

## FUNDAMENTAL HTML TERMS & CONCEPTS

TERM / CONCEPT	MEANING	SIMPLE EXAMPLE
HTML	The markup language used to structure content on a webpage.	<!DOCTYPE html>
Element	A complete piece of HTML made of an opening tag, content, and a closing tag.	<p>Hello</p>
Tag	The label wrapped in angle brackets that tells the browser what the element is.	<h1>
Attribute	Extra information added inside an opening tag to modify an element.	
Head	The part of the page that holds metadata (title, links, scripts).	<head>...</head>
Body	The visible page content — text, images, sections, etc.	<body>...</body>
Heading Tags	Titles and section labels from biggest to smallest.	<h1> to <h6>
Paragraph Tag	Basic text block.	<p>...</p>
Anchor Tag	Creates a link to another page or location.	<a href="https://...">Link</a>
Image Tag	Displays an image on the page; it does not need a closing tag.	
List Tags	Ordered lists (<ol>), unordered lists (<ul>), and list items (<li>).	<ul><li>Item</li></ul>

Div	A generic container used to group content.	<div>...</div>	
Span	An inline container for styling small bits of text.	<span>word</span>	
Class Attribute	Used to style or target multiple elements.	<div class="box">	
ID Attribute	Used to uniquely identify one element.	<div id="main">	
Semantic Tags	Meaningful page structure tags like <header>, <footer>, <section>, <article>.	<header>...</header>	
Comment	Notes that don't show on the page.	<!-- comment here -->	
DOCTYPE	Tells the browser the HTML version. Modern pages use HTML5.	<!DOCTYPE html>	
DOM (Document Object Model)	The browser's internal representation of the HTML page used for scripting.	No code, conceptual.	
Block vs Inline Elements	Block elements start on a new line; inline elements stay within text.	Block: <div> — Inline: <span>	

## HTML5 FUNCTIONAL ROLES & THEIR PURPOSES

HTML5 FEATURE / ELEMENT	WHAT IT DOES (FUNCTION)	SIMPLE EXAMPLE	
<header>	Defines the top section of a page or section (titles, nav, etc.).	<header>Site Title</header>	

<footer>	Defines the bottom section (credits, links, contact info).	<footer>© 2025</footer>	
<nav>	Wraps navigation links for menus and routing.	<nav><a href="#">Home</a></nav>	
<section>	Creates a meaningful content section.	<section>About Us</section>	
<article>	Represents a full, self-contained piece of content (blog, post, card).	<article>...</article>	
<aside>	Holds related but secondary content (sidebars, notes).	<aside>Tips...</aside>	
<main>	Marks the primary content area of the page.	<main>...</main>	
<figure>	Wraps images, charts, or media with an optional caption.	<figure><img><figcaption>...</figcaption></figure>	
<figcaption>	Provides a caption for a figure.	<figcaption>Description</figcaption>	
<audio>	Plays audio files directly in the browser.	<audio controls src="sound.mp3"></audio>	
<video>	Plays video with built-in controls.	<video controls src="video.mp4"></video>	
<canvas>	Draws graphics using JavaScript (games, charts, animations).	<canvas id="draw"></canvas>	
<svg>	Displays scalable vector graphics (icons, shapes).	<svg>...</svg>	
<progress>	Shows progress of a task.	<progress value="40" max="100"></progress>	

<meter>	Displays a value within a known range (e.g., battery level).	<meter value="0.7"></meter>	
<time>	Marks dates or times so browsers/AI can understand them.	<time datetime="2025-01-01">Jan 1</time>	
<mark>	Highlights text.	<mark>Important</mark>	
<details>	Creates a collapsible/expandable content box.	<details><summary>More</summary>...</details>	
<summary>	The title for the <details> toggle.	<summary>Click to expand</summary>	
<datalist>	Creates a dropdown suggestion list for inputs.	<datalist><option>Option</option></datalist>	
<output>	Displays the result of a calculation or script.	<output>0</output>	
Form Enhancements (required, placeholder, email, date, etc.)	Adds built-in validation and new input types without JavaScript.	<input type="email" required>	
Local Storage / Session Storage	Stores data in the browser for offline use — powered via JS.	JS-based, no direct tag.	

## CSS NOTES – CORE CONCEPTS & DEFINITIONS

CONCEPT	WHAT IT MEANS	SIMPLE EXAMPLE	
Selector	Targets an element to style.	p {}	
Property	The specific style you want to apply.	color:	
Value	The setting for that property.	color: red;	
Class Selector	Styles elements with a specific class.	.box {}	
ID Selector	Styles an element with a unique ID.	#main {}	
Universal Selector	Targets all elements.	* { margin: 0; }	
Descendant Selector	Styles elements inside other elements.	div p {}	
Padding	Space inside an element's border.	padding: 10px;	
Margin	Space outside an element's border.	margin: 10px;	
Border	The line around an element.	border: 1px solid black;	
Box Model	The full layout model: content → padding → border → margin.	Concept	

Display	Controls how an element behaves in layout (block, inline, flex, etc.).	display: flex;	
Flexbox	Layout system for aligning and spacing items easily.	display: flex; justify-content:center;	
Grid	Two-dimensional layout system for rows and columns.	display: grid; grid-template-columns: 1fr 1fr;	
Positioning	Controls element placement (static, relative, absolute, fixed, sticky).	position: absolute;	
Z-Index	Controls stacking order of overlapping elements.	z-index: 10;	
Color Values	Ways to set color: names, HEX, RGB, HSL.	color: #ff0000;	
Backgrounds	Controls background color, images, gradients.	background: url(pic.jpg);	
Fonts	Controls font family, size, weight, spacing.	font-size: 16px;	
Text Styling	Controls alignment, decoration, spacing.	text-align: center;	
Pseudo-classes	Target states of elements (:hover, :active, :focus).	a:hover { color: blue; }	
Pseudo-elements	Style parts of elements (::before, ::after).	p::after { content: "*"; }	
Media Queries	Make pages responsive to screen size.	@media (max-width: 600px) { ... }	
CSS Variables	Custom reusable values.	--main: #333; color: var(--main);	

Specificity	Rules that decide which styles win when there's conflict.	Concept	
Importing CSS	Link or import CSS into HTML or other CSS files.	<link rel="stylesheet" href="style.css">	
COMMENTS	leave a comment ignored by interpreter	/* */	

## POSITIONING

POSITION TYPE	WHAT IT DOES	KEY BEHAVIOR	
static	The default position for all elements.	Not affected by top, left, right, bottom. Follows normal page flow.	
relative	Moves an element relative to its original place.	Still takes up original space; offsets shift it visually.	
absolute	Removes the element from normal flow and positions it relative to the closest positioned ancestor.	If no ancestor is positioned, it uses the browser window.	
fixed	Stays locked in place on the screen when scrolling.	Positioned relative to the viewport. Common for navbars and buttons.	
sticky	Acts like relative until it reaches a scroll threshold, then sticks in place.	Great for headers that stay visible while scrolling.	

## SCALING

UNIT	NAME	RELATIVE OR ABSOLUTE?	WHAT IT MEANS / HOW IT SCALES
px	Pixel	Absolute	A fixed pixel on the screen. Does not scale with text settings.
%	Percentage	Relative	Percentage of the parent element's size (width/height/etc.).
em	Element Unit	Relative	Relative to the font-size of the element itself. Stacks when nested. Example: if parent is 16px, 2em = 32px.
rem	Root Element Unit	Relative	Relative to the root (html) font-size. Does not stack. Safer than em.
vh	Viewport Height	Relative to screen	1vh = 1% of the viewport's height.
vw	Viewport Width	Relative to screen	1vw = 1% of the viewport's width.
vmin	Viewport Minimum	Relative to screen	Based on the smaller of viewport width or height.
vmax	Viewport Maximum	Relative to screen	Based on the larger of viewport width or height.
fr	Fraction Unit (Grid only)	Relative	A “flexible fraction” of available grid space. Used only in CSS Grid layouts.
ch	Character Unit	Relative	Width of the character “0”. Useful for text-based sizing.
ex	x-height	Relative	Based on the height of lowercase “x” in the font. Rarely used.
pt	Point	Absolute	Print-based unit (1pt = 1/72 of an inch). Not ideal for screens.

cm	Centimeter	Absolute	Real-world centimeter. Uncommon in web design.
mm	Millimeter	Absolute	Real-world millimeter. Also uncommon.
in	Inch	Absolute	1 inch = 96px. Mostly for print styles.

<b>MAIN PROPERTIES (CONTAINER)</b>			
<b>PROPERTY</b>	<b>WHAT IT DOES</b>	<b>VALUES</b>	
flex-direction	Sets the direction of items.	row (default), row-reverse, column, column-reverse	
justify-content	Aligns items along main axis.	flex-start, center, flex-end, space-between, space-around, space-evenly	
align-items	Aligns items along cross axis.	flex-start, center, flex-end, stretch, baseline	
align-content	Controls spacing between multiple rows/columns.	Same as justify-content values	
flex-wrap	Allows items to wrap to the next line.	nowrap (default), wrap, wrap-reverse	
gap	Space between items.	ex: gap: 20px;	

<b>ITEM-LEVEL PROPERTIES</b>			
<b>PROPERTY</b>	<b>PURPOSE</b>	<b>NOTES</b>	
flex-grow	How much an item expands.	1 = takes remaining space.	
flex-shrink	How much item shrinks.	0 = never shrink.	

flex-basis	Starting size of an item.	Can use px, %, etc.	
flex	Shorthand: grow, shrink, basis.	Example: flex: 1 0 200px;	
align-self	Overrides container alignment for one item.	Same values as align-items.	

## DEFINING THE STRUCTURE

PROPERTY	WHAT IT DOES	EXAMPLE	
grid-template-columns	Defines column structure.	grid-template-columns: 1fr 1fr;	
grid-template-rows	Defines row structure.	grid-template-rows: 200px auto;	
repeat()	Repeat track patterns.	repeat(3, 1fr)	
fr Unit	Fraction of available space.	grid-template-columns: 2fr 1fr;	
gap	Space between rows and columns.	gap: 20px;	

## PLACING ITEMS

PROPERTY	FUNCTION	EXAMPLE	
grid-column	Place/size item across columns.	grid-column: 1 / 3;	
grid-row	Place/size item across rows.	grid-row: 2 / 4;	
grid-area	Name layout zones OR shorthand row/col.	grid-area: header; or grid-area: 1 / 1 / 2 / 3;	

<b>AUTO-PLACEMENT TOOLS</b>			
<b>PROPERTY</b>	<b>WHAT IT DOES</b>	<b>NOTES</b>	
grid-auto-rows	Sets row size for auto-created rows.	Example: 150px	
grid-auto-columns	Sets auto-created column sizes.	Example: 200px	
grid-auto-flow	Controls item flow.	row, column, dense	

  

<b>ALIGNMENT</b>			
<b>PROPERTY</b>	<b>PURPOSE</b>	<b>APPLIES TO</b>	
justify-items	Aligns content inside grid cells horizontally.	Items	
align-items	Aligns content inside grid cells vertically.	Items	
justify-content	Aligns the whole grid horizontally.	Grid	
align-content	Aligns the whole grid vertically.	Grid	



