syntax.

It's not case sensitive.

HTML

↓
Structure of web page

CSS

↓
style

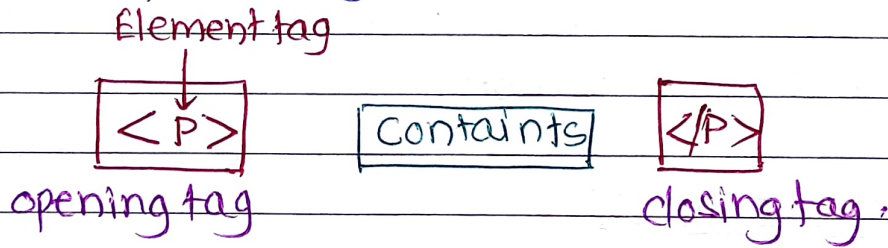
JavaScript

↓
Behaviour

HTML - 1) Annotations to containt
2) structure to page.

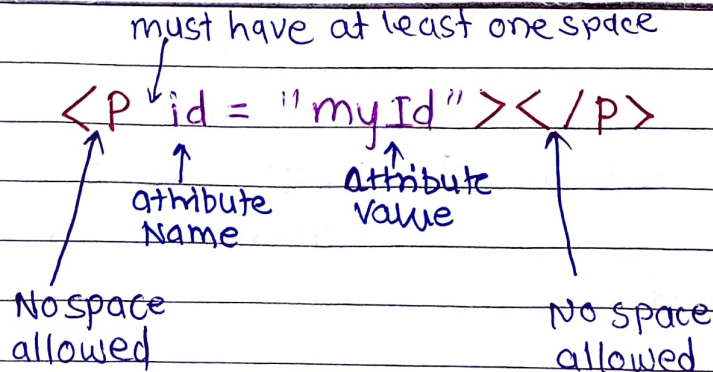
Anatomy of Tags

1)



Every opening tag doesn't have closing tag
e.g.
 , <hr>

2)



space is allowed anywhere else and ignored by browser.

attribut value is not compulsory to be in quotes but best practise to.

3)

✓ `<P> </P>`✗ `<P/>`

In HTML5 - for empty tag, only as a placeholder

Declaration

`<!doctype html>`
 ↳
 NoSpace

Declaration is used to inform browser that whether Page following HTML Standards or not.

Rendering

Rendering in HTML is Top to bottom @ everything runs from top to bottom.

HTML content models

Block level Elements

- 1) Render to begin on new line (By default)
- 2) May contain inline element
- 3) It also pushes next elements also to next line to be alone on same line.

Inline Elements

- 1) Render on the same line (by default)
- 2) Cannot have Block level elements in it.

HTML Elements

<!DOCTYPE html> - All HTML documents starts with document type declaration which specify that html doc. follows standard format which helps browser to display webpage correctly.

<html> - everything/every element present inside it

Header

<head> - It is the element specified between **<html>** and **<body>** tags

- It is the container for following elements.

i) **<title>** - It provide title to webpage to be displayed in browser's tab, search engine results, when page added to favourite etc.

- Important in SEO point of view.

ii) **<style>** - It is used for styling with CSS
CSS commands are written inside this element.

iii) **<script>** - Javascript commands are written in **<script>** element and **<noscript>** for specifying what should displayed when javascript not supported by browser.

iv) **<base>** - It is used to define base URL or targets for all relative URLs in page. It should have either href or target or both

`<base href=" " target=" " >`

targets are - blank

<

✓ `<link>` - It has two attributes -

`rel` - It defines relationship between document and external resource.

It can be "stylesheet" for css file

`href` - link of file.

eg

`<link rel="stylesheet" href="URL" >`

`<meta>` - It is used to specify different data. Its data not displayed on page but used by browser, search engine and other web services.

1) character sets

a) `<meta charset = "UTF-8" >`

2) keyword for SEO

b) `<meta name = "keywords" content = "HTML JS CSS" >`

3) Description of content

c) `<meta name = "description" content = "html tutorial" >`

4) author name

d) `<meta name = "author" content = "shubham" >`

5) viewport for different device support

e) `<meta name = "viewport" content = "width=device-width initial-scale=1" >`

6) Page refresh after 30 sec.

f) `<meta http-equiv = "refresh" content = "30" >`

width = device-width - default screen size.
 initial-scale = 1.0 - Default zoom level

Body

`<body>` - It defines the documents body inside `<html>` tag. Everything that is visible on webpage is specified in this body

Elements that falls inside Body are -

Headers

- | | | |
|---|--|---|
| <ul style="list-style-type: none"> i) <code><h1></code> ii) <code><h2></code> iii) <code><h3></code> iv) <code><h4></code> v) <code><h5></code> vi) <code><h6></code> | <p style="text-align: center;">↓
decrease importance
and size by default</p> | <p>- head-tags have Titles or subtitles of contents, proper use of heads from <u>h1 to h6</u> is required as per SEO.</p> |
|---|--|---|

Paragraphs

i) `<P>` - It has paragraph content anything separated by spaces are ignored by browser and appears as single line as possible.

ii) `<pre>` - It solves 'Poem Problem' as everything inside is shown in browser with predefined formats like spaces, line breaks or new lines without use of `
` `<hr>` tags.

iii) `<hr>` - Horizontal Rule It is empty tag and used within two paragraphs to separate them in html page.

iv) `
` - Line Break without starting new para. It is empty tag. It starts content next line.

Formatting the texts inside paragraphs or somewhere else.

- i) `` - Bold text
- ii) `` - strong text just like strong
- iii) `<i>` - Italic text
- iv) `` - emphasized text just like italic
- v) `<mark>` - Marked i.e. Background is highlighted
- vi) `<small>` - small text
- vii) `` - Deleted i.e. line through text.
- viii) `<ins>` - Inserted text
- ix) `<sub>` - subscripted - X_2
- x) `<sup>` - superscripted - X^2

We can define the styles for this tags also inside stylesheets.

i) `<blockquote>` - used to quote the content from a site or resource browser indent this quotes.

eg.

```
<blockquote cite="URL" >_quote </blockquote>
```

ii) `<q>` - Inserts quotation marks `" "` for in between it.

iii) `<abbr>` - used for abbreviation. after hovering gives title/long form of abbreviation.

```
eg. <p> The <abbr title="Ticket checker">TC</abbr>
```

abbreviation with title useful for SEO's, translators.

iv) `<address>` - It's used to quote address info of author, browser renders it as italic.

e.g. `<address>`

Written by : Shubham `
`
 Visit us at : address `
`
 Phone no. : xxxxx84 `
`

`</address>`

v) `<cite>` - It is used to quote title of certain work or sculpture or artwork etc. It is used to quote the name of picture, special text like poem below it.

vi) `<bdo>` - used to override direction of text e.g. following reverses direction as shubham to mahbush

e.g. `<bdo dir="rtl">` Shubham `</bdo>`

Comments

`<!--` this is commented text will not be shown in browser or will not affect code. `-->`

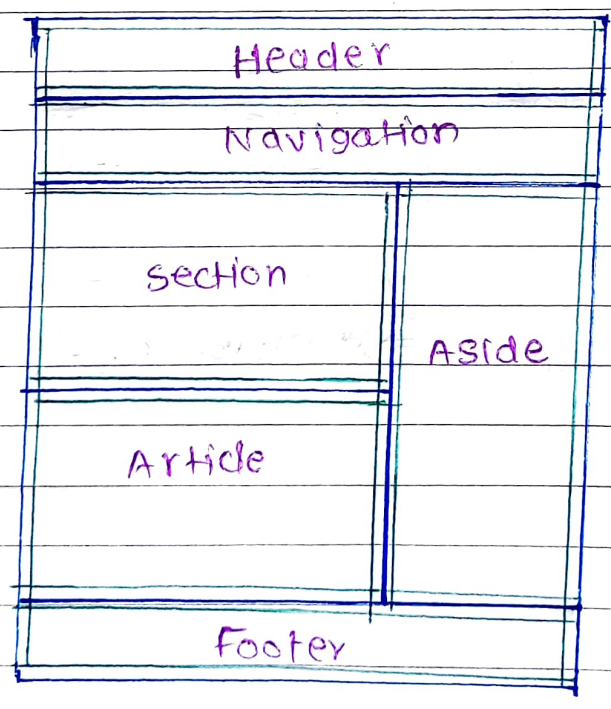
Footer

<footer> - It defines footer of webpage and includes contact information, link addresses etc.

Aside

<aside> - It is used to display side elements on the webpage. like sidebar.

Webpage layout



HTML semantics

semantic elements - It clearly describes its meaning to browser and developer.

e.g. semantic elements which clearly defines its contents are - `<form>`, `<table>`, `<article>`

Non semantic elements which do not clarify meaning are - `<div>`, ``

<article>

Specifies independent, self contained content and should be possible to distribute it independently from rest of the website.

eg. blog post, newsarticle.

<section>

Defines section in document
can be nested in `<article>` or `<article>`
can be nested to `<section>`,

<header>

Represent introductory contents

- 1) Icon, picture, logo
- 2) Heading elements (`h1 - h6`)
- 3) Authorship info.

there can be many headers but one header cannot be placed inside `<footer>` another `<header>` or `<address>`

<footer>

Represent contents like -

- 1) Related docs
- 2) Authorship info
- 3) Copyright
- 4) Back to top links
- 5) contact info.

There can be many footer elements.

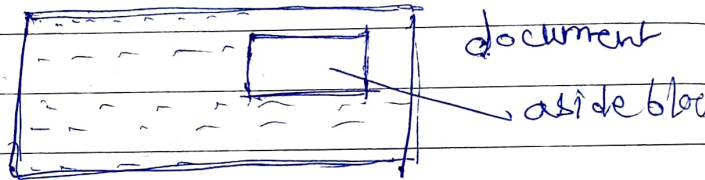
<nav>

defines set of navigation links. The set of links provides inside which are in the form of Navigation like. HTML | css | JS | jquery

<aside>

It include content which is directly related to main content idea and represented as cell or row or column as sidebar. like in newspaper.

e.g.



<figure>

<figcaption>

<figure> tag has self contained content like diagrams, illustrations, photo code listing etc while <figcaption> has caption for that figure and defined inside <figure> tag. defines actual image.

<details>

<summary>

<details> tag specifies additional details that user can open close on demand. it is oftenly used with <summary>

<details>

<summary> Expands </summary>

<P> this is expanded view </P>

</details>

Output

▶ Expands

⇒
on click ▶

▼ Expands

this is expanded view

Semantic comments

1) Well chosen content of H1 element is crucial for SEO (search engine optimization)

<h1>	</h1>	↓ most important
<h2>	</h2>	
<h3>	</h3>	

Browser shows these elements in different styles. But these elements are just to define the structure of the webpage and not for styling.

<h1> tag should have the central idea of the topic as it is the most important in the SEO point of view. Gives better ranking in SEO.

2) Semantic elements allow for a more meaningful expression of the structure of the HTML page.

<section>	article tag inside/nested to section tag much more sense but there is such a hard rule to use the same structure always.
<article>	
</article>	
</section>	

These things don't give more functionality to the webpage. They are just used to specify the structure of the webpage in a meaningful way.

Semantic elements make humans and computers better understand the structure and improve ranking in SEO.

HTML-Lists

Typical list made by spaces and new line are not lists as spaces ignored by HTML.

1) Ordered Lists

```

<ol>                                     ← ordered list start
  <li> Element 1 </li>
  <li> Element 2 </li>
  <li> Element with sublist
    <ol>
      <li> ele 1 </li>
      <li> ele 2 </li>
    </ol>
  </li>
  <li> Element 3 </li>
</ol>                                     ← ordered list end
  
```

list items.

nested ordered list

Output

- 1) Element 1
- 2) Element 2
- 3) Element 3 with sublist
 - 1) ele 1
 - 2) ele 2
- 4) Element 4

2) Unordered List

Use of unordered lists. `` `` don't order the items according to numbers.

```

<ul>
  <li>
  </li>
  <li>
  </li>
</ul>
  
```

3) Description List

<dl>

description list

<dt>

</dt>

list element title

<dd>

</dd>

list element description

<dt>

</dt>

list element title

<dd>

</dd>

list element description

</dl>

HTML Links - Hyperlinks

Internal links (Relative links) - The links which are giving local pages from same website.

Absolute links (External links) - The links for other websites.

Link <a>

 Name to link

URL with
or without
(www)

for extra info
of website
when we hover
over it.

Name that
appears on
which we
click.

Attributes of link

<a> - Hyperlink

<href> - URL

<title> - It gives title for link seen by hovering.

<target> - specifies where to open the linked document
It has following values.

1) self - opens the documents in same tab/window

2) blank - opens the documents in new tab/window

3) parent - opens the documents in parent tab/window

4) top - opens the documents in full body of window

e.g.

Name of link

 - Used for image as link it is used inside
<a> tag

<mailto> - Used inside href to send email.

 send email

Links to sections - Links to different sections of same page. also known as Bookmark

 #section1

 #section2

<section id = "section1" >

</section >

<section id = "section2" >

</section >

// #section is similar to selector by identity

Button as Link - For button we need to add some JavaScript codes.

<button onclick = "document.location = default.asp" >

HTML Tutorial

</button >

HTML - Iframes

It is used to display a webpage within webpage

<iframe src = "url" title = "Iframe" > </iframe >

OR

<iframe src = "url" name = "iframe" title = "Name" >

</iframe >

<p >

 </p >

Displaying Images

1) Image

```
<img src = "URL" alt = "Image name"
      height = "1000" width = "500" >
```

 - closing tag is not used.

style attribute can also used for styles.
 e.g. style = "width: 24px height: 10px"

It also supports .gif files for animations.

2) Picture <picture> - used to display different images for different fits according to device or window size.

<picture>

```
<source media = "(min-width: 650px)" srcset = "URL" >
<source media = "(min-width: 465px)" srcset = "URL" >
<img src = "URL" >
```

</picture>

<picture> can be also used when some browser don't support every img format we can define every format of same image.

<picture>

```
<source srcset = "img-girl.jpg" >
<source srcset = "img-girl.png" >
<img src = "img-girl.gif" alt = "girl" >
```

</picture>

Browser pick first matching image and ignores other elements (TOP to bottom approach).

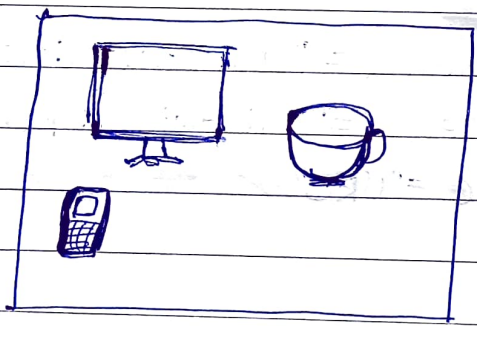
img always specified as last child element when other elements don't match (last one applied).

3) Image Map - It allows to click over specific areas of image and go to respective link

```

```

```
<map  
<area shape="rect" coords="..." alt="comp" href="comp"  
<area shape="rect" coords="..." alt="phone" href="phone"  
<area shape="rect" coords="..." alt="cup" href="cup"  
</map>
```



we can click on computer, cup, phone and go to respective links.

HTML Tables

Table tags - used to create table.

```

<table><caption> _____ </caption> // define table
  <tr> // define row
    <th> _____ </th> // define row data as
    <th> _____ </th> heads
    <th> _____ </th> // caption defines
  </tr> name to table.

  <tr>
    <td> _____ </td> //define row data as
    <td> _____ </td> data cells
    <td> _____ </td>
  </tr>

</table>

```

HTML IFrame

IFrames are used to display a web page within a web page.

```
<iframe src = "link" style = "border : 2px solid red;"
        title = "Iframe example">
</iframe>
```

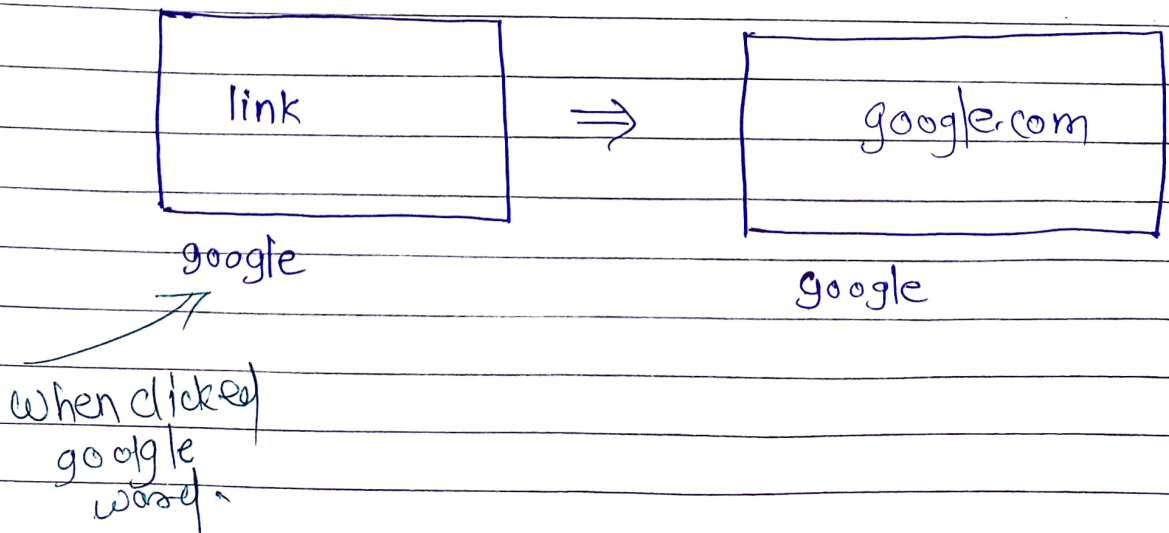
src attribute take link to display inside iframe
style attribute defines style applied to iframe
title attribute defines title for iframe It is not compulsory but good practise to mention it.

We can use IFrame as target window for link to open as follows-

```
<iframe src = "link" style = "border : 2px solid"
        name = "iframe_a" title = "Iframe example">
</iframe>
```

```
<P><a href = "https://www.google.com"
    target = "iframe_a" > Google </a> </P>
```

Output



HTML Block & Inline

<div> - - It is block level element used as container for other HTML elements.

- It starts block new line and take up as much width as necessary.

 - It is inline element starts next to element and also next element starts from same line.

- Used as markup of specific text.

HTML File Path

File paths are used when linking to external files like -

- i) Web pages
- ii) Images
- iii) style sheets
- iv) Java scripts.

1) Absolute file paths - It takes full URL

e.g. "https://www.google.com/images/picture"

2) Relative file paths - It is best practise to use relative file paths by this all links work in own computer as well as current public domain and future public domains.

`src = "picture.jpg"` - Picture located in the same folder as the current page.

`src = "images/picture.jpg"` - Picture located in images folder in the current folder.

`src = "/images/picture.jpg"` - Picture located in images folder at the root of the current web.

`src = "../picture.jpg"` - Picture is located in images folder one level up from the current folder.

HTML Forms

Form

`<form>` - all the form element are enclosed inside it.

Form elements

i) `<input>` - For taking inputs. have many attributes.

Input attributes

i) type =

i) "button" - It has functions like giving popup
e.g.

```
<input type="button" onclick="alert('Hello')"  
value="click here">
```

ii) "text" - for single line input field.
e.g.

```
<input type="text" id="username" name="username">
```

iii) "password" - defines password fill shown by asterisk or filled dots.

iv) "Submit" - To submit data inside form elements to form handler
form handler is server page for processing data it is specified in action attribute of form element

```
<form action="/action_page.php">
```

```
<input type="text" id="fname" name="fname">
```

```
<input type="submit" value="submit">
```

```
</form>
```

v) "reset" - It reset all form elements to default values, In the form of button.

```
<input type="reset">
```

vij) "radio" - user select only one choice out of all radio type input elements in form elements.

eg.

```
<form>
```

```
<input type="radio" id="m" name="gender" value="male">
```

```
<input type="radio" id="f" name="gender" value="female">
```

```
</form>
```

Combination of elements grouped with name attribute. One of option selected out of this grouped elements.

vii) "checkbox" - Allows user select multiple options. Similar to radio.

viii) "color" - Allow user to select colour from a list.

ix) "date" - Allow user to select date from a calendar.

```
<input type="date" id="bday" name="BirthDay">
```

OR

```
<form>
```

```
<input type="date" id="max" name="max" value="1972-11-11">
```

```
max = "1972-11-11"
```

```
<input type="date" id="min" name="min" value="1989-11-11">
```

```
max = "1989-11-11"
```

```
</form>
```

x) "datetime-local" - specifies date and time input field, with no time zone.

xi) "email" - Depending upon browser support email is validated and submitted automatically.
e.g.

```
<input type="email" id="email" name="email">
```

xii) "file" - To choose file upload from pc/phone storage.
e.g.

```
<input type="file" id="myfile" name="myfile">
```

xiii) "month" - allow user pick month and year.

xiv) "number" - To enter numbers and according to specified restrictions.
e.g.

```
<input type="number" id="@quantity" name="@quantity"
min="0" max="100" step="10"
value="30">
```

xv) "range" - slider provided with min, max, & steps by default min and max are at 0 to 100.

```
<input type="range" id="vol" name="vol"
min="50" max="100" step="2">
```

xvi) "search" - To search with default search engine of browser.

```
<input type="search" id="gsearch" name="gsearch">
```


xvii) "tel" - To enter phone number

```
<input type = "tel" id = "phone" name = "phone"
pattern = "[0-9]{3} - [0-9]{2} - [0-9]{3}" />
```

123 - 45 - 678

xviii) "time" - Pick time from time picker.

xix) "url" - Depending upon browser support
url is validated and submitted.

```
<input type = "url" id = "page" name = "page" />
```

xx) "week" - select week and year.

```
<input type = "week" id = "week" name = "week" />
```

ii) value = specify initial value / default value to any type field according to that type field.

iii) id = It has nothing to do with element data it used by CSS & JS

iii) name = It is variable name associated with data input and hence used common name for many elements grouped like radio where only one item selected.

iv) readonly ~~###~~ for read only value, specified after ~~value~~ attributes at last and have no value. Just specifies

v) disabled ~~###~~ specifier after value attributes. the space is unusable unclickable.

vi) size = specify width of Box in terms of no. of characters default value is 20.

vii) max length = max no. of characters as input.

viii) max = min = max and min is specified with different elements in case of date and in same element in case of numbers, range etc.
step =

vi) multiple ~~###~~ - for multiple file for type like file, email

x) Pattern = "[A-Za-z]{3}"
"[0-9]{3} - [0-9]{2} - [0-9]{3}"
works with data, time, search, ID,
password, tel, text.

xi) Placeholder = It is a short hint displayed in
Box used for - number, data, time,
placeholder = "12-34-5678"

xii) required ~~###~~ - It is defined for inputs which
compulsory to fill and prompt
not filled.

xiii) autofocus ~~###~~ - The element to be filled first
when page loaded is highlighted
to get focus first;
e.g. Box is having thick border.

xiv) list = list is provided as from which
is selected.

```
<form>  
<input list = "browser"  
<datalist id = "browser"  
  <option value = "chrome">  
  <option value = "safari">  
  <option value = "opera">  
</datalist>  
</form>
```

XV) autocomplete = Depending upon initial inputs browser predicts values. to insert this values are initially inserted once,

Values are = "on" or "off"

Entity References

1) To avoid Rendering issues -

<u>Instead of</u>	<u>Use</u>
1) $<$ - less than	<code>&lt;</code>
2) $>$ - greater than	<code>&gt;</code>
3) $&$ - ampersand	<code>&amp;</code>

2) provide characters not available on keyboard.

- 1) `©` copyright ©
- 2) `®` Registered trademark ®
- 3) `£` Pound £
- 4) `€` euro €

3) safeguard against more limited character encoding.

- 1) ` ` Nonbreaking Space - It doesn't allow the space to be split when line ends, characters on either side are always together.

- 2) `"` whole paragraph are contained inside two `"`, so that they displayed well in different browsers otherwise some characters may changed abruptly

4) Greek letters

- i) &Alpha α
- ii) &Beta β
- iii) &Gamma γ
- iv) &Delta Δ
- v) &Epsilon ε
- vi) &Zeta ζ

5) Symbols

- i) &larr ←
- ii) &rarr →
- iii) &uarr ↑
- iv) &darr ↓
- v) &spades ♠
- vi) &clubs ♣
- vii) &hearts ♥
- viii) &diams ♦

Entities are also specified by entity number

UTF-8 - character set is default in HTML. If not set, it includes characters for all symbols entities or emojis.

e.g.,

- 😊 is 128516 ⇒ 😀
- ♥ is 128151 ⇒ 💗

- i) UTF-8 shares same character to ASCII from 0
- ii) UTF-8 shares same character to ANSI & 8859 for values 160 to 255
- iii) UTF-8 have different characters from 256 to 101000 characters.

Responsive Design

Responsive design - site's layout adapts to the size of the device for this adaptability we use @media query.

- 1) Different screen sizes of different devices
- 2) screen modes - (landscape / portrait),

Syntax

@media (conditions) and / or (condition) {

P {

}

← stylesheet when condition of media query gets satisfied.

div {

}

}

conditions are - (min-width : --px);
(max-width : --px);
(orientation : Landscape / Portrait)

Example

@media (min-width : 900px) and (max-width : 1000px) {

P {

font-size : 17px;

div {

font-size : 17px;

}

}

this example changes font size of p and div elements to 17px when screen size become in between of (900-1000 px).

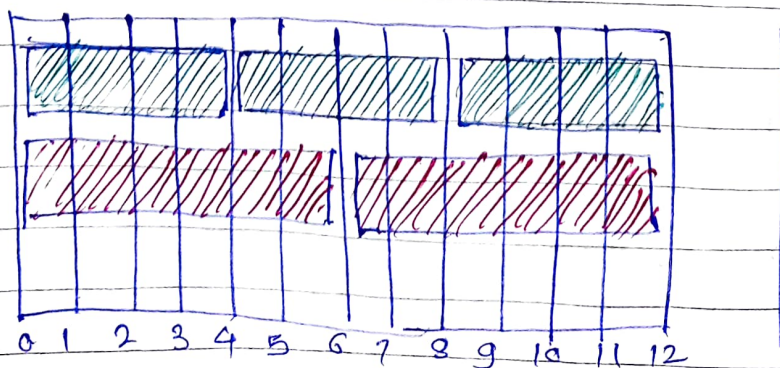
Breakpoint overlapping - when conditions of media query such that it becomes ambiguous to set style overlapping of breakpoint said to happen and it should be avoid carefully

The Alternative to Responsive design

- 1) There are alternatives like - server detects the type of browser compatibility and respective, loads version of site as there are already two versions (mobile/desktop) But
- 2) Even then mobile devices has different screen configurations hence to fit the page content properly responsive design is important,

12-Column Grid Responsive layout.

screen size/width is splitted into 12 columns (because of ease of operating with factors of 12) and design is made with respect to these 12 parts



Steps -

- 1) we define media query for specific screen size
- 2) Inside that media query we define 12 classes
- 3) For each class we define the width size
- 4) According to need this classes are assigned to the elements. i.e. when particular screen size is their that width is given to element,
- 5) we define many such media query with each having 12 classes and this classes are assigned to element only one condition is true at a time hence only one class is active for an element.

Example`<style>``*row { width: 100%; }``@media (min-width: 1200px) {`

`.col-1-1, .col-1-2, .col-1-3,`
`.col-1-4, .col-1-5, .col-1-6,`
`.col-1-7, .col-1-8, .col-1-9,`
`.col-1-10, .col-1-11, .col-1-12`

`{``float: left;``border: 1px solid green;``}``.col-1-1 {``width: 8.33%;``}``.col-1-2 {``width: 16.66%;``}``.col-1-3 {``width: 25%;``}`

There are 12 such class declarations until width becomes 100%

Another such media query is made for size less than 1999px (less than 2000px) and greater than 800px also.

one media query with less than 799px and greater than 400px they also have classes with widths like first media query.

e.g.

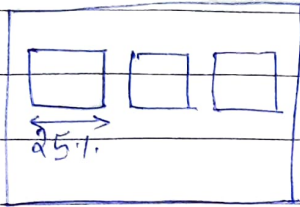
```

@media {min-width: 800px} and {max-width: 1999px}
.col-m-1, .col-m-2 -----
{
  float: left;
}
.col-m-1 {
  width: 8.33%;
}
.col-m-2 {
  width: 16.66%;
}
:
:
:
  
```

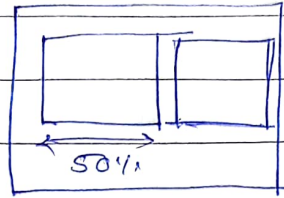
These classes are assigned to elements like
 <div class = "row">
 <div class = "col-1-3 col-m-6" col-s-12"> --<div>
 <div class = "col-1-3 col-m-6" col-s-12"> --<div>
 </div>

In this way when screen size changes elements gets different widths of screen, and they are aligned to left because of floating property

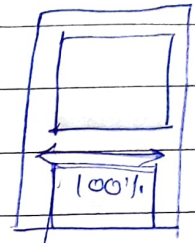
1200 PX



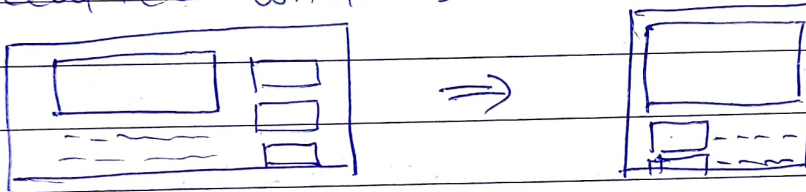
800 PX



400 PX



this is similar to youtube as we reduce size after certain point youtube video element occupies whole screen.



HTML URL ENCODE

URL - uniform resource locator

URL is used to address webpage or document on web means our browser requests document from web server using URL.

Web address follows syntax rule like -

Scheme :// **Prefix** . **domain** : **Port** / **path** / **filename**

Scheme - type of internet service (http/https/ftp)

prefix - domain prefix (Default for http is www)

domain - Internet domain name

Port - port number at host (http is 80)

path - path at server (if omitted: root directory)

filename - name of document.

Common URL Schemes

1) http (hypertext transfer protocol)	Common webpage Not Secure/Noency
2) https (secured hypertext transfer protocol)	secure webpage with encryption.
3) ftp file transfer protocol	downloading & uploading files
4) file	A file on computer

HTML Computer Code

1) `<kbd>` - used to defines keyboard keys.

e.g.

```
<P> <kbd>ctrl+s </kbd> </P>
```

2) `<code>` - used for computer codes (programmes)

e.g.

```
<P> <code>
  x+y = 5+6;
  print (x+y);
</code> </P>
```

3) `<samp>` - used to define computer output

e.g.

```
<P> <samp>
  This page is not found
</samp> </P>
```

4) `<var>` - used to define variable in programming or in mathematical expression.

e.g.

```
<P> In area of triangle.
  <var> h </var> is height and
  <var> b </var> is breadth
</P>
```

5) `<pre>` is used to contain `<code>` inside it so that `<code>` have white spaces inside it otherwise `<code>` doesn't maintain white spaces.