



FULLSTACK WEB DEV

HTML

A large, bold, orange text "HTML" is displayed. To its right is a stylized orange folder icon. The word "HTML" is printed in white on the top flap of the folder. Inside the folder, there are white double-angle brackets "</>".

MASYNTECH



MASYNTECH



www.masynctech.com



ROLES OF



GETTING STARTED



ROLES OF HTML, CSS & JAVASCRIPT



Basic Explanation

- Website creation like making a movie
- **HTML**: Script outlining plot and characters
- **CSS**: Set and costume design setting tone and style
- **JavaScript**: Director, guiding interactions and dynamic changes based on user input



Deep Dive

- Analogy: Web development as building a house
- **HTML**: Foundational structure, like the house frame
- **CSS**: Styling and layout, like painting and furniture
- **JavaScript**: Dynamic interactions, akin to home automation
- Create cohesive and interactive web experiences

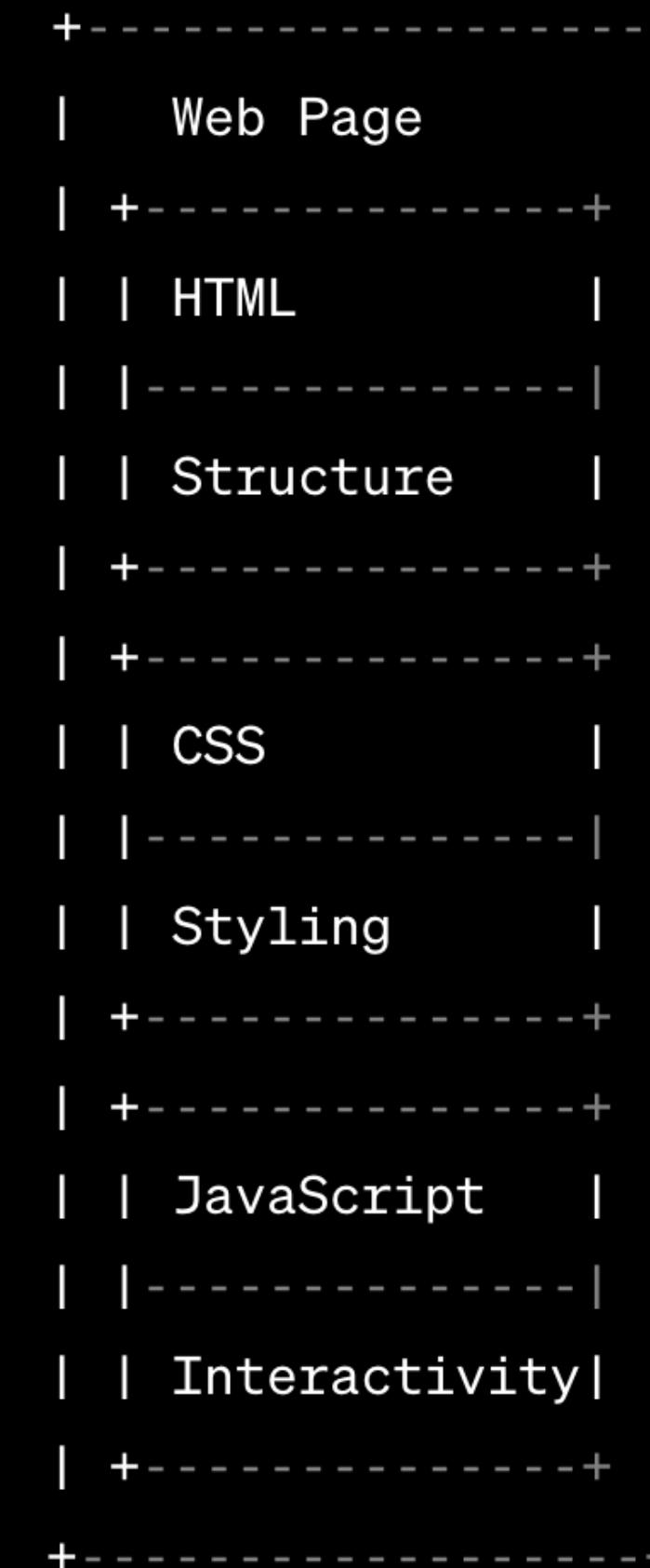


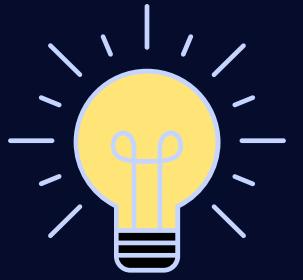
Analogue

- Web development as building a custom car
- **HTML**: Car's frame, shapes and sizes
- **CSS**: Paint job and interior design, aesthetics
- **JavaScript**: Engine and electronics, functionality and interaction

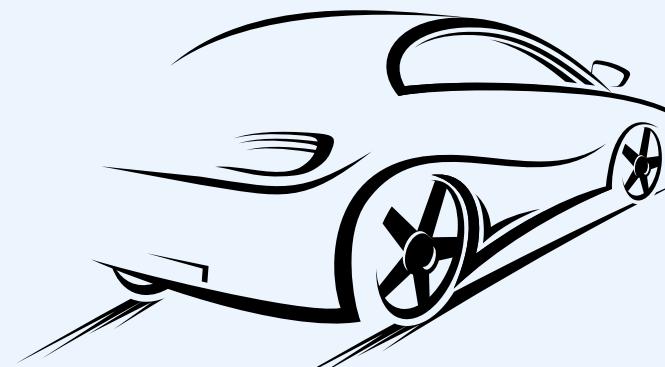


ROLES OF HTML, CSS & JAVASCRIPT

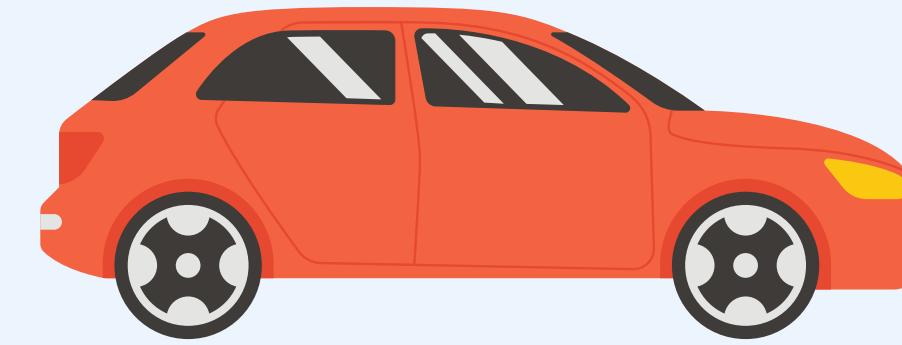




ROLES OF HTML, CSS & JAVASCRIPT



Noun



Adjective



Verb



ROLES OF HTML, CSS & JAVASCRIPT



Basic Explanation

- Website creation like making a movie
- **HTML**: Script outlining plot and characters
- **CSS**: Set and costume design setting tone and style
- **JavaScript**: Director, guiding interactions and dynamic changes based on user input



Deep Dive

- Analogy: Web development as building a house
- **HTML**: Foundational structure, like the house frame
- **CSS**: Styling and layout, like painting and furniture
- **JavaScript**: Dynamic interactions, akin to home automation
- Create cohesive and interactive web experiences



Analogue

- Web development as building a custom car
- **HTML**: Car's frame, shapes and sizes
- **CSS**: Paint job and interior design, aesthetics
- **JavaScript**: Engine and electronics, functionality and interaction



When to use?

- Web development: HTML, CSS, JavaScript harmony
- **HTML**: Sets the stage
- **CSS**: Adds visual flair
- **JavaScript**: Brings it to life



Summary

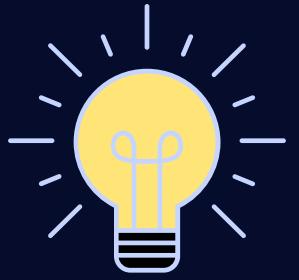
- HTML structure: `DOCTYPE`, `<html>`, `<head>`, and `<body>`.
- `<head>` holds meta-info and resources.
 - `<body>` has user-visible content.
 - Key for effective web development.



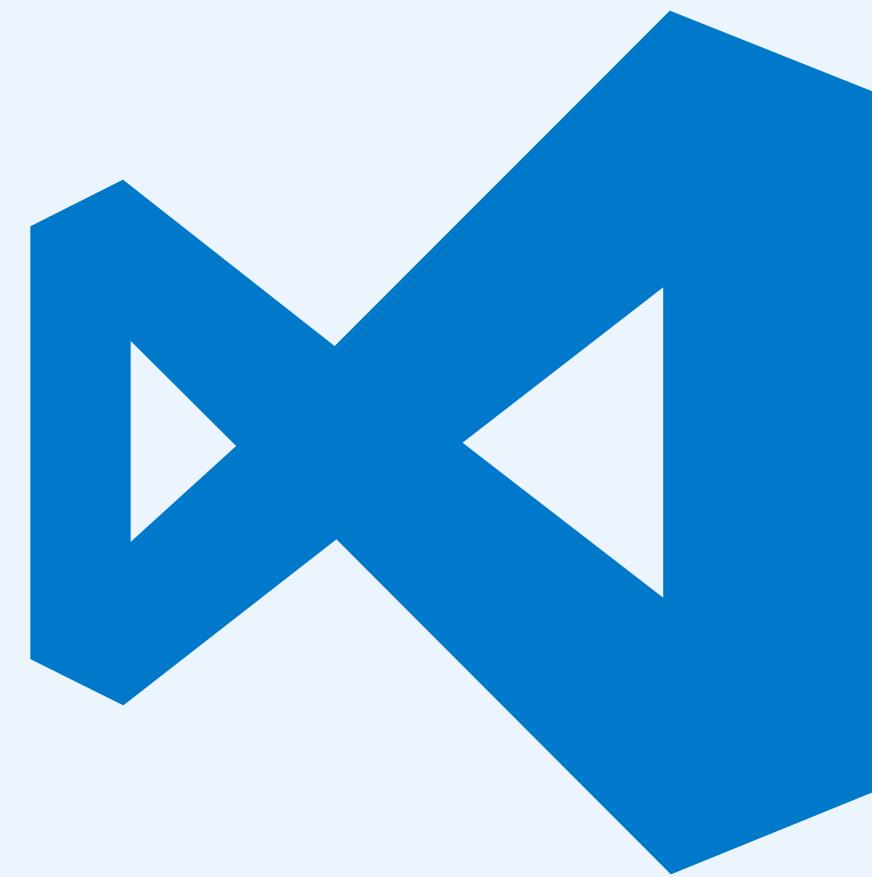
SETTING UP ENVIRONMENT



GETTING STARTED



OVERVIEW

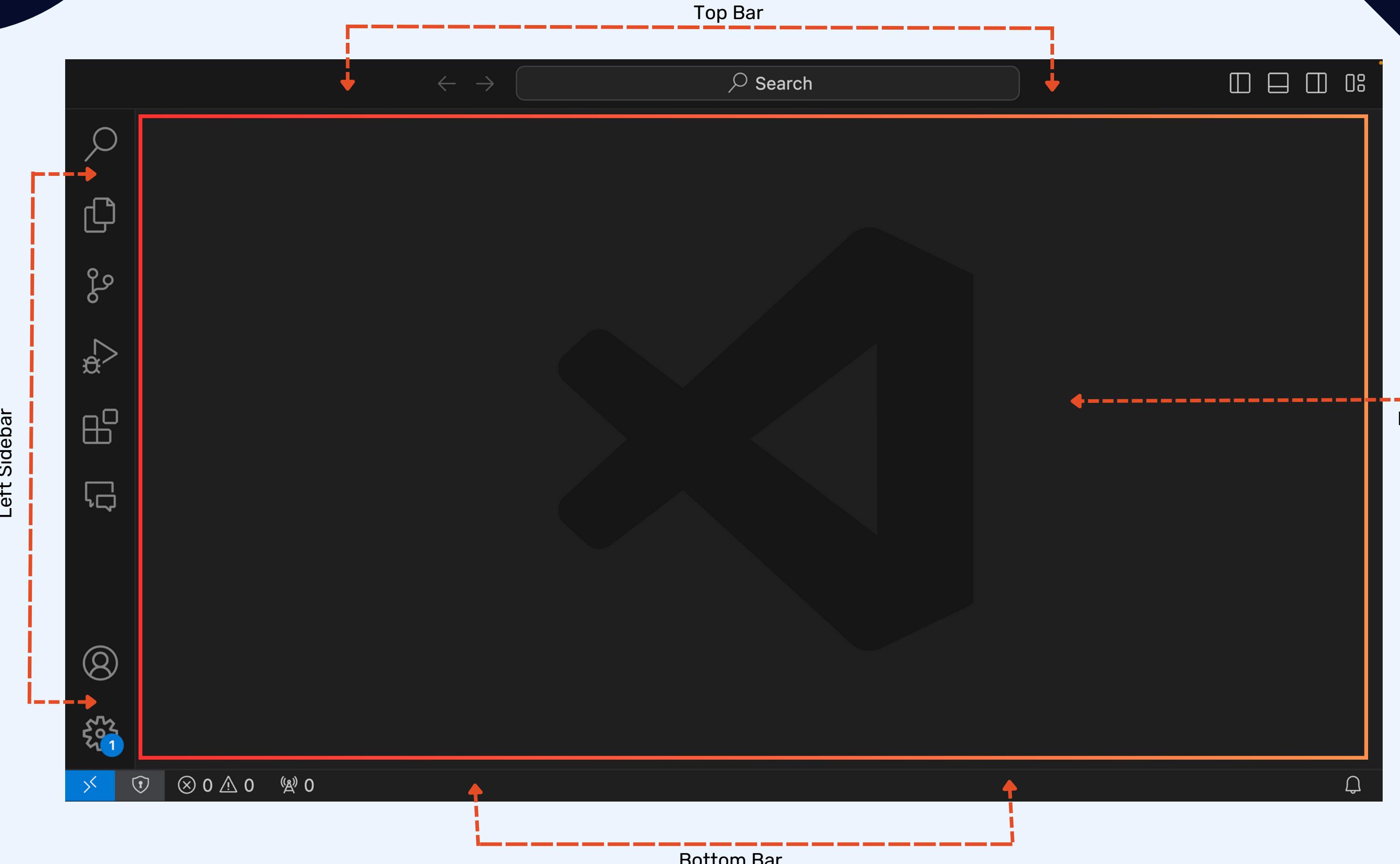




EXPORING VS CODE



GETTING STARTED





GETTING STARTED

DEMO TIME



GETTING STARTED

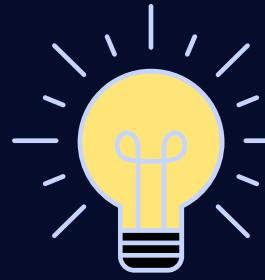


HTML

STRUCTURE



GETTING STARTED



HTML STRUCTURE



Basic Explanation

- HTML document structure:
- Declaration
- Head
- Body
- Similar to a book's cover, TOC, chapters, and index



Deep Dive

- HTML document components:
- `DOCTYPE` declaration for HTML version
- `` tag wraps entire content (root)
- `` section for meta-info, styles, resources
- `` contains visible content for the user



Analogue

- Analogy: Writing a screenplay for HTML structure
- `DOCTYPE`: Title page, script type
- ``: Characters, settings (not visible)
- ``: Actual script, dialogue, actions (visible to audience)



Important Rules

1. Begin with a `DOCTYPE` for browser compatibility.
2. Opt for semantic tags for clarity.
3. `` for meta-info, CSS, non-render-blocking JS.
4. Place visible content inside the `` tag.



When to use?

- HTML structure fundamental in web development
- Serves as the skeleton for projects
- Essential for blogs, web apps, and more



Summary

- HTML structure: `DOCTYPE`, ``, ``, and ``.
- `` for meta-info and resources.
 - `` for user-visible content.
 - Key for effective web development.



HTML STRUCTURE



Syntax

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
    <title>My First Webpage</title>  
</head>  
  
<body>  
    <h1>Hello, World!</h1>  
</body>  
  
</html>
```

- `<!DOCTYPE html>`: Declares HTML5 usage to the browser.
- `<html>`: Container for the entire webpage.
- `<head>`: The "brain" with metadata (not visible to users).
- `<title>`: Sets webpage title, displayed in browser tab.
- `<body>`: Contains user-visible content.
- `<h1>`: Header with large, bold text.
- A beginner-friendly introduction to HTML essentials.

DEMO TIME



GETTING STARTED



HTML TAGS



GETTING STARTED



HTML TAGS

Non-Self Closing Tags



```
<tagname attribute1="value1" attribute2="value2">Content</tagname>
```

Syntax

- `<**tagname**>`: Opening tag
- `attribute1="value1"`: Modify tag's behavior/appearance
- `Content`: Enclosed content
- `</**tagname**>`: Closing tag



HTML TAGS

Self Closing Tags



```
<tagname attribute1="value1" attribute2="value2" />
```

- `<tagname />`: Self-closing tag
- `attribute1="value1"`: Modify tag's behavior/appearance

Syntax



HTML TAGS



Basic Explanation

- HTML tags act as labels for browser display.
- `<p>` for paragraphs, `<h1>` for big headlines, etc.



Analogue

- Analogy: HTML tags as sandwich components
- Opening and closing tags as bread slices
- Content as the filling between slices
- Some tags have attributes as toppings or sauces



Important Rules

1. Close tags unless self-closing.
2. Lowercase tag names for consistency.
3. Quote attribute values (e.g., ``).
4. Prefer semantic tags for readability and SEO.
5. Properly nest tags (inner first, then outer).



Deep Dive

- HTML tags come in pairs: opening and closing.
- Opening tags may have attributes for behavior/appearance.
- Some tags are self-closing.
- Tags instruct the browser how to render enclosed content.



When to use?

- HTML tags create webpage elements, from headers to links.
- Building blocks for structured content presentation.



Summary

- HTML tags form webpage foundation.
- Pairs: opening and closing tags.
- Wrap content, may have modifying attributes.
- Crucial for web development mastery.

DEMO TIME



GETTING STARTED



HTML

COMMENTS



GETTING STARTED



HTML COMMENTS



Syntax

```
<!-- This is a comment -->
```

- `<!--`: Comment start.- `This is a comment`: Content, not displayed but in source.
- `-->`: Comment end.



HTML COMMENTS



Basic Explanation

- HTML comments: Not visible on the webpage.
- Appear in the source code.
- Primarily for developers to read and understand.



Analogue

- Analogy: HTML comments as penciled notes in a book
- Help remember important details
- For developers' eyes only



Important Rules

1. Avoid overusing comments to prevent clutter.
2. Maintain professionalism; no sensitive info.
3. Properly open (`<!--`) and close (`-->`) comments.



Deep Dive

- HTML comments start with ``.
- Content between delimiters is ignored by the browser.
- Place comments between tags, even within others.
- Annotate code for understanding and notes.



When to use?

HTML comments serve multiple purposes:

- **Documentation:** Explain complex sections.
- **Debugging:** Temporarily remove code for testing.
- **Placeholders:** Note areas for future work.



Summary

HTML comments: Start ``.

- Ideal for documentation, debugging, and clarity.
- Don't affect displayed content on the webpage.



TEXT FORMATTING TAGS



GETTING STARTED



TEXT FORMATTING TAGS

Blockquotes and Citations

- `<**blockquote**>`: Quotation from another source.
- `<**cite**>` attribute: Source specification.
- `<**q**>`: Inline quotation.
- `<**abbr**>`: Abbreviation/acronym, `title` for full version.
- `<**code**>`: Single-line code.
- `<**pre**>`: Preformatted text, fixed-width font.
- `<**address**>`: Contact info for document's author/owner.
- `<**cite**>`: References creative work title.
- `<**bdo**>`: Overrides text direction.
- `<**time**>`: Represents 24-hour time/date in Gregorian calendar.

DEMO TIME



GETTING STARTED



LIST TAGS





HTML LIST TAGS



Basic Explanation

- HTML tags for creating lists
- Similar to bullet points or numbering in documents



Analogue

- Analogy: Writing a shopping list
- Unordered list: Jotting items without order (bullet points)
- Ordered list: Prioritizing items (numbers)
- Description list: Terms and meanings



Important Rules

- `` in `` and ``
- `<dt>` with `<dd>` in `<dl>`
- Proper nesting, e.g., `` inside `` in ``



Deep Dive

- HTML lists: Unordered (``), Ordered (``), Description (`<dl>`)
- **Unordered:** Bullet points
- **Ordered:** Numbers/letters
- **Description:** Terms and descriptions



When to use?

- `` (Unordered List): Order doesn't matter, e.g., ingredients.
- `` (Ordered List): Sequence matters, e.g., recipe steps.
- `<dl>` (Description List): Glossaries, key-value pairs, e.g., dictionary.



Summary

- HTML list types: ``, ``, `<dl>`
- Structure information logically for readability

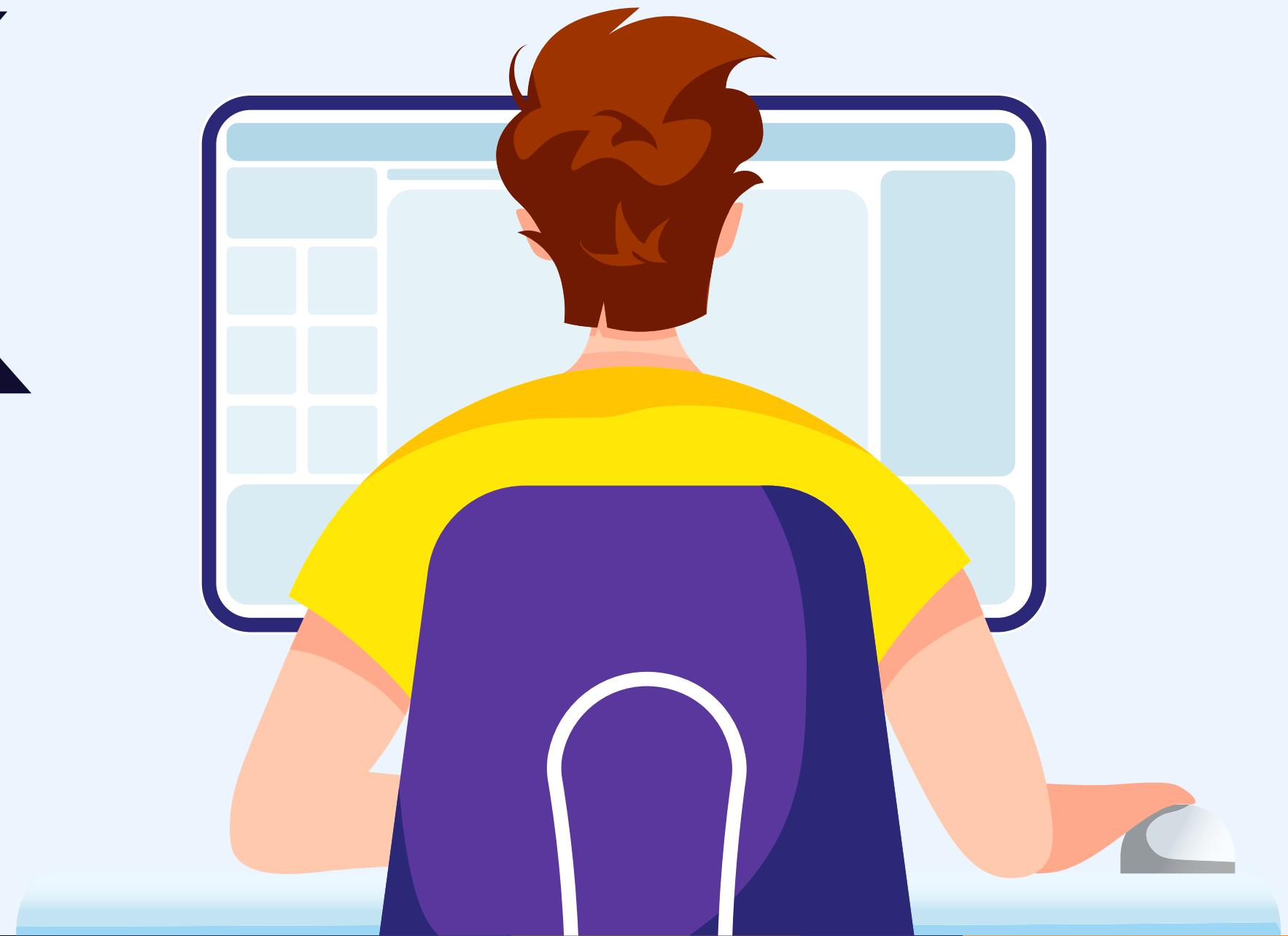


DEMO TIME





LINK TAGS





HTML LINK TAG



Basic Explanation

- `<a>` tag: Creates hyperlinks in HTML
- Enables navigation between webpages or within the same page



Analogue

- Analogy: Digital magazine clickable elements
- Clicking headlines for full stories
- Clicking advertisements to go shopping
- Implemented using `<a>` tag in HTML



Important Rules

- Make links distinguishable from text.
- Use descriptive link text, not "click here."
- Be cautious with `target="_blank"` for external links.
- Provide `alt` descriptions for image links.
- Avoid linking to harmful or irrelevant sites for SEO and trust.



Deep Dive

- `<a>` tag usage:
 - Link to another webpage
 - Link to a different page section
 - Link to an email address
 - Link to downloadable files
 - Link to other protocols (e.g., `tel:` for phone numbers)
- Primary attribute: `href` specifies link destination



When to use?

- `<a>` tag for various purposes:
 - Navigating to another website
 - Page navigation within content
 - Downloads (PDFs, software)
 - Initiating actions (email, phone)



Summary

- `<a>` tag: Versatile for creating hyperlinks in HTML
- Enables webpage navigation, resource access, and more.



DEMO TIME





DIV TAG



HTML TAGS



HTML DIV TAG



Basic Explanation

- `<div>` tag: Container in HTML
- Groups elements together
- Lacks specific semantic meaning
- Primarily used for styling (CSS) and scripting (JavaScript)



Analogue

- Analogy: Packing belongings for a move
- `<div>` as a box for grouping content
- Decorate or label it as needed
- Like using `<div>` to structure and style web content



Important Rules

- Prefer semantic HTML elements when applicable.
- Keep related content within a `<div>`.
- Avoid excessive use of `<div>` for readability and maintainability.



Deep Dive

- `<div>` tag: Block-level element
- Starts on a new line, full width
- Groups elements for unified manipulation
- Primarily for styling, layout control



When to use?

- `<div>` tag use cases:
- Grouping content for styling or layout
- Creating diverse page layouts with CSS
- Targeting groups of elements with JavaScript



Summary

- `<div>` tag: Versatile container
- Groups elements for styling and content management on webpages.

DEMO TIME





IMAGE & PICTURE TAGS



HTML TAGS

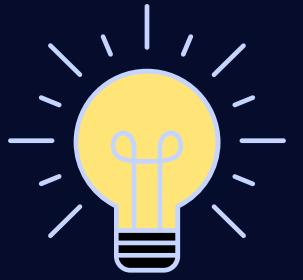


IMAGE AND PICTURE TAGS



```

```

Syntax

- `` attributes:
- `src`: Specifies image URL.
- `alt`: Provides text description for accessibility and when image is unavailable.



IMAGE AND PICTURE TAGS



Syntax

```
<picture>
    <source media="media_query" srcset="URL_of_image1">
    <source media="media_query" srcset="URL_of_image2">
    <!-- More <source> elements can be added if needed -->
    
</picture>
```

- `<source>` attributes:
 - `media`: Specifies media condition for display.
 - `srcset`: Specifies image source URL.
- `` inside `<picture>`:
 - Fallback for unsupported `<picture>` or unmet media conditions.

Customize image display for various devices and scenarios, enhancing user experience.



IMAGE AND PICTURE TAGS



Basic Explanation

- `` tag for displaying images.
- `<picture>` tag for responsive images with multiple versions.
- Adapt images to different devices and resolutions.



Deep Dive

- `` tag for displaying images with a fixed source (`src`).
- `<picture>` tag with `<source>` element for responsive images.
- Browsers select image source based on factors like screen size or file type support.



Analogue

- Analogy: Watching a movie on different screens
- `<picture>` adapts image quality based on viewing conditions
- `` shows one standard version of the image



When to use?

- `` for simple images.
- `<picture>` for responsive design across devices.



Important Rules

- ``: Include `alt` attribute for accessibility and missing image.
- `<picture>`: Add `` fallback for unsupported browsers.
- Optimize images for faster webpage loading.



EMOJIS IN HTML



HTML TAGS



EMOJIS IN HTML



Syntax

Directly pasting the emoji

```
<p>I love HTML! 😍</p>
```

Using Unicode

```
<p>I love coding! &#x1F4BB;</p> <!-- This will display the 🖥 emoji -->
```

Using CSS

```
.btn-love::before {  
    content: "♥";  
    padding-right: 5px;  
}
```



EMOJIS IN HTML



Basic Explanation

- Emojis: Cute faces and symbols used in texts and online.
- Add emojis in HTML pages like regular text.



Deep Dive

- Emojis are characters with Unicode codes.
- Unicode ensures consistent display across devices and platforms.
- Browsers render emojis as specified symbols in HTML.



Analogue

- Analogy: Emojis as universal language
- "😊" universally understood as a smiley face
- HTML enables seamless inclusion of emojis on web pages.



When to use?

- Use emojis for flair, emotion.
- Provide visual cues.
- Enhance relatability and engagement in content.



Important Rules

- Use emojis in moderation for professionalism.
- Maintain message clarity with emojis.
- Verify cross-platform compatibility.
- Consider cultural sensitivity in emoji use.



Summary

- Emojis enrich HTML content, provide cues, engage users.
- Use emojis in moderation for effectiveness.



DEMO TIME





HTML

ENTITIES



HTML TAGS



HTML ENTITIES



Basic Explanation

- HTML Entities: Special characters in HTML.
- Reserved or non-ASCII characters.
- Examples: `<`, `>`, `&`, and more.



Analogue

- Analogy: Texting with special emoji codes.
- HTML Entities for special characters not easily typed.



Important Rules

- Close entities with a semicolon.
- Prefer named entities for readability.
- Be aware of browser support variations.



Deep Dive

- HTML Entities format: `&` start, `;` end.
- Needed for characters with special HTML meanings.
- Render symbols like `<`, `>` as text.
- Use for characters not on the keyboard, like ©.



When to use?

- Use HTML Entities:
 1. For reserved HTML characters (<, >, &).
 2. Special characters like ©, ™.
 3. Preserve white spaces in preformatted text.
 4. Safeguard user-generated content against XSS attacks.



Summary

- HTML Entities start with `&` and end with `;`.
- Display special or reserved characters like <, >, &.
- Essential for web design and security.
- Close with a semicolon, prefer named entities for readability.

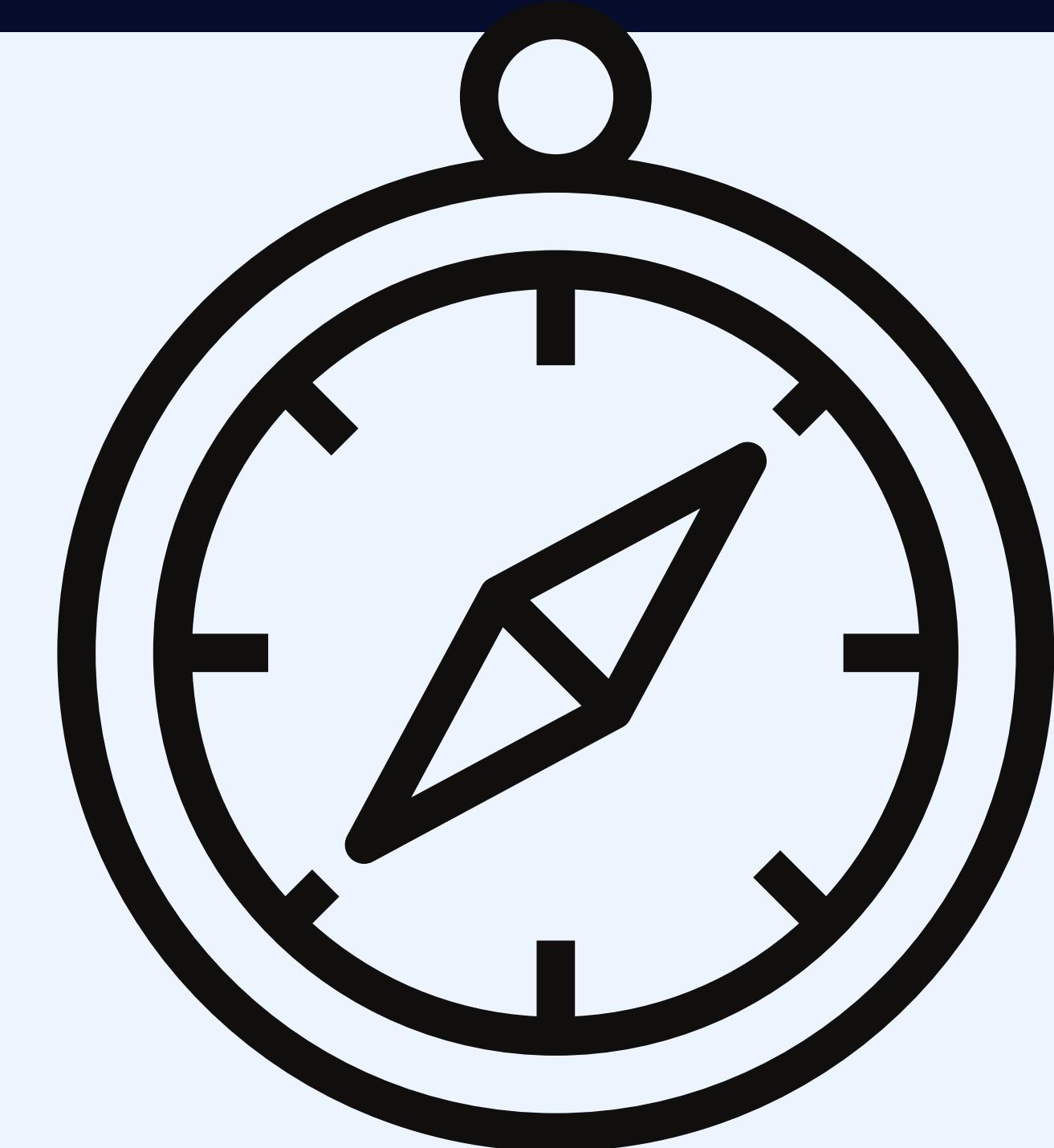


DEMO TIME

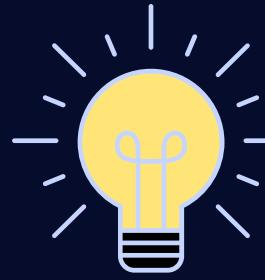




HTML NAVIGATIONS



HTML TAGS



HTML INTERNAL NAVIGATION



Basic Explanation

- HTML Internal Navigation: Navigate within a single webpage.
- Jump between sections without manual scrolling.



Analogue

- Analogy: Table of contents in a long article.
- Clicking items jumps to different sections.
- Internal navigation allows jumping within the article.



Deep Dive

- Internal navigation uses `<a>` tags with named anchors.
- Components: Destination point (identified by `id`) and link (`<a>` with `href` using `#` and `id`).



Important Rules

- Best practices for internal navigation:
 1. Unique `id` attributes.
 2. Meaningful `id` placement.
 3. Descriptive clickable text.
 4. Thorough testing for functionality.



When to use?

- Use internal navigation for:
 1. Lengthy web pages with sections or chapters.
 2. One-page websites with segmented content.
 3. "Back to Top" links for quick navigation to the page's start.



Summary

- HTML Internal Navigation with `<a>` and named anchors.
- Ideal for long webpages or one-page sites.
- Use unique `id` attributes for effective implementation.
- Ensure intuitive navigation for users.



DEMO TIME



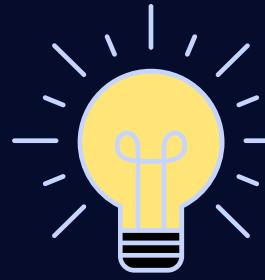


HTML

HYPERLINKING



HTML TAGS



HTML HYPERLINKING



Basic Explanation

- HTML Hyperlinking: Create clickable elements
- Click directs to another page or resource



Analogue

- Analogy: Hyperlinking as adding doors to a building
- Each door (link) leads to a different room (webpage)
- Clicking link opens the door to a new room



Important Rules

- Best practices for hyperlinks:
 1. Use descriptive text, avoid vague phrases.
 2. Open external links in new tabs with `target="_blank"`.
 3. Add `alt` text for image links, use ARIA roles for accessibility.
 4. Distinguish links with color or underlining.
 5. Regularly check and fix broken links.



Deep Dive

- Hyperlinks essential for web navigation and resource access
- HTML `<a>` element used for hyperlinks
- `href` attribute stores URL
- Clickable content within `<a>` tags



When to use?

- Use hyperlinks for:
 1. Webpage navigation
 2. Linking to external resources
 3. Initiating actions (email, call)
 4. Internal navigation within one webpage



Summary

- HTML Hyperlinking: Clickable elements for navigation and resource access
- Anchor (`<a>`) element
- Enhances user experience with clear pathways
- Styling and maintenance for an optimized website



DEMO TIME





DEPLOYING HTML WEBSITES



DEPLOYMENT



DEPLOYING HTML WEBSITE



Basic Explanation

- Deploying HTML website: Making it accessible on the internet
- Transition from private rehearsal to public performance



Analogue

- Analogy: Deploying a website as a theater performance
- Renting a theater (hosting)
- Unique show name (domain)
- Performing the play (uploading files)
- Inviting the audience to watch online



Important Rules

- Key considerations for website deployment:
 1. Select a suitable hosting provider.
 2. Perform regular data backups.
 3. Ensure mobile responsiveness.
 4. Keep software and plugins updated.
 5. Implement HTTPS for security.



Deep Dive

- Steps for deploying an HTML website:
 1. **Web Hosting:** Obtain server space for site files.
 2. **Domain Name:** Web address for visitors.
 3. **Upload Files:** Transfer HTML, CSS, JS via FTP or control panel.
 4. **Testing:** Ensure website functions correctly.



When to use?

- Reasons to deploy a website:
 1. Share personal/business site globally.
 2. Showcase portfolios, blogs, e-commerce.
 3. Present information to a wider audience.



Summary

- Deploying HTML website: Making it live online
- Steps: Hosting, domain, upload, testing
- Best practices for security, accessibility, efficiency

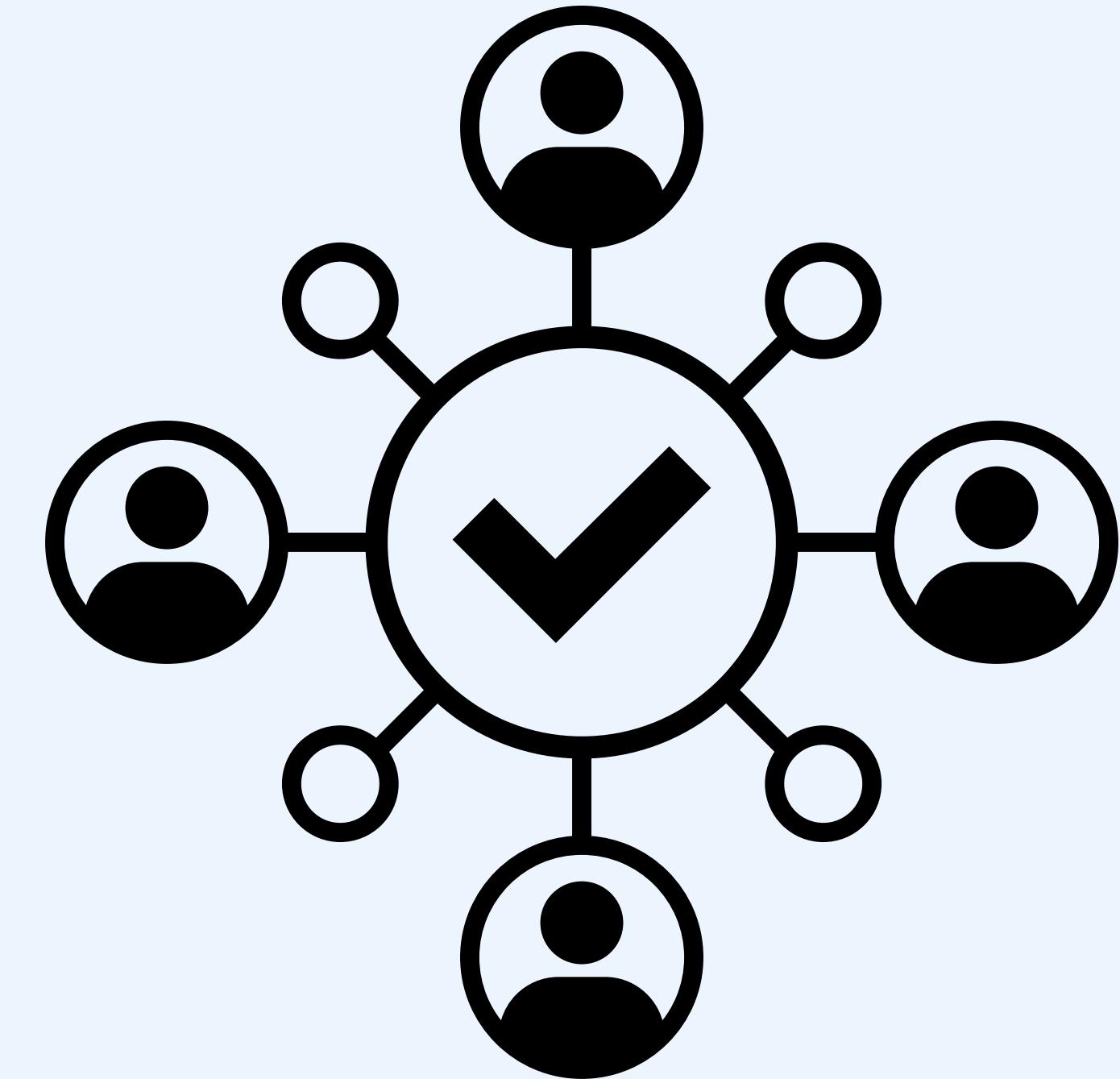


DEMO TIME



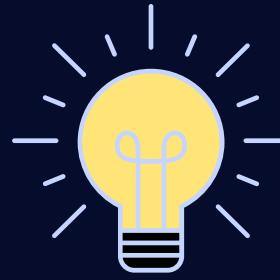


USING FONT ICONS



ICONS

HOW TO USE FONT BASED ICONS (FONTAWESOME)



Require CDN



```
<!-- Include Font Awesome CDN in the head section -->
<link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css" />
```

Require CDN

```
<!-- Use the icon in the body section -->
<i class="fas fa-user"></i>
```

Syntax

- Code for using Font Awesome icons:
 1. `<link rel="stylesheet" href="...">` : Link to Font Awesome stylesheet.
 2. `<i class="fas fa-user"></i>` : HTML tag for displaying icons.
 - `<i>` or `` : Commonly used tags for icons.
 - `class="fas fa-user"` : Specifies style and icon, varies by library.

HOW TO USE FONT BASED ICONS (FONTAWESOME)



Basic Explanation

- Font-based icons: Special fonts displaying icons
- Similar to emojis, versatile
- Add text to HTML, appears as icon



Analogue

- Analogy: Font-based icons like switching font to "Emoji"
- Typing keywords or codes transforms into icons
- Icons like search magnifier or shopping cart



Deep Dive

- Advantages of font-based icons over image-based icons:
 - Scalable (vector-based, no quality loss)
 - CSS styling for customization (color, size, shadow)
 - Libraries like Font Awesome, Material Icons, Ionicons offer diverse icons as font files.



When to use?

- When to use font-based icons:
 1. Ensure visual consistency on various platforms.
 2. Customize icons in terms of color, size, style.
 3. Reduce HTTP requests for performance with multiple icons.



Important Rules

- Font-based icon best practices:
 1. Verify library's license for commercial use.
 2. Provide "alt" text or aria-labels for accessibility.
 3. Use CSS for styling while preserving original design.



Summary

- Font-based icons: Special fonts displaying icons
- Easy resizing and styling
- Enhance UI consistency and interactivity
- Check license and ensure accessibility



DEMO TIME





USING SVG



SVG

HOW TO USE SVG (SCALABLE VECTOR GRAPHIC)



Basic Explanation

- SVG: Scalable Vector Graphics format
- Used for vector images
- Allows unlimited zoom without pixelation
- Contrast with raster images (JPEG, PNG), which pixelate when enlarged.



Deep Dive

- SVG: XML-based vector image format
- Describes 2D graphics and apps in XML
- Editable in text editor due to XML structure
- Maintains high quality at any size
- Supports interactivity and animation for dynamic experiences.



Analogue

- Analogy: SVG as LEGO of web graphics
- XML tags as LEGO pieces with roles
- Build from simple shapes to complex illustrations
- Scalable without quality loss, like LEGO models.



Important Rules

- SVG best practices:
 1. Use for simple shapes/icons for scalability.
 2. PNG or JPEG for complex/multi-color images.
 3. Ensure accessibility with `alt` attributes.
 4. Compress SVG files for size and performance.



Summary

- SVGs: High-quality digital LEGO graphics
- Scalable, animate, interact
- Ideal for crisp, resizable icons, logos, illustrations
- Ensure accessibility, use judiciously based on image complexity.

HOW TO USE SVG (SCALABLE VECTOR GRAPHIC)



Require CDN



```

```

Using the `object` Tag

```
<object type="image/svg+xml" data="image.svg">Your browser does not support SVG</object>
```

Inline SVG

```
<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="black" stroke-width="3" fill="red" />
</svg>
```

- Inline SVG example:
- `<svg>`: SVG container.
- `<circle>`: Draws a circle.
- `cx` and `cy`: Define center coordinates.
- `r`: Radius.
- `stroke`, `stroke-width`, `fill`: Define color, stroke width, and fill color of the circle.



DEMO TIME





HTML FAVICONS





FAVICONS IN HTML



Basic Explanation

- Favicon: Small branded icon in browser tabs
- Enhances website recognition in open tabs



Deep Dive

- FAVICONS: Typically 16x16 or 32x32 pixel square images
- Formats: .ico, .png, .jpg, .svg
- Used in bookmarks, brand visibility
- Modern browsers support multiple sizes for home screen shortcuts, app icons.



Analogue

- Analogy: Favicon as a website's profile picture in tabs
- Helps identify tabs when multiple are open



When to use?

- Use cases for favicons:
 1. Professional websites for brand recognition.
 2. E-commerce sites for visibility in multiple tabs.
 3. Web apps for an application-like experience in browsers.



Important Rules

- Favicon best practices:
 1. Include for professionalism and UX.
 2. Use .ico (universal) or modern formats (PNG, JPG, SVG).
 3. Use square images to prevent distortion.
 4. Keep it simple and recognizable at small sizes.



Summary

- Favicon: Tiny branded icon in browser tabs
- Extends brand presence onto user's browser and bookmarks list



FAVICONS IN HTML



Syntax

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

<!-- For modern browsers supporting PNG -->
<link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png">

<!-- For iOS devices -->
<link rel="apple-touch-icon" sizes="180x180" href="/apple-touch-icon.png">
```



HTML

TABLES



TABLES



HTML TABLE OVERVIEW



Basic Explanation

- HTML tables: Organize and display info in grid format
- Similar to spreadsheet with rows and columns



Deep Dive

- HTML table tags:
 - `<table>`: Begins and ends the table.
 - `<tr>`: Defines table rows.
 - `<th>`: Table headers (bold, centered).
 - `<td>`: Table data cells (regular text, left-aligned by default).



Analogue

- Analogy: HTML tables like organizing a semester timetable
- Organize data on web pages in rows and columns



When to use?

- Use cases for HTML tables:
 1. Tabular data (scores, timetables, product features).
 2. Data comparisons.
 3. Content layout (though CSS Grid and Flexbox are modern alternatives).



Important Rules

- HTML table best practices:
 1. Use for tabular data, not layout.
 2. `<thead>`, `<tbody>`, `<tfoot>` for structure.
 3. `<caption>` for title/summary.
 4. `<colgroup>` and `<col>` for complex tables and styles.



Summary

- HTML tables: Display structured data in rows and columns
- Use for tabular data, not page layout



DEMO TIME





HTML TABLE- STRUCTURE



TABLES



HTML TABLE STRUCTURE



Basic Explanation

- HTML table structure: Grid with rows and columns
- Rows with header cells (` `) and standard cells (` `) | |



Analogue

- HTML tables: Grid like setting for organizing data
- ` `: Labels for rows and columns |
- ` `: Cells containing data |



Important Rules

1. Semantic structure with ``, ``, `` for accessibility.
2. Tables for data, not layout; rely on CSS for layout.
3. `scope` attribute in ` ` to clarify header's relation. |
4. `` for table summary/title, improving accessibility.



Deep Dive

- Key HTML table elements:
 - `<table>`: Table container.
 - `<tr>`: Table row.
 - `<th>`: Table header cell.
 - `<td>`: Table data cell.
- **HTML5 introduced:**
 - `<thead>`: Header rows container.
 - `<tbody>`: Body content container.
 - `<tfoot>`: Footer rows container.



When to use?

- Displaying data in a structured format
- Creating charts and comparisons
- Showing schedules or timetables
- Presenting simple database outputs



Summary

- HTML table: Grid for data
- Rows: `| | |
| --- | --- |
|` with ` ` (header) or ` ` (data) cells | |
- Structure with ``, ``, `` for readability and accessibility.



NESTED HTML TABLE



TABLES



DEMO TIME



HTML FORMS OVERVIEW



TABLES



HTML FORMS OVERVIEW



Basic Explanation

- **HTML forms:** Collect user data
- Vary from simple search to complex registration



Analogue

- Analogy: HTML form as an online survey/questionnaire
- Various question-answer formats:
 - Multiple-choice (radio buttons)
 - Fill-in-the-blank (text fields)
 - Selection from a list (dropdowns)



Important Rules

1. Associate `<label>` with form controls for accessibility.
2. Use `required` for mandatory fields.
3. Validate data both client-side (user) and server-side (security).
4. Utilize HTML5 input types (e.g., `email`, `number`) when fitting.
5. Select `method` attribute (`GET` or `POST`) based on data nature.



Deep Dive

- HTML forms structure input controls.
- Use elements like `<form>`, `<input>`, `<textarea>`, `<select>`, `<button>`.
- `<form>`: Container for form controls.
- `<input>`: Versatile for various input types.
- `<textarea>`: Multiline text input.
- `<select>`: Dropdown list.
- `<button>`: Clickable, can submit form.



When to use?

- User registration and login pages
- Feedback or contact forms
- Search bars
- Surveys and questionnaires
- E-commerce checkout forms



Summary

- HTML forms collect structured user input.
- Include text boxes, checkboxes, radio buttons, etc.
- Vital for dynamic, interactive websites.
- Enable user registration, search, data submission.



HTML FORMS STRUCTURE



TABLES



HTML FORMS STRUCTURE



Basic Explanation

- HTML Forms structure: Arrangement of form elements
- Inside `<form>` tag: Text fields, buttons, checkboxes, etc.



Analogue

- Analogous scenario: Job application form
- HTML structure enables actions:
 - Typing name (text field)
 - Selecting job position (dropdown)
 - Uploading resume (input control)
- All within `<form>` element



Important Rules

1. Wrap form controls in `<form>` element.
2. Associate text with controls using `<label>` for accessibility.
3. Group related elements with `<fieldset>` and `<legend>`.
4. Use `name` and `id` attributes for unique identification, especially for server-side processing.



Deep Dive

- `<form>`: Container for form elements
- Input controls within `<form>`:
 - **Text fields:** `<input type="text">`
 - **Radio buttons:** `<input type="radio">`
 - **Checkboxes:** `<input type="checkbox">`
 - **Dropdown lists:** `<select>` and `<option>`
 - **Buttons:** `<button>` or `<input type="button">`
- `<label>`: Describes form controls
- `<fieldset>` and `<legend>`: Group related form controls



When to use?

- Creating sign-up or login pages
- Building complex, multi-step forms
- Implementing search functionality
- Designing interactive quizzes or surveys



Summary

- HTML form: Comprised of `<form>` element
- Container for form controls like text fields, radio buttons, checkboxes
- Proper structure essential for effective and accessible forms



DEMO TIME



HTML 5 OVERVIEW





HTML 5 OVERVIEW



Basic Explanation

- HTML5: Latest version of HTML
- Introduces new features and elements
- Enhances interactivity, accessibility, and development ease for websites.



Analogue

- Analogy: HTML5 like a new smartphone model
- Retains old functions, adds new features
- New tags and capabilities enhance website functionality



Important Rules

1. Prioritize semantic elements for readability and SEO.
2. Enhance accessibility with ARIA roles and attributes.
3. Prefer native HTML5 features over third-party plugins for performance and security.
4. Implement progressive enhancement: start with basics, add advanced features for capable browsers.
5. Validate HTML5 code for compliance with standards.



Deep Dive

- HTML5 major upgrade over HTML 4.01, XHTML
- Introduces semantic tags: `<header>`, `<footer>`, `<article>`, `<section>`
- Enhances content organization, SEO
- Advanced multimedia: `<audio>`, `<video>` (replaces Flash)
- Offline web apps, real-time communication, graphics (Canvas, SVG)
- Mobile-friendly, cross-platform compatibility



When to use?

- Building responsive websites for desktop and mobile.
- Creating multimedia-rich sites without third-party plugins.
- Developing offline-capable web apps.
- Implementing real-time applications like chat apps.



Summary

- HTML5: Latest and advanced HTML version
- Introduces new features and elements
- Enables interactive, accessible, flexible websites
- Offers semantic elements, multimedia capabilities, and more for modern web development.



HTML5 COMMON ELEMENTS





HTML 5 COMMON ELEMENTS



Basic Explanation

- HTML5 common elements: Building blocks for webpages
- Include headings, paragraphs, links, images, lists, and more.



Analogue

- Analogy: HTML5 elements as magazine sections
- `<h1>` for headlines
- `<p>` for main article text
- `` for images
- `<aside>` for sidebars
- New HTML5 elements provide content context,



Important Rules

1. Prioritize semantic tags for readability, SEO.
2. Prefer semantics over default `<div>` (e.g., `<section>`, `<article>`).
3. Provide alt text for multimedia accessibility.
4. Maintain proper element nesting, hierarchy.
5. Use HTML5 form validation for enhanced UX (e.g., `required`, `min`, `max`).



Deep Dive

HTML5 introduces semantic elements to enhance content structure:

- `<header>`: For document or section headers
- `<footer>`: Document or section footers
- `<nav>`: Navigation link container
- `<article>`: Self-contained content
- `<section>`: Thematic content grouping
- `<aside>`: Tangentially related content
- `<figure>` and `<figcaption>`: Images with captions

Simplify and improve webpage structure.



When to use?

- Use HTML5 semantic elements for:
 - Enhanced UX and SEO
 - Improved accessibility
 - Responsive layouts
 - Multimedia-rich content



Summary

- HTML5 introduces semantic elements for better webpage structure.
- Enhance developer understanding, search engine parsing.
- Improve usability, accessibility, and SEO.



DEMO TIME

