

ANATOMY OF A HTML TAG:

- It has opening and closing tags, and they surround some content.
- Some tags have both but some have only opening tags such as (<hr> or
).
- Every HTML element has predefined attributes like id, class, which follow rules for its value.
- Attribute is a name-value pair, which should be unique for the whole web page.
- Attributes can only be specified on the opening tag.
- It is good practice to surround an attribute's value in quotes.
- We have to provide both opening and closing tags, even in the situation when we don't want to fill any content for the time being and fill it later dynamically.

BASIC HTML DOCUMENT STRUCTURE:

- Each document should start with: **<!doctype html>**.
- If we leave this declaration, it will signal the browser that our HTML doesn't follow HTML5 standards.
- Then comes **<html>** tag which contains the entire document.
- Then **<head> ... </head>** which describes main content, for e.g., character encoding, title, external resources, meta-data about the main content, etc. Title is necessary to be included.
- Then **<body> ... </body>** which is the root of all content visible to the user. (Referred as Viewport)
- The code is interpreted from top to bottom, sequentially.

HTML CONTENT MODELS:

- Content model refers to the full behavior the browser applies to the elements belonging to that content model and to their nesting rules of their elements.
- Which elements are allowed to be nested in which elements.
- All elements fall under 2 categories of traditional content model structure.

<u>Block-Level Elements</u>	<u>Inline Elements</u>
Renders to begin on a new line (default).	Renders elements on the same line.
Browser places the element in a new line.	Would act if there is no new line character.
Allowed to contain inline elements.	Allowed to only contain other inline elements.
Allowed to contain block elements.	Not allowed to contain block elements.
Roughly flow content (HTML5 Category Name)	Roughly Phrasing Content (HTML Category Name)

- DIV will push all the other inline elements in next line even if inline elements don't require a separate line but since DIV requires a separate line, it pushes them.

SUMMARY:

SEMANTICS:

- It has some inherent meaning; the names have some inherent meaning.
- Element that implies some meaning to the content.
- Elements tell something about the content.
- These elements should not be used for styling, they convey structure of your page.

TYPES OF SEMANTICS TAGS:

- Heading tags. (H1 – H6). H1 is the most important heading and H6 is the least important.
- Header tag. Contains information about the page, company log, tagline, navigation (<nav>) which is used to navigate within the website.
- Section tags. Article tags are usually contained within them.
- An article can also have sections within it.
- Aside tag. Communicates that something inside it is related to the main content of the page but not directly related to it.
- Footer tag. They have the footer information.
- All of these tags are block-level elements. We could have used DIV for all but semantics give meaning to the content!

LISTS:

- They are used to group related content together.
- Can be ordered as well as unordered.
- We can nest a list within an outer list item.

HTML CHARACTER ENTITY REFERENCES:

- If we want the browser to interpret special HTML characters as regular content, we need a way to escape them or tell the browser to not interpret them as HTML characters.
- For <, we use <
- For >, we use >
- For &, we use &
- HTML has a lot of different character entity references, one of them is copyright.
- For copyright symbol, we use ©
- Another entity reference () is used when we don't want some number of words or text or wrap (or we want them to wrap together). In other words, we don't want those words to split or go in a next line separately when browser screen gets reduced.

- We use this reference between words and we remove all other spaces.
- This is misused to add spaces, instead if we want spaces, we should wrap the content in a span tag and then apply margin to add extra space.
- Another useful entity reference is used when someone is trying to write an HTML based-email. Some characters get messed up (quotes). We use "

IMAGES:

- We use tag with src, width, height, and alt attributes.
- We can omit width and height but it can lead to some messy display of content.
- If our image doesn't get loaded or gets loaded slowly, then our content would jump if we hadn't specified width and height.
- Specifying width and height allows the browser to reserve a space for the image and the content don't jump then when it gets loaded slowly.

LINKS:

- We specify an anchor tag (<a>) with an attribute href containing an absolute or relative link.
- If we provide no directory information, the browser will assume that the file is in the same directory.
- It is also good to specify title attribute that helps visually impaired to get thru the webpage.
- The content between <a> ... is the content we will be able to click on to go to the link.
- <a> tag is both an inline and block-level element and we can place any element within it.
- We can also use a <div> within an <a> tag in which we can place the clickable content.

1. Internal:

- They point to internal webpages of the application.
- They point to different files of our main application.

2. External:

- They are hosted on a different domain than our website.
- Usually start with https.
- Target attribute is used with external links.

3. Fragment Identifier:

- Links that point to different sections of the same page/file.
- Method 1: we have id attribute for tags we want to link to, we use that id in href of <a> tag like #id_name.
- Method 2: we can have an <a> tag within an element with "name" attribute specified, we use that name in href of <a> tag like #name.
- They are very useful for Single Page applications.