

Frontend



WELCOME

TechPro

[Panagiotis Karanikolaos]

[Fundamentals of Web
Development]

[05/11/2024]

TechPro

Meet Your Instructor











| Nice to Meet You! 🤗

Panagiotis Karankolaos

- **Professional Identity:**
 - Full-Stack Developer.
 - Backend development using Java and Spring Framework.
 - Frontend development with JavaScript.
- **Experience:**
 - Full-stack developer with 5 years of experience in the tech industry
 - Previous experience as a mathematics teacher in the private sector
- **Educational Background:**
 - Bachelor's degree in Mathematics
 - Master's degree in Information Systems
 - Transitioned from education to tech, leveraging strong analytical and problem-solving skills



| Before We Start...

-  Practice Makes Perfect: Watching alone won't help. Dive in and code!
-  Stay Positive: Feeling lost at first is normal. Don't doubt your coding potential!
-  Ask Away: No question is 'stupid.' Every query is a step towards learning.
-  Set Goals: Small, achievable goals can lead to big successes. Celebrate every win!
-  Embrace Challenges: Difficult topics enhance your problem-solving skills. Tackle them head-on!
-  Collaboration: Working with others can illuminate new perspectives and solutions.
-  Patterns over Syntax: Understanding patterns is key to mastering programming concepts.
-  Stay Healthy: Taking care of your physical and mental health is essential for sustained success.
-  Dopamine Hits: Break tasks into small chunks to achieve quick wins that boost your motivation.
-  Not Too Hard Though: Keep the challenge level balanced to stay engaged without burning out.

| Learning Approach

Theory & Hands-On

Engage with both conceptual explanations and practical exercises to solidify understanding and skills.

Interactive Sessions

Participate in Q&A sessions to clarify doubts and deepen knowledge.

Quizzes

Test understanding of key concepts regularly.

Hands-On Assignments

Apply learned skills to solve real-world problems.

Code Reviews

Participate in peer and instructor reviews to refine coding practices.

Break Time

Take a short break of 10-15 minutes during sessions to refresh and maintain focus.



Rules for Engagement

Homework:

- **Complete assignments**
- **Come Prepared**

Classroom Rules:

- **Be Polite**
- **Respectful Interaction**
- **Considerate Behavior**
- **Active Participation**

| Availability & Support

Reach out:

- pnskaranik@gmail.com



Agenda

01 – Basic Structure of an HTML

02 – Common HTML Tags

03 – HTML Attributes And Elements

Basic Structure of an HTML

| What does a frontend developer do?

What is Frontend Development?

Frontend development refers to the part of a website or application that the user sees and interacts with. It's the "face" of the application or website.

What Does a Frontend Developer Do?

A frontend developer designs and builds the visual part (what we see) and how it functions.

- Designing the User Interface (UI): Creating buttons, menus, images, and everything the user sees and uses.
- Ensuring Interactivity (UX - User Experience): How the buttons and menus react when the user interacts with them (e.g., when you click a button and a window opens).

| HTML, CSS and (little) JavaScript

HTML: HTML (HyperText Markup Language) is the fundamental language used for creating websites. It is the "skeleton" of the website, defining its structure and content. Think of HTML like the building materials of a house: the bricks, walls, and windows.

CSS: CSS (Cascading Style Sheets) is added to make the website visually appealing. It's like the decoration of the house: the colors, furniture, and how all the elements are arranged.

JavaScript: JavaScript is used to add interactivity to a website. While HTML is the "skeleton" and CSS is the "decoration," JavaScript is like the electrical system of a house: it brings the website to life, making things move, respond to clicks, and interact with the user.

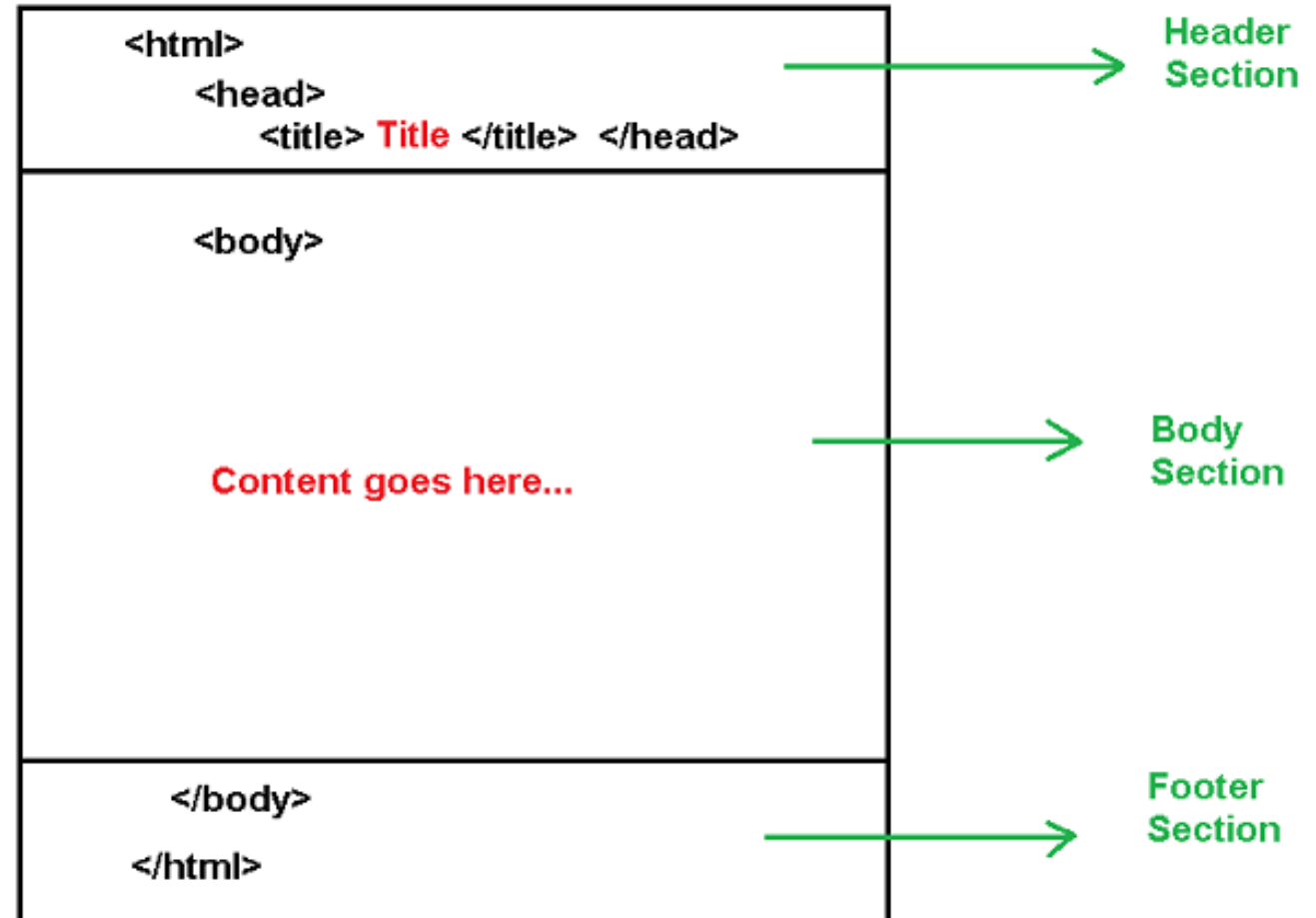


| Basic Structure of an HTML Document

HTML is like the foundation of a house:

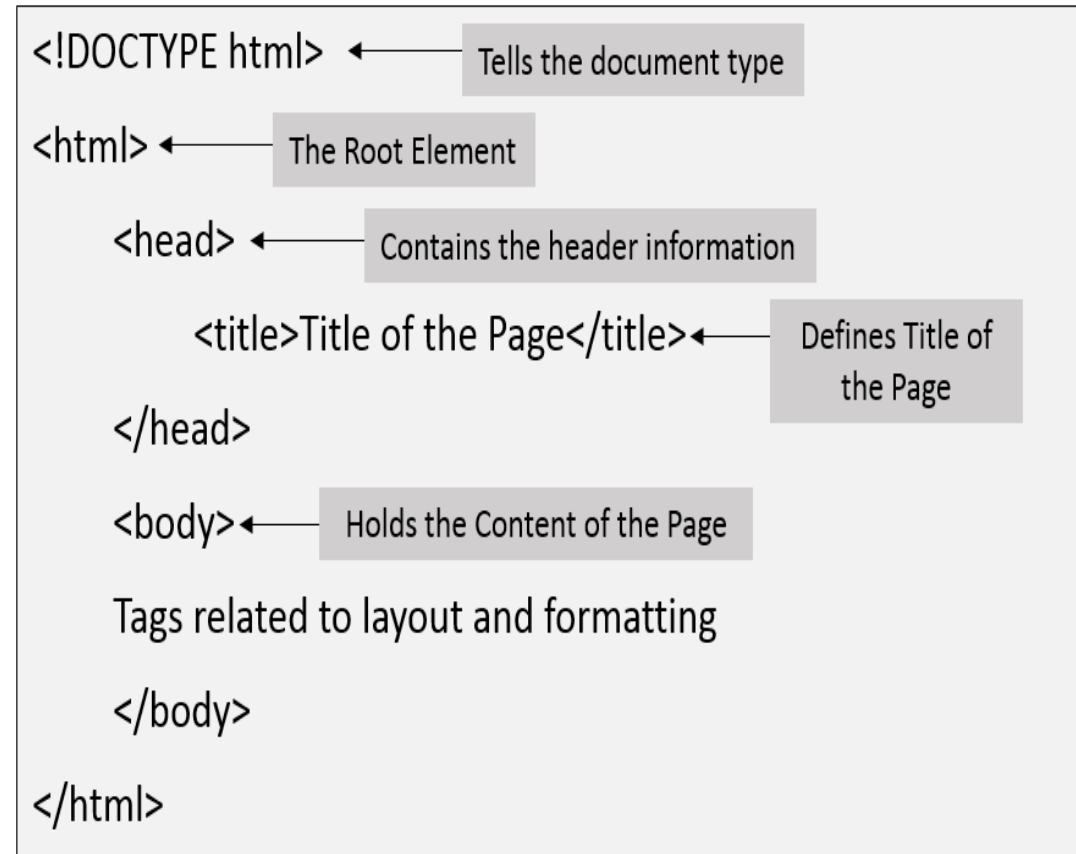
It organizes the content of the website.

Every HTML document starts and ends with the `<html>` tag, which tells the browser that this is an HTML document.



| HTML Structure

- **<!DOCTYPE html>**: Declares to the browser that HTML5 is being used.
- **<html>**: The main element that contains all other elements of the webpage.
- **<head>**: The section that contains metadata about the webpage, such as the title and links to CSS files or scripts.
- **<title>**: The title that appears in the browser tab.
- **<body>**: The visible content of the webpage is displayed here.



Common HTML Tags

| What is an HTML tag

HTML tags are like labels used to mark up the content of a web page. They are enclosed in angle brackets (< >).

Key Points:

- Most tags come in pairs: an opening tag and a closing tag.
- The opening tag indicates where the content starts, while the closing tag shows where it ends. The closing tag includes a forward slash (/) before the tag name.

Example

```
<p>This is a paragraph.</p>
```

Self-Closing Tags

- Some tags do not have a closing tag and are called self-closing tags.
- These are used for elements that do not contain any content between them.

Example

```

```



| List of HTML Tags

- <!--...-->
- <!DOCTYPE>
- <a>
- <abbr>
- <acronym>
- <address>
- <applet>
- <area>
- <article>
- <aside>
- <audio>
-
- <base>
- <basefont>
- <bdi>
- <bdo>
- <big>
- <blockquote>
- <body>
-

- <button>
- <canvas>
- <caption>
- <center>
- <cite>
- <code>
- <col>
- <colgroup>
- <data>
- <datalist>
- <dd>
-
- <details>
- <dfn>
- <dialog>
- <dir>
- <div>
- <dl>
- <dt>
-
- <embed>
- <fieldset>
- <figcaption>
- <figure>
-
- <footer>
- <form>
- <frame>
- <frameset>
- <h1> to <h6>
- <head>
- <header>
- <hr>
- <html>
- <i>
- <iframe>
-
- <input>
- <ins>
- <kbd>
- <label>
- <legend>
-
- <link>
- <main>
- <map>
- <mark>
- <meta>
- <meter>
- <nav>
- <noframes>
- <noscript>
- <object>
-
- <optgroup>
- <option>
- <output>
- <p>
- <param>
- <picture>
- <pre>
- <progress>
- <q>
- <rp>
- <rt>
- <ruby>
- <s>
- <samp>
- <script>
- <section>
- <select>
- <small>
- <source>
-
- <strike>
-
- <style>
- <sub>
- <summary>
- <sup>
- <svg>
- <table>
- <tbody>
- <td>
- <template>
- <textarea>
- <tfoot>
- <th>
- <thead>
- <time>
- <title>
- <tr>
- <track>
- <tt>
- <u>
-
- <var>
- <video>
- <wbr>

● Not supported in HTML5



| Header Tag

- Helps structure content with titles and subtitles.
- Improves SEO (Search Engine Optimization)



```
1 <h1>H1 title</h1>
2 <h2>H2 title</h2>
3 <h3>H3 title</h3>
4 <h4>H4 title</h4>
5 <h5>H5 title</h5>
6 <h6>H6 title</h6>
```

H1 title

H2 title

H3 title

H4 title

H5 title

H6 title



| Paragraph Tag

- Used to group related text into logical sections.
- Helps separate content, making it more readable and organized.



```
1 <p>This is a paragraph of text.</p>  
2 <p>Another paragraph goes here.</p>
```

HTML Paragraph Example

This is a paragraph of text.

Another paragraph goes here.



| Hyperlink Tag

- Used to create links to other pages or sections of content.
- The href attribute specifies the URL of the link.



```
1 <a href="https://www.google.com/">Κάντε κλικ εδώ για να μεταφερθείτε στο Google</a>
```

HTML Link Tag Example

[Κάντε κλικ εδώ για να μεταφερθείτε στο Google](https://www.google.com/)



| Image Tag

The tag is used to embed an image in an HTML page.

- src - Specifies the path to the image
- alt - Specifies an alternate text for the image, if the image for some reason cannot be displayed



```
1 
```



| Lists Tag

- `` (unordered list): Creates a list with bullet points.
- `` (ordered list): Creates a numbered list.
- ``: Used for each list item, regardless of the list type.

```
1  <ul>
2      <li>First bullet point</li>
3      <li>Second bullet point</li>
4  </ul>
5
6  <ol>
7      <li>First numbered item</li>
8      <li>Second numbered item</li>
9  </ol>
```

- First bullet point
 - Second bullet point
-
1. First numbered item
 2. Second numbered item

| Div Tag

The **<div>** tag is a general block-level element used to group other HTML elements into a section, typically for layout and formatting purposes.

The **<div>** itself has no built-in styling but is often used with CSS to define the layout of a page (such as creating a container).



```
1  <div>
2      <h2>Section 1</h2>
3      <p>This is the first section inside the container.</p>
4  </div>
```

HTML Attributes And Elements

| What is an HTML Element?

- An HTML element is a building block of a webpage.
- It consists of an opening tag, content, and a closing tag (unless it's self-closing)
- Elements define how content is structured and presented on the webpage.

HTML Element Example

`<p>This is a paragraph inside an HTML element.</p>`

Think of an HTML element as a container in everyday life, like a box that holds specific items.

The box itself is the element, the label on the box is the tag, and the items inside are the content.

- Opening tag: It's like opening the box with a label that says what's inside.
- Content: What you find inside the box (e.g., text, images, or other boxes/elements).
- Closing tag: Seals the box when you're done.




| Nesting HTML Elements

Nesting occurs when one HTML element is placed inside another. This is crucial for creating more complex web structures.

Think of nesting like boxes within boxes. You can have smaller boxes (elements) inside larger ones, creating a hierarchy.

Nesting HTML Elements Example



```
1 <div>
2     <h1>This is a heading inside a div</h1>
3     <p>This is a paragraph inside the same div.</p>
4 </div>
```

| Block-level vs. Inline Elements

HTML elements can either be block-level or inline, determining how they are displayed on the webpage

- Block-level elements: Take up the full width of their container and always start on a new line.
- Inline elements: Only take up as much space as needed and appear within the same line as surrounding elements.

Example

`<p>This is a highlighted word inside a paragraph.</p>`

- The `<p>` is a block-level element.
- The `` is inline, highlighting part of the text.

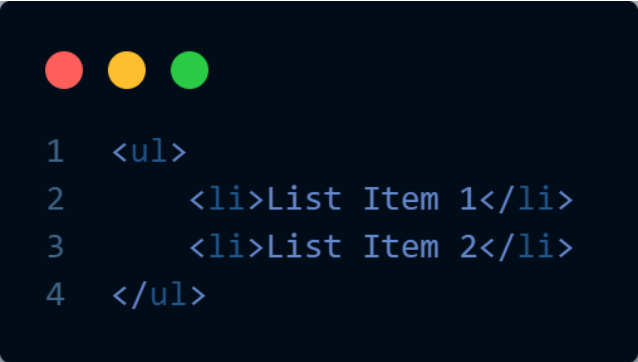
| Parent and Child Elements

HTML elements can be in a parent-child relationship when one element contains another. The parent element wraps around one or more child elements.

Think of a parent-child relationship like a folder containing files. The folder (parent) can have one or more files (children) inside it.

Parent-child relationships help structure the layout of nested content in HTML. Here, the `` (unordered list) is the parent element, and each `` (list item) is a child element.

Example



```
1 <ul>
2   <li>List Item 1</li>
3   <li>List Item 2</li>
4 </ul>
```

Here, the `` (unordered list) is the parent element, and each `` (list item) is a child element.

| What is an HTML Attribute?

- An attribute is extra information added to an HTML element to control its behavior or describe its content.
- Attributes modify HTML elements, giving them specific characteristics or additional properties.
- Placed inside the opening tag of an element in the format:

Example

```
<p attribute-name="attribute-value">Some Paragraph Content</p>
```

Think of Attributes Like Instructions on a Box Just like a label on a box might say "Fragile" or "Handle with Care," HTML attributes give extra instructions for elements.

For instance, alt="A description" on an image is like a label describing what the image is about.



| Why Do We Use Attributes?

Add Functionality

Attributes let elements perform special actions.

For example, the href attribute in `Click here` makes a clickable link that takes users to another page. Similarly, src in `` tells the browser where to find an image file.

Provide Context and Accessibility

Attributes help make content more accessible, especially for visually impaired users.

For example the alt attribute on images provides a text description that screen readers use to convey image content to users who cannot see the images.

```

```

Enable Styling and Scripting

Attributes like class and id are crucial for styling with CSS and adding interactivity with JavaScript.

Examples

- `<div class="header">` allows CSS to style all elements with the class header.
- `<button id="submitBtn">Submit</button>` lets JavaScript target this specific button for an action (like submitting a form).

| The Most Common HTML Attributes

href - Hyperlink Reference

Used with the <a> (anchor) tag to specify the URL a link points to.

Example

```
<a href="https://www.example.com">Visit Example</a>
```

src - Source

Used with , <audio>, and <video> tags to specify the path to a file.

Example

```

```

id - Unique Identifier

Assigns a unique name to an element, allowing it to be targeted with CSS or JavaScript.

Example

```
<div id="main-header">Welcome to My Website</div>
```

class - Class Name

Groups elements for CSS styling or JavaScript interactions. Multiple elements can share the same class.

Example

```
<p class="highlight">This text is highlighted.</p>
```





Feedback Discussion



Thank you