

MODULE - I

①

INTRODUCTION TO HTML

What is HTML and where did it come from?

→ The history begins with the ARPANET in 1960's
to the first public specification of the HTML by

Jim Berners-Lee - 1991.

→ Then, HTML's codification by the world wide
web consortium in 1997

→ In 1998, the W3C froze the HTML
specification at version 4.01. They stated that,

all computers may potentially understand
the published language used by the worldwide
web [the published language used by the worldwide
web is HTML (HyperText Markup Language)].

→ HTML is defined as a markup language.
→ A markup language is a way of annotating
a document in such a way as to make the annotation
distinct from the text being annotated.

→ Markup languages such as HTML, Tex, XML and XHTML allow users to control how text and visual elements will be laid out and displayed.

→ In simple, markup is a way to indicate information about the content that is distinct from the content.

→ This "information about content" in HTML is implemented via tags.

XHTML

→ Instead of growing HTML, the W3C turned its attention in the late 1990's to a new specification called XHTML 1.0, a version of HTML that used stricter XML syntax rules.

→ The goal of XHTML with its strict rules was to make page rendering more predictable by forcing web authors to create web pages without syntax errors.

Scanned by CamScanner

→ Like HTML, XML is a textual markup language set by the W3C.

→ A simple XML document:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<art>
  <painting id="290">
    <title> Balcony </title>
    <artist>
      <name> Monet </name>
      <nationality> French </nationality>
    </artist>
    <year> 1868 </year>
    <medium> oil on canvas </medium>
  </painting>
```

A XML document which follows all the syntax rules are called "well-formed".

→ XML-based syntax rules are:

- The main rules are:
 - * There must be a single root element.
 - * Element names are composed of any of the valid characters (punctuation symbols and spaces are not allowed)

- * Element names can't start with a number.
- * Element and attributes names are case sensitive.
- * Attributes must always be within quotes.
- * All elements must have a closing element.

→ Two versions of XHTML were created:

XHTML 1.0 strict and XHTML 1.0 Transitional.

strict version => rendered by a browser using the strict syntax rules. Tag supported and described by the W3C XHTML 1.0 strict specification.

transitional version => It is a forgiving flavor of XHTML, and was meant to act as a temporary transition to the eventual global adoption of XHTML strict.

→ As a key part of the standards movement in the web development community in 2000's was the use of HTML validators. → It is a means of verifying that a web page's markup followed the rules for XHTML transitional or strict.

→ The page has to adhere to strict rules for the code to be valid.

→ The code must be well-formed and correctly nested.

→ In mid 2000's, W3C presented a draft version of XHTML 2.0.

↳ It drops the backwards compatibility with HTML and XHTML 1.0.

↳ Browsers would become significantly less forgiving of invalid markup.

↳ It also dropped familiar tags such as ``, `<a>`, `` and headings such as `<h1>`.

↳ This development result in a gradual discomfort for browser manufacturers and web development communities.

HTML 5

→ A group of developers at opera and Mozilla formed the WHATWG (Web Hypertext Appn Technology working group) within the W3C.

→ WHATWG was focussed less on semantic purity and more on the web as it actually existed.

→ The work at WHATWG progressed quickly and by 2009, the W3C stopped work on XHTML 2.0 and instead adopted the work done by WHATWG and named it HTML5.

There are 3 main aims to HTML 5:-

- ① Specify unambiguously how browsers should deal with invalid markup.
- ② Provide an open, nonproprietary programming framework (via JavaScript) for creating rich web applications.
- ③ Be backwards compatible with the existing web.

HTML Syntax

Elements and Attributes:-

→ HTML documents are composed of textual content and HTML elements.

→ HTML Element is often used interchangeably with the term tag.

→ However, an HTML element is a more expansive term than tag that encompasses the name within angle brackets, i.e. a tag and the content within the tag.

→ An HTML element is identified in the HTML document by tags.

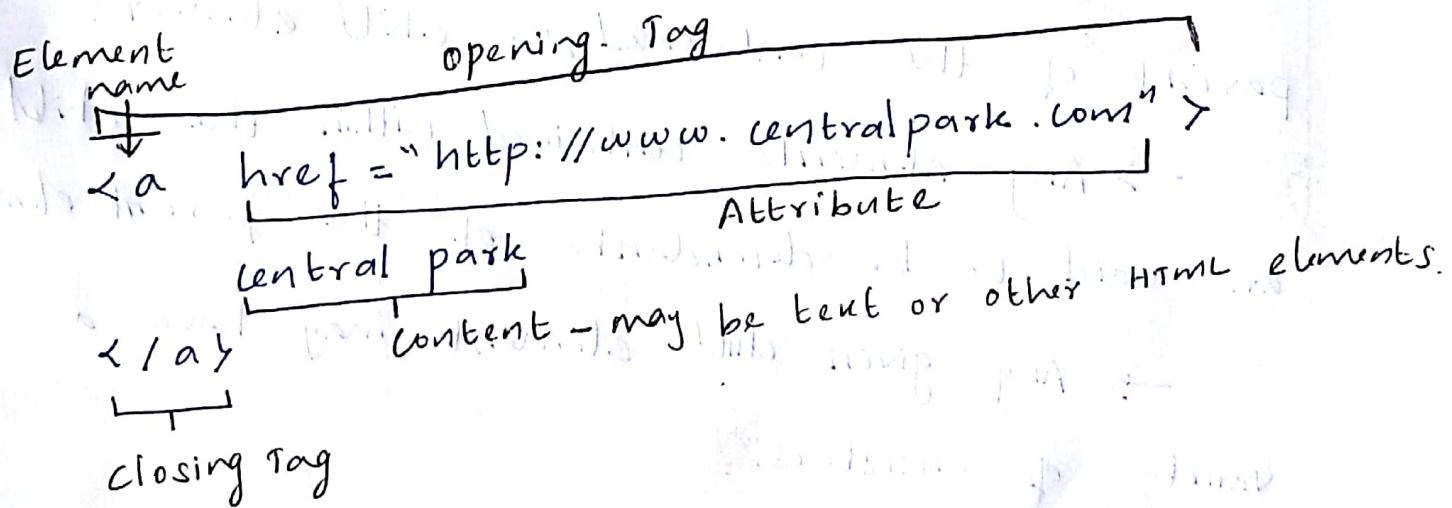
→ A tag consists of the element name within angle brackets.

→ The element name appears in both the beginning tag and the closing tag, which contains a forward slash followed by the element's name, again all enclosed within angle brackets.

→ HTML element can also contain attributes.

→ An HTML attribute is a name = value pair that provides more information about the HTML element.

→ HTML attribute values have to be enclosed in quotes. In HTML5, quotes are optional.



ex:2

 ↑ element name

 ↳ Trailing slash

The above example illustrates the different parts of an HTML element.

→ An empty element does not contain any text content; it is an instruction to the browser to do something.

→ The empty element had to be terminated by a trailing slash.

→ In HTML 5, the trailing slash in empty elements is optional.

* Nesting HTML Elements :-

→ HTML element will contain other HTML elements.

→ The container element is said to be a parent of the contained, or child element.

→ Any elements contained within the child are said to be descendants of the parent element.

→ Any given child element may have a variety of ancestors.

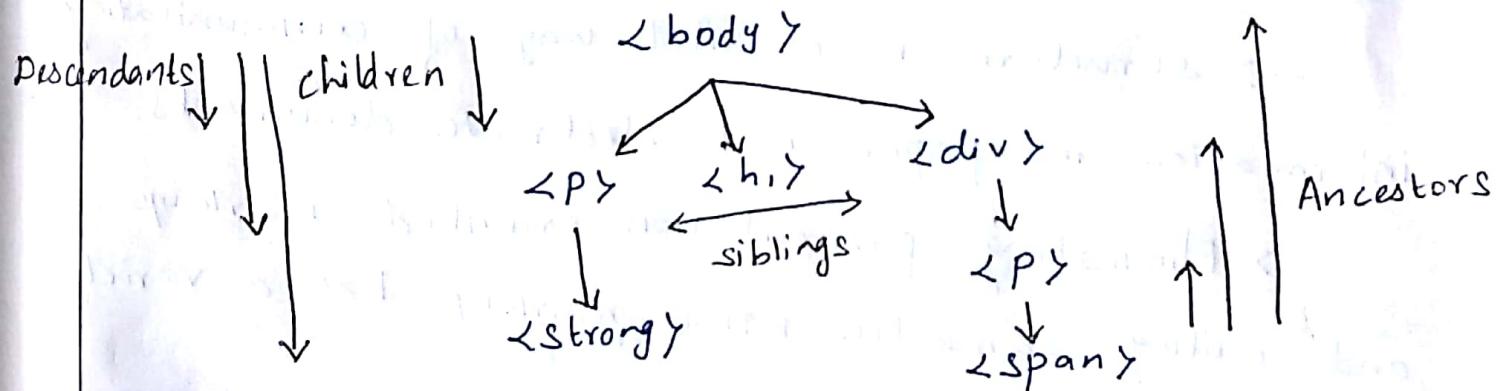
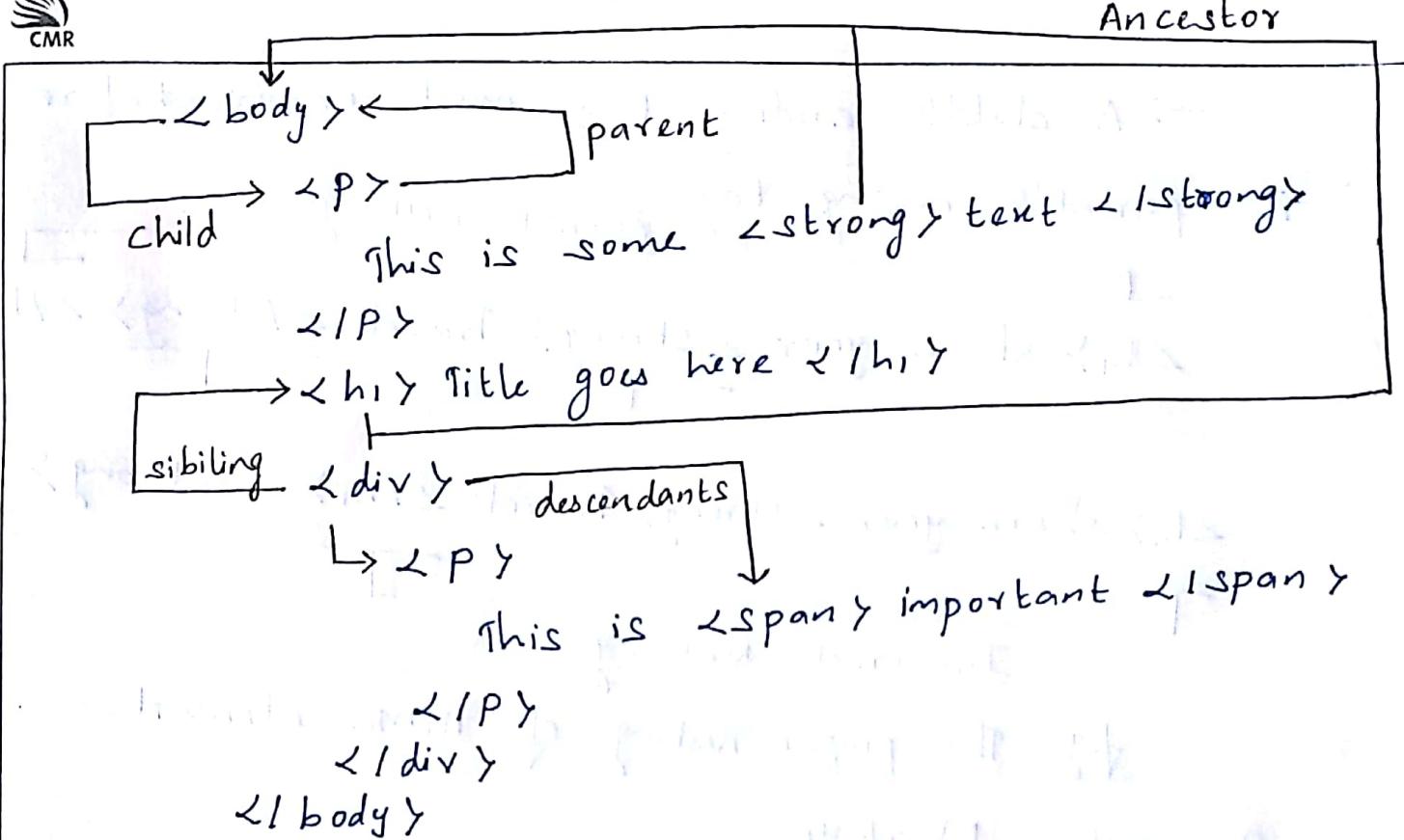


fig: HTML Document Outline.

→ In order to properly construct this hierarchy of elements, your browser expects each HTML nested element to be properly nested.

→ A child's ending tag must occur before its parent's ending tag. correct nesting

`<h1> share your Travels </h1>`

`<h1> share your Travels </h1> `

Incorrect Nesting

fig : The proper nesting of HTML elements

Semantic Markup :-

→ Structure is a vital way of communicating information in paper and electronic documents.

→ Eliminating presentation oriented markup and writing semantic HTML markup has a variety

of important advantages :

① Maintainability :-

→ Semantic markup is easier to update

and change than web pages that contains a great deal of presentation markup.

② faster :-

semantic web pages are typically quicker to author and faster to download.

③ Accessibility :-

Not all web users are able to view the content on web pages.
→ Users with sight disabilities experience the web using voice reading software.
→ visiting a webpage using voice reading software can be a very frustrating experience if the site does not use semantic markup.

④ Search Engine Optimization :-

For many site owners, the most important users of a website are the various search engine crawlers.
→ These crawlers are automated programs that cross the web scanning sites for their content, which is then used for users search queries.
→ semantic markup provides better instructions for these crawlers.
→ It tells them what things are important content on the site.

Structure of HTML Documents :-

e.g.: <!DOCTYPE html>

<title>

A very small document

</title>

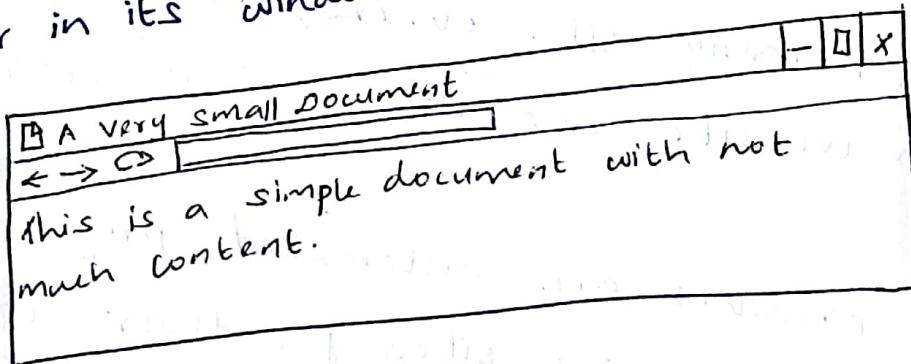
<p>

This is a simple document with not much content.

</p>

→ <title> element is used to provide a broad description of the content.

→ the title is not displayed within the browser window, Instead, the title is typically displayed by the browser in its window.



uses of title tag:-

- * It is used by the browser for its bookmarks and its browser history list.
- * The operating system use the page title in the windows task bar or mac dock.

* search engines will typically use the page's title as the linked text in their search engine results pages.
↳ it improves a page's rank in most search engines.

→ It provides a key role in determining what a given page is about.

→ Browsers limit the length of the title that is displayed in the tab or window to about 60 characters.

```
<!DOCTYPE html>
<html>
  <head lang="en">
    <meta charset="utf-8" />
    <title> share your travels -- New York -- central park
    <link rel="stylesheet" href="css/main.css" />
    <script src="js/html5shiv.js" />
    <script>
  </head>
  <body>
    Main Heading goes here <h1> ...
    ....
  </body>
</html>
```

→ The above example illustrates a complete HTML5 document.

DOCTYPE: [Document Type Definition]

→ This element tells the browser what type of document it is about to process.

→ It does not indicate what version of HTML is contained within the document.

→ HTML5 doctype is quite short in comparison to one of the standard doctype specifications for XHTML:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/
DTD/xhtml1-transitional.dtd">
```

→ The XHTML doctype instructed the browser to follow XHTML rules.

→ doctype was used to tell the browser to render an HTML document using the so-called

standards mode algorithm.

→ Or render it with the particular browser's older nonstandards algorithm called quirks mode.

→ Document Type Definitions (DTD) define a document's type for mark up languages such as HTML and XML.

→ In these both languages, DTD must appear near the beginning of the document.

→ DTD have their own syntax that defines allowable element names and their order.

→ The following code from the official XHTML DTD defines the syntax of the `<p>` element.

```
<!ELEMENT p (%inline;)*>
<!ATTLIST p
    %attrs;           -- attributes defined elsewhere --
    %textAlign;        -- values: left, center, right, alig
```

* Head and Body:

→ HTML 5 does not require the use of `<html>`.

→ `<head>` and `<body>` elements.

→ However, in XHTML, they were required.

→ The `<html>` element is sometimes called the root element as it contains all the other HTML elements in the document.

(16)

<head lang="en">

→ It also has a lang attribute. This optional attribute tells the browser the natural language that is being used for textual content in the HTML document is English.

→ This doesn't change how the document is rendered in the browser; rather search engines and screen reader software can use this information.

→ HTML pages are divided into two sections:

* the head <head> ... </head>
* the body <body> ... </body>

→ The head contains descriptive elements about the document, such as its title, any style sheet or Javascript files it uses and other types of meta information used by search engines and other programs.

→ The body contains content that will be displayed by the browser.

→ Head - carries of variety of other elements.

→ The first of these is the <meta> element.

<meta charset="utf-8"/>

↳ → The character encoding for the document is UTF-8. In brief, it's nothing but the character set standard is being implemented in the document.

↳ → Every character is represented by a standardized bit pattern.

↳ → The original ASCII standard of the 1950's defined English upper and lowercase letters as well as a variety of common punctuation symbols using 8 bits for each character.

↳ → UTF-8 is a more complete variable-width encoding system that can encode all 110,000 characters in the Unicode character set. [Supports 100,000 different long scripts]

```
<link rel="stylesheet" href="css/main.css"/>
```

↳ → This specifies an external CSS style sheet file that is used with this document.

↳ → Styles can also be defined within an HTML document.

↳ → For consistency's sake, most sites place most or all of its style definitions within one or more external style sheet files.

→ In this example, the file being referenced (`main.css`) resides within a subfolder called `css`.

`<script>src="js/html5shiv.js"></script>`

→ This references an external `JavaScript` file.

→ `JavaScript` code can be written directly within the `HTML` or contained within an external file.

Q. Elements :-

Quick review of HTML Elements :-

→ HTML contains many structural and presentation elements:

1) Headings :-

→ HTML provides six levels of heading hierarchy through `h1` through `h6`.

→ With the higher heading numbers indicating a heading of less importance `h6`.

→ Headers are an essential way of document

→ Headers are an essential way of document structure of authors to show their readers the structure of the document.

→ Headings are also used by the browser to

create a document outline for the page.

→ Every web page has a document outline.

→ Every web page has a document outline.

① <body> know as additional info this app
<h1> share your travels </h1>
② <h2> New-York - central park </h2>
③ <p> photo by Randy Connolly
archive = "http://www.centralpark.com/">
④
⑤
<h3> Reviews </h3>
⑥ <div> By Ricardo on 15, 2015 </div>
⑦ <div> By Susan on October 1, 2015 </div>
⑧ <div> copyright 2015 </div>
⑨ <div> I love Central Park </div>
<small> Share your travels </small>

→ like list of contents.

→ The document outline is used by the browser to render the page.

→ It is used by web authors when they write javascript to manipulate elements in the document or when they use CSS to style different HTML elements. → The browser has its own default styling rules.

→ for each heading level, the browser has its own default styling → These are easily modified and customized via CSS.

→ The below example shows some of the possible ways to style a heading.

HTML

```
<h1>SHARE YOUR TRAVELS</h1>
```

CSS

```
h1 {  
    margin: 0 0 0 0; padding: 20px; text-align:  
    center; color: #A61C07; font-family: 'Wosifer',  
    cursive; font-size: 60pt; line-height: 54pt;  
    background: url(images/header-background.jpg)  
    repeat-x; height: 120px;  
}
```

2) paragraphs and divisions:

- item ② defines two paragraphs, the most basic unit of text in an HTML document.
- `<P>` tag is a container and can contain HTML and other inline elements, i.e. `` & `<a>`.
- item ⑥ illustrates the definition of a `<div>` element.
This element is also a container element and is used to create a logical grouping of content. It has no intrinsic presentation; it is frequently used in contemporary CSS-based layouts to mark out sections.

3) Links:

- item ③ defines a hyperlink using the `<a>` element. The `a` stands for anchor.
- Links are created using the `<a>` element. A link has 2 main parts:
 - * The destination
 - * The label.
- The label of a link can be text or another HTML element such as an image.

Destination

HTML a href = "http://www.centralpark.com"

and central park \rightarrow modi

2/a and \rightarrow Label (text) of head going onward

2/a href = "index.html" and calls this with

 \rightarrow modi

2/a \rightarrow Label (link)

fig): Two parts of a link.

\rightarrow Anchor element can be used for a wide range of links. These includes

- * Links to external sites.
- * Links to other pages or resources within the current site.
- * Links to other places within the current page.

* Links to particular locations on another page.

* Links that are instructions to the browser to

start the user's email program.

* Links that are instructions to the browser to

execute a JavaScript function.

* Links that are instructions to the mobile browser

to make a phone call.

to do these things with



CMR 1. Link to external site

(23)

1. ` central park `
2. Link to resource on external site
` centralpark `
3. Link to another page on same site as this page.
` Home `
→ ` Go to Top of Document `
4. Link to another place on the same page.
` reviews for product `
``
→ Defines anchor for a link to another place on same page.
5. ` reviews for product `
link to specific place on another page.
6. Link to email.
` someone `
7. Link to JavaScript function.
` see this `
8. Link to telephone (Automatically dials the number when user clicks on it using a smartphone browser)
` call toll free (800) 922-0579 `

→ These are the different link destinations.

4) URL Relative Referencing:

→ When referencing page or resource on an external site, a full absolute reference is required.

→ A complete URL; a protocol (typically, http://), the domain name, any paths and then finally the file name of the desired resource.

→ When referencing a resource that is on the same server as your HTML document, use briefer relative referencing.

→ If the URL does not include the "http://" then the browser will request the current server for the file.

→ If all the resources for the site reside within the same directory, refer via their filename.

→ To add to many files in a single directory, a relative pathname is required along with the filename.

→ The pathname tells the browser where to locate the file on the server.

Locate the file on the server (e.g.)

- pathnames on the web follow unix conventions.
 - forward slashes (" / ") are used to separate directory names from each other and from file names.
 - Double periods (" .. ") are used to reference a directory "above" the current one in the directory tree.
- share your travels.

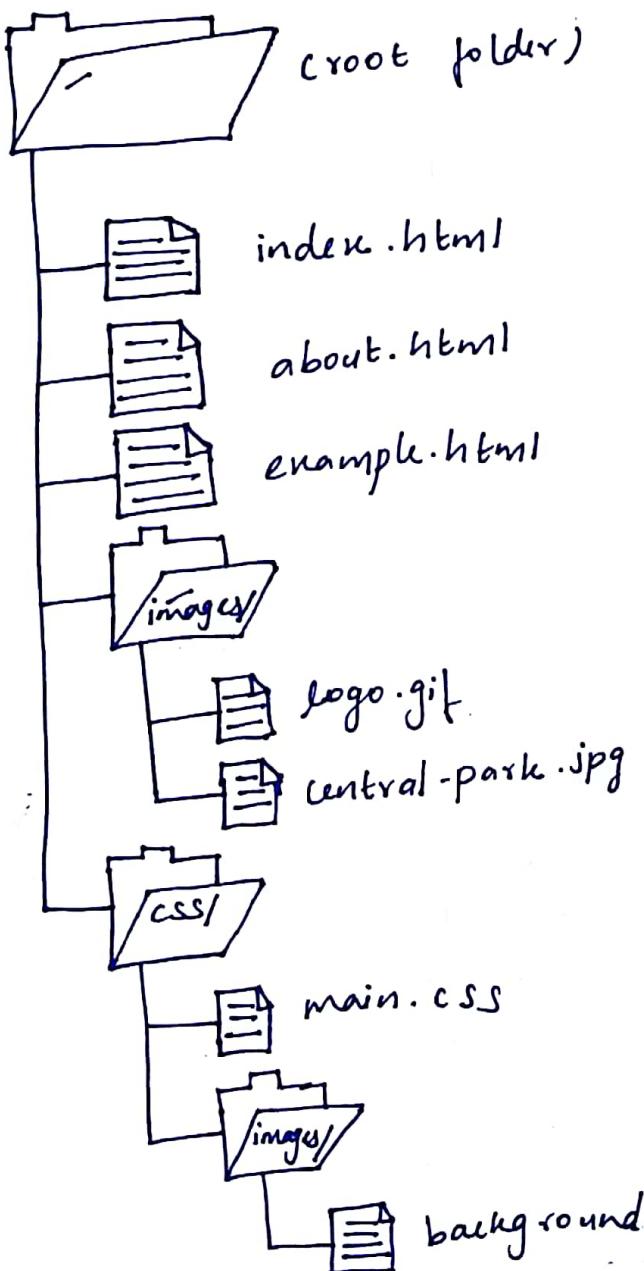


fig : Example site
directory tree.

Relative Link Type

① same directory

To link to a file within the same folder

Examples

``

② child directory

To link to a file within a subdirectory, use the name of the subdirectory and a slash before the file name.

``

③ Grandchild / Descendant directory

→ To link to a file that is multiple directories below the current one, construct the full path by including each subdirectory name before the file name.

``

④ parent / Ancestor directory

use ".." to reference a folder above the current one.

``

⑤ sibling directory

use ".." to move up to the appropriate level, and then use the same technique for child or grand child directories.

``

``

Relative Link TypeExamples.⑥ Root Reference

An alternative approach for ancestor and sibling references is to use the so-called root reference approach. Begin the reference with the root reference ("/").

NOTE: This will work only for server, not on your local machine. `http://localhost:8080/` are given

⑦ Default Document

Web servers allow references to directory names without file names. The web server will serve the default document, which is usually a file called `index.html` (apache) or `default.html`.

``
``

To link to `index.html` in `members` from `about.html`, use either
`` or
``

5) Inline Text Elements

The elements, which do not disrupt the flow of text i.e. cause a line break are called inline elements.

e.g. ` <time> <small>`

Examples

1. `<a>` - Anchor (links)
2. `<abbr>` - abbreviation
3. `
` - line break

4. `<cite>` - citation
5. `<code>` - display code
6. `` - Emphasis
7. `<mark>` - highlight text

8. `<small>` - display fine-print

9. `` - mark text that will receive style from CSS.

10. `` - strongly important. It will bold text.

11. `<time>` - time and date data.

6) Images :-

→ `img` tag defines an image tag after `div` with `img src = "images/central-park.jpg" alt = "Central Park"`

`title = "Central Park" width = "80" height = "40" />`

→ `` tag is the oldest method for displaying an image.

→ But it is not the only way, images can be added via `background-image` property in CSS.

7) Character Entities :-

→ These are special characters for symbols for which there is either no easy way to type them via keyboard or which have reserved meaning. e.g.: `<`, `>`

Entity Name Entity number

prescription (.) for p

non breakable space

less than symbol ("`<`")

greater than symbol ("`>`")

the © copyright symbol

1. `&nbsp` ` `

non breakable space

2. `<` `<`

less than symbol ("`<`")

3. `>` `>`

greater than symbol ("`>`")

4. `©` `©`

the © copyright symbol

& euro;
& trade;
& numl;

2 # 8364; The € Euro symbol
2 # 8482; The ™ Trademark symbol
2 # 252; The ü - ie) small u with umlaut mark.

8) Lists:

→ HTML provides three types of lists:

* unordered lists:- items in no particular order;

→ collections of items rendered by the browser as a

these are by default rendered by the browser as a bulleted list.

→ Unordered list have become the conventional way to markup navigational menus.

* Ordered lists:- items that have a set order;

→ collections of items rendered by the browser as a

these are by default rendered by the browser as a numbered list.

* definition lists:- name and definition pairs.

→ collection of terms and their definitions.

e.g.

```

<ul>
    <li><a href="index.htm">Home</a></li>
    <li><a href="#">About Us</a></li>
    <li><a href="#">Products</a></li>
    <li><a href="#">Contact Us</a></li>
</ul>

```

Output

- Home
- About Us
- Products
- Contact Us

 Introduction

 Background

 My Solution

: etc

 methodology

 Results

 discussion

 ...

 Conclusion

HTML5 Semantic Structure Elements :-

→ As HTML5 was being developed, researchers at Google and Opera had their search spiders examine millions of pages to see what were the most common id and class names.

→ Their findings helped standardize the names of the new semantic block structuring elements in HTML5.

→ The idea behind using these elements is that your markup will be easier to understand.

fig: sample <div>-based XHTML Layout with
(HTML5 equivalents).

Hand-drawn tree diagram of an HTML document structure:

```

<body>
  <div id="header">
    <div id="logo-headings">
      ...
    </div>
    ...
    <div id="top-navigation">
      ...
    </div>
  </div>
  <div id="main">
    <div id="left-navigation">
      ...
    </div>
    <div class="content">
      <div class="story">
        ...
        <div class="story">
          ...
        </div>
        <div class="story">
          ...
        </div>
      </div>
      ...
      <div>
        <img alt="blog-photo" class="blog-photo"/>
        <p class="photo-caption">...</p>
      </div>
      ...
      <div>
        <div class="related-stuff-on-right">
          ...
        </div>
        <div></div>
      </div>
    </div>
  </div>

```

The diagram uses numbered circles (1-9) to indicate nesting levels and specific sections:

- 1**: The outermost level, containing the main sections.
- 2**: A bracket groups the header and top navigation sections.
- 3**: A bracket groups the left navigation and content sections.
- 4**: A bracket groups the left navigation and main content sections.
- 5**: A bracket groups the content section.
- 6**: A bracket groups the story elements.
- 7**: A bracket groups the image and its caption.
- 8**: An arrow points from the "photo-caption" class label to the `<p>` element.
- 9**: A bracket groups the related stuff on the right and the final closing div.

1. This layout uses `<div id="footer">` instead of `<div>` for layout purposes.

2. `<div>` is semantic - `<div id="main">`

3. `<div>` is semantic - `<div id="header">`

4. `<div>` is semantic - `<div id="body">`

Example layout using new HTML5 semantic structure elements.

```
<body>
  <header>
    <hgroup>
      ...
      </hgroup>
      ...
      <nav>
        ...
        <nav>
          ...
          <h1>Main Title</h1>
        </nav>
        ...
        <nav>
          ...
          <ul>
            ...
            <li>Link 1</li>
            ...
            <li>Link 2</li>
            ...
            <li>Link 3</li>
          </ul>
        </nav>
      </header>
      <div id="main">
        <div>
          ...
          <article>
            ...
            <article>
              ...
              <article>
                ...
                <img alt="Image 1" />
                ...
                <img alt="Image 2" />
                ...
                <img alt="Image 3" />
              </article>
              ...
            </article>
            ...
          </div>
        </div>
      </div>
    </body>
```

```

<aside>
...
<aside>
</section>
<section>
...
</section>
<div>
<footer>
...
</footer>
</body>

```

* Header and footer :-

→ Header contains the site logo and title,

horizontal navigation links.

→ Footer contains less important material such as smaller text versions of the navigation, copyright notices, information about the site's privacy policy,

twitter feeds or links to other social media.

→ Header and footer can also be used in other

HTML5 containers.

e.g.: <article> <section>

```
<header>
```

```
  
```

```
  <h1> Fundamentals of Web Development </h1>
```

.....

```
</header>
```

```
<article>
```

```
  <header>
```

```
    <h2> HTML5 semantic structure Elements </h2>
```

```
    <p> By <em> Randy Connolly </em> </p>
```

```
    <p> <time> September 30, 2015 </time> </p>
```

```
  </header>
```

```
<article>
```

fig: Heading Example - Demonstrates both the uses of the hgroup element.

* Heading Groups:-

→ The hgroup element can be used to group

the headings together within one container.

→ The hgroup element should only contain

```
<h1>, <h2> etc ..
```

ex. <header>

```
  <hgroup>
```

```
    <h1> chapter-1 </h1>
```

```
    <h2> An Introduction </h2>
```

```
  </hgroup>
```

```
</header>
```

<article>: example: `<article> <h1> HTML5 semantic structure </h1>`

<hgroup>: adds a heading group structure elements `<h2> <h3> <h4> <h5> <h6>`

<h2> HTML5 semantic structure elements

<h3> overview `<h3> Overview </h3>`

<h4> `<h4> Details about article`

<h5> `<h5> Information about article`

* Navigation: `<nav>` `<nav> Home `

The `<nav>` element represents sections of a

page that contains links to other pages or to other parts within the same page.

→ HTML5 does not apply any special presentation to `<nav>` element.

→ It is used for major navigation blocks.

eg: `<nav role="navigation">`
` Home `
` About us `
` Browse `
` Contact `

<nav> `<nav> Home `

* Articles and section:

The `<article>` element represents a section of

content that forms an independent part of a document or site. eg) a magazine or newspaper article.

→ The section element represents a section of a document, typically with a title or heading.

* Figure and Figure captions:

→ The figure element represents some flow content, optionally with a caption, that is self-contained and is typically referenced as a single unit from the remaining flow of the document.

→ <figure> element can be used not just for images but for any type of essential content that could be moved to a different location in the page or document and the rest of the document would still make sense.

eg: <p> This photo was taken on October 22, 2011 with a Canon EOS 30D camera at 1/160 sec and f/2.8. </p>

<figure>

<figcaption> Conservatory pond in central

park.

</figcaption>

<figure>

</p>

* Aside :-

→ The `<aside>` element is similar to the `<figure>` element in that it is used for marking up content that is separate from the main content on the page.

→ It represents a section of a page that consists of content that is tangentially related to the content around the `<aside>` element.

→ This element is used for sidebars, pull quotes, groups of advertising images or any other grouping of non-essential elements.

What is CSS?

CSS - Cascading style sheets.

→ CSS is a W3C standard for describing the appearance of HTML elements.

→ It is used to define the presentation of HTML documents.

→ With CSS, we can assign font properties, colors, sizes, boxes, borders, background images on the page.

→ CSS can be added directly to any HTML element within the `<head>` element or to a separate text file that contains CSS.

Benefits of CSS:-

① Improved control over formatting :-
The degree of formatting control in CSS is significantly better than that provided in HTML. CSS gives web authors fine grained control over the appearance of their web content.

② Improved site maintainability :-
Websites become significantly more maintainable because all formatting can be centralized into one CSS file or a small handful of them. This allows you to make site-wide visual modifications by changing a single file.

③ Improved accessibility :-
css - driven sites are more accessible by keeping presentation out of the HTML, screen readers and other accessibility tools work better.

④ Improved page download speed :-
A site built using a centralized set of CSS files for all presentation will also be quicker to download because each individual HTML file will contain less style information and markup and thus be smaller.

⑤ Improved output flexibility :-
CSS can be used to adopt a page for different output media. This approach to CSS page design is often referred to as responsive design.

CSS versions :-
1990 → Variety of different stylesheets standards were proposed, Javascript stylesheets by netscape in 1996.

1996 → CSS Level 1

1997 → CSS Level 2

After June 2011 → CSS 2.1 become official W3C recommendation.

→ CSS 3 → manageable for both browser manufacturers
and web designers.

→ W3C divided it into a variety of different CSS 3 modules: CSS Selectors, CSS Namespaces, CSS media queries

and CSS Color.

Browser Adoption :-

→ 1996 ⇒ first major browser to support CSS.

2000 ⇒ IE5 ⇒ reached almost 100% CSS 1 support.

IE5, IE6, IE7 ⇒ uneven support for CSS 2.

→ since CSS was designed to be a styling language, text styling is quite easy.

→ However, CSS was not really designed to be a layout language, so authors often find it tricky

dealing with all floating elements, relative positions, inconsistent height handling, overlapping margins

and non-intuitive naming.

→ CSS 3 is a good step forward in terms of making CSS easier to use.

CSS syntax :-

A CSS document consists of one or more style rules.

→ A rule consists of a selector that identifies the HTML element, followed by a series of property: value pairs.

→ Each pair is also called a declaration.

→ The series of declaration is also called the declaration block.

Syntax:

```

selector { property1: value1;
            property2: value2;
            ...
            propertyN: valueN; }
    
```

→ Each declaration block contains multiple declarations separated by semicolons.

→ A declaration block can be put together on a single line or spread across multiple lines.

→ The browser ignores white space between your CSS rules, so you can format the CSS however you want.

→ Each declaration is terminated with a semicolon.

→ The semicolon for the last declaration in a block is optional.

eg: `em { color: red; }`

↳ `em` is `selector` → `value`
↳ `color: red;` is `declaration` → `property`

`selector : { declarations }`

```
P {  
    top right bottom left  
    margin: 5px 0 0 0;  
    font-weight: bold;  
    font-family: Arial, Helvetica, sans-serif;  
}
```

fig: CSS syntax.

1) Selectors :-

→ Every CSS rule begins with a selector.

→ The selector identifies which element or elements in the HTML document will be affected by the declarations in the rule.

2) Properties :-

→ Each individual CSS declaration must contain a property.

→ These property names are predefined by the CSS standard.

→ The CSS 2.1 recommendation defines over a hundred different property names.

property Type

	<u>property</u>
1. Fonts	font, font-family, font-size, font-style, font-weight, @font-face.
2. Text	letter-spacing, line-height, text-align, text-decoration, text-indent.
3. Color and Background	background, background-color, background-image, background-position, background-repeat, color.
4. Borders	border, border-color, border-width, border-style, border-top, border-top-color, border-top-width
5. Spacing	padding, padding-bottom, padding-left, padding-right, padding-top, margin, margin-bottom, margin-left, margin-right, margin-top.
6. sizing	height, max-height, min-width, width, min-height, min-width, width.
7. Layout	bottom, left, right, top, clear, display, float, overflow, position, visibility, z-index.

8. list-style-type, list-style, list-style-image, list-style-color.

* Values:

→ Each CSS declaration also contains a value for a property.

→ The unit of any given value is dependent upon the property.

→ Some property values are from a predefined list of keywords.

→ Others are values such as length measurements, percentages, numbers without units, color values and URLs.

→ CSS supports a variety of different ways of describing colors.

Method Description

1. Name : Uses one of 17 standard color names. CSS has 140 standard names.

2. RGB : Uses three different numbers between 0 and 255 to describe the red, green, and blue values of the color.

Example

color : red;

color : hotpink; /css3

color : rgb(255, 0, 0)

color : rgb(255, 65, 80);

3. Hexadecimal: uses a six-digit hexadecimal number to describe the red, green and blue values of the color; each value of the three RGB values is represented between 0 and ff. The hexadecimal number is preceded by a hash or pound symbol (#).

4. RGBA: This defines a partially transparent background color. The "a" stands for "alpha" which is a term used to identify transparency that is a value between 0.0 (fully transparency) and 1.0 (fully opaque).

5. HSL: Allows you to specify a color using Hue, saturation and light values. This is available in CSS 3.

eg.
color: hsl(0, 100%, 100%);

color: hsl(330, 59%, 100%);

→ There are multiple ways of specifying a unit of measurement.

→ Different devices have different physical sizes as well as different pixel resolution and because the user is able to change the browser's size or its zoom mode, these absolute units don't always make sense with web element measures.

Table: Different units of measure in CSS

	<u>Unit</u>	<u>Description</u>	<u>Type</u>
1.	px	pixel. In CSS2, this is relative measure, while in CSS3 it is absolute ($1/96$ of an inch).	Relative (CSS2) Absolute (CSS3)
2.	em	Equal to the computed value of the font-size property of the element on which it is used. When used for font-sizes, the em unit is in relation to the font size of the parent.	Relative.

3. vh A measure that is always relative to another value. The precise meaning of vh varies depending upon the property in which it is being used.
4. vw size in relation to the x-height of an element's font. (css3 only)
5. vch size in relation to the width of the zero ("0") character of an elements font
6. rem stands for root em; which is the font size of the root element. (css3 only) rem is constant throughout the entire document.
7. vw, vh stands for viewport width and height. Both are relative percentage values (blw 0 and 100)
8. in Inches Absolute
9. cm centimeters Absolute
10. mm Millimeters Absolute
11. pt points ($\frac{1}{72}$ of an inch) Absolute
12. pc pica ($\frac{1}{6}$ of an inch) Absolute

Location of styles :-

→ CSS style rules can be located in three different locations:-

1) Inline styles :-

→ Inline styles are style rules placed within an HTML element via the style attribute.

e.g.: `<h1>share your travels </h1>`

`<h2 style="font-size: 24pt; font-weight: bold;">`

Reviews

→ An inline style only affects the element it is defined within and overrides any other style

definitions for properties used in the inline style.

→ Selector is not necessary with inline styles and that semicolons are only required for separating multiple rules.

→ Using inline styles is generally discouraged since they increase bandwidth and decrease maintainability.

→ It can be handy for quickly testing out a style change.

2) Embedded style sheets :- Inside style tag (e.g.)

- Also called internal sheets.
- These are style rules placed within the style element. i.e) inside the <head> element of <style> element.

eg : <head> share your travels. </title>
<style>

```

h1 { font-size: 24pt; }
h2 { font-weight: bold; }
h2 { font-size: 18pt; }

<body>
    <h1> share your travels </h1>
    <h2> New York - Central Park </h2>
</body>

```

</head>

→ since each document (HTML) has its own style
 elements; it is more difficult to consistently style multiple documents when using embedded styles.

→ It is useful when quickly testing out a style that is used in multiple places within a single HTML document.

3) External style sheet: ~~style~~ apne behaddi ki (जो निम्न बांधकाम के लिए वापस आया है)

- These rules are placed within a external file with the .css extension.
- It provides the best maintainability.
- When you make a change to an external style sheet and HTML documents that reference that stylesheet, will automatically use the update version.

→ To reference an external style sheet, use a <link> element within the <head> element.

<link> element within the <head> element.

eg: <head lang="en">
<title> share your travel </title>
<link rel="stylesheet" href="style.css" />

→ We can link several style sheets at a time, each linked style sheet will require its own <link> element.

→ There are 3 different types of stylesheets:

① Author-created stylesheet. → inline, internal, external.

② User-style sheet. → user tell the browser to display pages based on individuals own custom stylesheet.

③ Browser-style sheet. → defines the default styles the browser uses for each HTML element.

→ Within each element we have a style attribute which contains styling information in form of style.

Selectors :-

→ To define CSS rules, use a selector, p, to tell the browser which elements will be affected by the property & values.

1) Element selectors :

→ It selects all instances of an given HTML element.

e.g.: `<h1> share your travels </h1>`

```
h1 {  
    font-size: 24pt;  
    color: yellow;  
}
```

→ To select all elements by using the universal element selector i.e.) * [Asterisk] character.

e.g. * { color: green; } // applies to all the html elements.

→ To select a group of elements by separating the different element names with commas, to reduce the size and complexity of your CSS file.

e.g.: `ul, li, p, div, aside {` margin: 0; padding: 0; `}`

2) Class selectors :-

→ To simultaneously target different HTML elements regardless of their position in the document tree.

→ It takes the form: period(.) followed by the class name.

e.g.: <head>

<style> .first { color: red; }

• first

font-style: italic;

color: red;

}

<style>.first { color: red; }

</head>

<body>

<h1 class="first">Reviews </h1>

<p class="first"> By susan on OCT 2014

3) Id selectors :-

→ To target a specific element by its id attribute regardless of its type or position.

→ If any HTML element has been labeled with an id attribute, it can be used for styling

→ It takes the form : pound/Hash (#)

Ex: <P id="latest">

Reviews by ricardo on 15 sep - 2015

HTML

"Finally</P>"

CSS #latest {

font-style: italic; color: red;

color: red;

}

4) Attribute selectors :- selects individual elements.

→ It provides a way to select HTML elements either by the presence of an element attribute or

by the value of an attribute.

eg: [title] { cursor: help; padding-bottom: 5px; }

CSS

<div> image 1
 image 2
 image 3 </div>

<div>

HTML

 image 1
 image 2
 image 3

</div>

| Selector | Matches | Example |
|--|--|--|
| [] | A specific attribute | [title] |
| [=] | A specific attribute with a specific value | a [title = "planet"] matches any <a> element whose title attribute is exactly "planet" |
| [~=] | A specific attribute whose value matches at least one of the words in a space-delimited list of words. | [title] ~="countries" |
| [^=] | A specific attribute whose value begins with a specified value. | a [href ^= "mailto"] |
| [*=] | A specific attribute whose value contains a substring. | img [src *="log"] |
| [#=] | A specific attribute whose value ends with a specified value. | a [href \$=".pdf"] |
| 5) <u>pseudo-Element</u> and <u>pseudo-class</u> <u>selectors</u> : | | |
| → To select something that does not exist explicitly as an element in the HTML document but which is still a recognizable selectable object. | | |
| <u>Syntax:</u> element name : pseudo-class selector name | | |

| <u>selector</u> | <u>Type</u> | <u>Description</u> |
|------------------|----------------|--|
| 1. a:link | pseudo-class | selects links that have not been visited. |
| 2. a:visited | " | selects links that have been visited. |
| 3. :focus | " | selects elements that have the input focus. |
| 4. :hover | " | selects element that the mouse pointer is currently above. |
| 5. :active | " | selects an element that is being activated by the user. |
| 6. :checked | " | selects a form element that is currently checked. |
| 7. :first-child | " | selects an element that is the first child of its parent. |
| 8. :first-letter | pseudo-element | selects the first-letter of an element. (for paragraph) |
| 9. first-line | " | selects the first line of an element. |

eg: <style>

a:link {

border: 1px solid black;

text-decoration: underline;

and now with color: blue;

}

.medium

and half when visited

a:visited {

text-decoration: underline;

and half when color: purple;

elsewhere normal

a: hover {

border: none;

color: green;

half remains same

besides a:active

{

background-color: yellow;

background:

<style>

banana

- 6) Contextual Selectors :-
- This is also called combinator.
 - It allows you to select elements based on their ancestors, descendants or siblings.
 - A descendant selector matches all elements that are contained within another element.
 - The character used to indicate descendant selection is the space character.
 - eg: `#main div p { ... }`
- Content ←**

Table

| selector | example | description |
|---------------------|--------------------------|---|
| 1. Descendant | <code>div p</code> | selects all <code>p</code> elements within <code>div</code> element. |
| 2. child | <code>div > h2</code> | selects direct child of specified element. |
| 3. Adjacent sibling | <code>h3 + p</code> | (selects the first <code>p</code> after <code>h3</code>) |
| 4. General sibling | <code>h3 ~ p</code> | (selects all the <code>p</code> elements that share the same parent as the <code>h3</code>). |

The Cascade : How styles interact

→ The cascade in CSS refers to how conflicting rules are handled.

→ The CSS uses the following cascading rules

principles:

1) Inheritance :- all basic modifications are inherited

```
eg 1 body {
    font-family: Arial;
    color: red;
    border: 8px solid green;
    margin: 100px;
}
div {
    font-weight: bold;
    margin: 5px;
    border: 1px solid green;
}
```

Inherited ← { ... } not inherited ← { ... }

→ In the first example, some of the properties are inherited, however all the text in the other elements in the document will be in Arial font and colored red.

2) specificity :-

→ It is how the browser determines which style rule takes precedence when more than one style rule could be applied to the same element.

class styling

eg:

<body>

this text is not within a p element

<p> reviews </p>

<div>

<p> By richards </p>

<p> easy on the HDR buddy </p>

</div>

<hr/>

<p> By susan </p>

<div>

<p class="last" id="very last">

thanks for posting! :)

</p>

</div>

</body>

→ It works in the browser is that the browser

assigns a weight to each style rule.

→ When several rules apply, the one with the greatest weight takes precedence.

→ Class selector take precedence over element selectors and id selectors take precedence over class selectors.

→ The precise algorithm the browser is supported to use to determine specificity is quite complex.

body {

font-weight: bold;

color: red;

}

div {

font-weight: normal;

color: green;

}

div:after {

color: yellow;

}

.last {

color: blue;

}

#very last {

color: orange;

}

div:after {

color: red;

}

#very last:after {

color: blue;

}

div:after {

color: green;

}

div:after {

color: black;

}

specificity value

element selector
overrides ①

```
div {
  color: green;
}
```

0001

descendant selector

example: .div

overrides ②

.class: red

class and attribute selector

.attr: red

overrides ③

id selector

overrides ④

id + additional
selectors

overrides ⑤

inline style

attribute style

```
div form {
  color: orange;
}
```

0002

example {

```
  color: blue;
}
```

```
a[href$=".pdf"] {
  color: blue;
}
```

first example {

```
  color: magenta;
}
```

div #first example {

```
  color: grey;
}
```

```
<div style="color: red;>
```

0100

0101

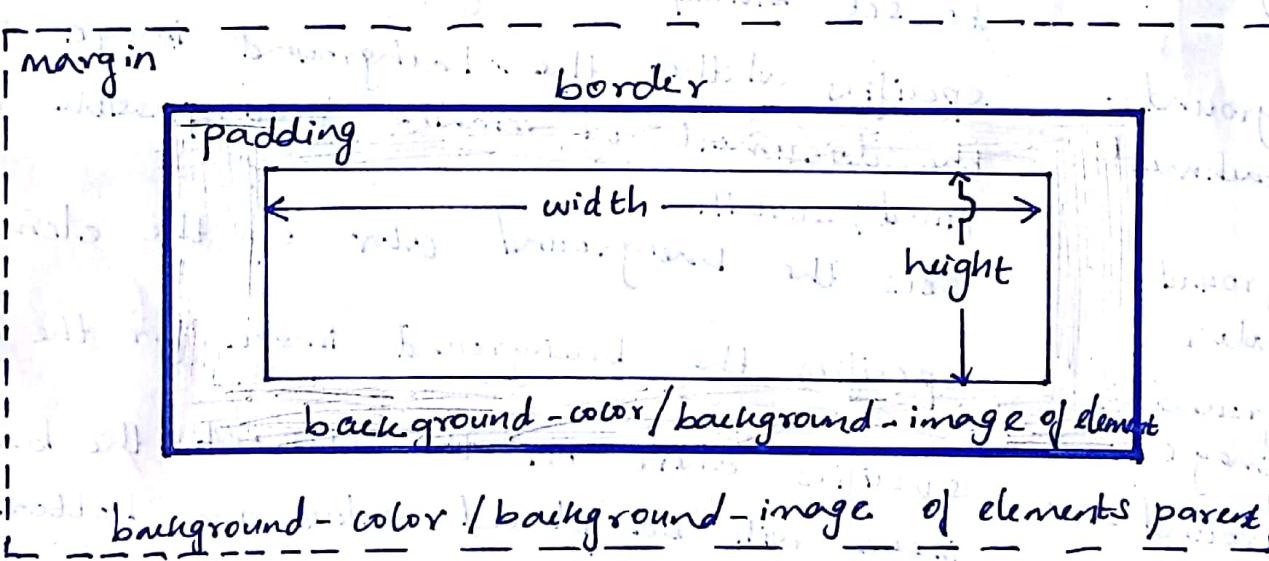
1000

→ When the same property is defined multiple times within a single declarations block, the last one will take precedence.

e.g.; #example { color: orange; color: magenta; } color: orange; \Rightarrow is applied overrides color: magenta;

The Box Model:-

→ In CSS, all HTML elements exist within an element box as shown in the below figure.



Every CSS rule begins with a selector. The selector identifies which element or elements in the HTML document will be affected by the declaration in the rule.

Background :- A property name used mainly as -

In the figure, the background color or image of an element fills an element out to its border.

→ In contemporary web design, it has become extremely common to use CSS to display purely presentational images rather than using the `img` element.

→ The below table lists the most common background properties.

Table 11.11

| property | description |
|-----------------------|---|
| background | A combined shorthand property that allows you to set multiple background values in one property. |
| background-attachment | specifies whether the background image scrolls with the document or remains fixed. possible values are fixed, scroll. |
| background-color | sets the background color of the element. |
| background-image | specifies the background image for the element. |
| background-position | specifies where on the element the background image will be placed. values are bottom, center, left and right. |
| background-repeat | determines whether the background image will be repeated. values: repeat-x, repeat-y, no-repeat |
| background-size | It modifies the size of the background image. |

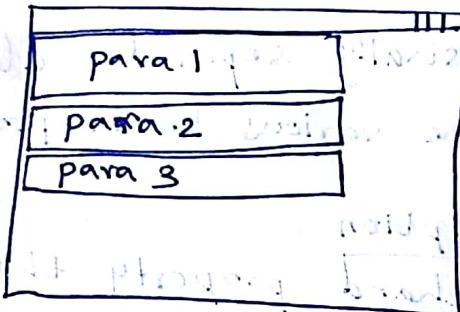
Borders:-

- It provides a way to visually separate elements.
- The following table lists the various border properties.

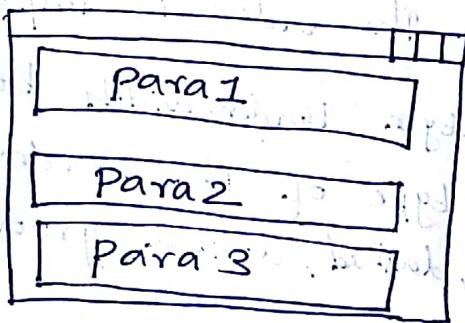
| <u>property</u> | <u>description</u> |
|------------------|---|
| 1. border | A combined shorthand property that allows to set the style. The order is important and must be: border-style border-width border-color. |
| 2. border-style | specifies the line type of the border. values are solid, dotted, dashed, double, groove, ridge, inset and outset. |
| 3. border-width | The width of the border in a unit. (not in pixels). A variety of keywords (thin, medium etc) are also supported. |
| 4. border-color | The color of the border in a color unit [border-color: top right bottom left]. |
| 5. border-radius | The radius of a rounded corner. |
| 6. border-image | The URL of an image to use as a border. |

Margins and Padding:-

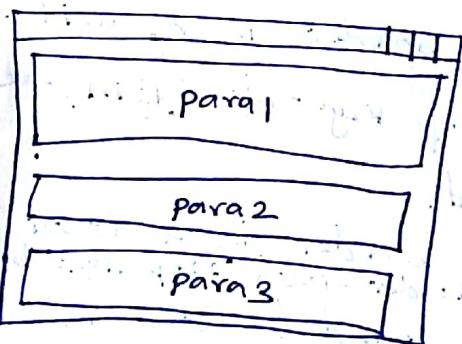
- Margins and padding are essential properties for adding white space to a web page which can help differentiate one element from another.
- Margins add spacing around an element's content.
- padding adds spacing within elements.
- Borders divide the margin area from the padding area.



```
P { border: solid 1pt red;
margin: 0;
padding: 0; }
```



```
P { border: solid 1pt red;
margin: 30px;
padding: 0; }
```



```
P { border: solid 1pt red;
margin: 30px;
padding: 30px; }
```

Box dimensions:-

→ Box dimensions are width and height properties.

eg: `div { box-sizing: content-box;
width: 200px;
height: 100px;
padding: 5px;
margin: 10px;
border: solid 2pt black; }`

→ In this program, width and height are absolute.

`outline: 2px solid black; border: 1px solid black;`

`border-sizing: border-box;`

border width is a property naming mistake.

→ It is possible to control what happens with the content if the box's width and height are not large enough to display the content using overflow property.

values: visible; hidden; scroll; auto.

CSS Text Styling:

Font Family:-

web font stack ⇒ a series of alternate fonts to use in case the original font choice is not on the user's computer.

e.g.: `p { font-family: serif; }` → serif is 3rd choice

`P { font-family: cambria, georgia, "times new roman"; }` → none is available → serif; choose default generic serif font.

→ If Cambria is not available then choose georgia.

→ The alternative fonts are separated by commas.

→ If the font name has multiple words then the entire name must be enclosed in quotes.



font stack

(66)

eg : font-family { "Hoefler Text", Cambria, "Times New Roman", serif; }

→ One common approach is to make your font stack contain in this order, in the following, ideal, alternative, common and generic.

→ In the above example,

Hoefler Text → ideal.

Cambria → alternative

Times New Roman → common

serif → serif

* Font sizes :-

eg: body { font-size: 100%; } // browser default font size is 16 pixels

h3 { font-size: 1em; } // $1.0 \times 16 = 16$ pixels

p { font-size: 1.125em; } // $1.125 \times 16 = 18$ pixels

h2 { font-size: 1.5em; } // $1.5 \times 16 = 24$ pixels

h1 { font-size: 2em; } // 2em is $2 \times 16 = 32$ pixels

→ To specify a font size, both em units and percentages are relative to the parents font size.

→ To over the issues using em and %, CSS supports a new relative measure rem.

paragraph properties

The below table lists the text properties:

| <u>property</u> | <u>Description</u> |
|---------------------|--|
| 1. letter-spacing | Adjust the space between letters. can be the value normal or a length unit. |
| 2. line-height | Specifies the space between baselines. the default value is normal or a length unit. |
| 3. list-style-image | Specifies the URL of an image to use as the marker for unordered lists. |
| 4. text-align | Aligns the text horizontally in a container element. values are left, right center and justify. |
| 5. text-decoration | Specifies whether the text will have lines below, through or over it. possible values are none, underline, overline, line-through and blink. |
| 6. text-direction | Specifies the direction of the text, left-to-right (ltr) or right-to-left (rtl) |
| 7. text-indent | Indents the first line of a paragraph by a specific amount. |

8. **text-shadow** A new CSS3 property that can be used to add a drop shadow to text.
9. **text-transform** Changes the capitalization of text. possible values are none, capitalize, lowercase and uppercase.
10. **vertical-align** Aligns the text vertically in a container element. values are top, middle and bottom.
11. **word-spacing** Adjust the space between words. values are normal or length unit.
12. **list-style-type** selects the marker type for ordered and unordered lists.
13. **line-height** sets the height of each line. values are normal, inherit, and length unit.
14. **font-size** sets the size of the font. values are normal, inherit, and length unit.
15. **font-weight** sets the weight of the font. values are normal, bold, and bolder.
16. **font-style** sets the style of the font. values are normal, italic, and oblique.
17. **font-family** sets the font family. values are serif, sans-serif, monospace, and cursive.
18. **font-color** sets the color of the font. values are black, white, and color name.
19. **font-decoration** sets the decoration of the font. values are underline, overline, line-through, and none.