SYSTEM TRON INTERNSHIP NOTES
WEB DEVELOPMENT PART 1
HTML

Chapter 1: Introduction to HTML

1. What is HTML?

2. How does the web work?

3. HTML's role in web development

4. Basic structure of an HTML document

Chapter 1: Introduction to HTML

HTML stands for "Hypertext Markup Language." It's the language that makes up most of the websites you see on the internet. In this chapter, we'll explore what HTML is, how the web works, HTML's role in web development, and the basic structure of an HTML document. Let's break it down in easy language with examples:

1. What is HTML?

Think of HTML as the building blocks of a website. It's a set of instructions that web browsers (like Chrome, Firefox, or Safari) follow to display web pages. HTML uses a system of tags, which are like labels that tell the browser how to show content.

Example:

```html
<!DOCTYPE html>
<html>
  <head>
    <title>My First Web Page</title>
  </head>
  <body>
```

```
    <h1>Hello, World!</h1>
    <p>This is a basic HTML page.</p>
  </body>
</html>
```

Here, we have simple HTML tags like `<html>`, `<head>`, `<title>`, `<body>`, `<h1>`, and `<p>`.

2. How does the web work?

The web is like a massive library, and each website is a book. When you type a web address (URL) into your browser and hit Enter, your browser sends a request to a web server, which is like the librarian. The server finds the right book (web page), and the browser displays it using HTML.

3. HTML's role in web development

HTML is the backbone of web development. It structures the content of a web page. But websites are more than just text; they have pictures, links, and interactive features. To make websites look good and do cool things, we also use other technologies like CSS (Cascading Style Sheets) for styling and JavaScript for interactivity.

4. Basic structure of an HTML document

An HTML document has a specific structure. It starts with a declaration (`<!DOCTYPE html>`) and contains the following parts:
- `<html>`: The main container for everything.
- `<head>`: Contains metadata, like the title of the page.
- `<title>`: Sets the title shown in the browser tab.
- `<body>`: Holds the visible content of the web page.

In the "Example" above, you can see this structure in action.

As you learn more about HTML, you'll discover various tags and attributes to create links, images, tables, forms, and more. These tags and attributes allow you to build complex and interactive web pages.

This chapter is just the beginning. Understanding HTML is the first step to becoming a web developer and creating your own websites.

Chapter 2: HTML Document Structure

1. HTML Document Structure

2. The `<!DOCTYPE>` declaration

3. The `<html>` element
4. The `<head>` element
5. The `<title>` element
6. The `<meta>` element
7. The `<link>` element
8. The `<style>` element
9. The `<script>` element
10. The `<body>` element

Chapter 2: HTML Document Structure

In this chapter, we will learn about the basic structure of an HTML document. We'll break it down into several parts, and I'll explain each part using easy-to-understand language and examples.

HTML Document Structure:
An HTML document is like a recipe for a web page. It tells web browsers how to display your content. Every HTML document has a specific structure.

The `<!DOCTYPE>` Declaration:
This is the first thing you write in your HTML document. It tells the browser which version of HTML you're using. It doesn't have any visible content; it's just a declaration. Here's an example:

```html
<!DOCTYPE html>
```

The `<html>` Element:
This is the main container of your entire web page. It includes everything on the page, like text, images, and links. It has two parts, the `<head>` and the `<body>`.

```html
<html>
  <!-- This is where the head and body go -->
</html>
```

The `<head>` Element:
The head contains information about your web page, like its title and other settings. Users don't see what's inside the head; it's for the browser and search engines.

```html
<head>
  <title>My Awesome Website</title>
  <meta charset="UTF-8">
  <!-- Other things like styles and scripts go here -->
</head>
```

The `<title>` Element:
The title element sets the title of your web page, which appears in the browser's tab. It's also used in search results. For example:

```html
<title>My Awesome Website</title>
```

The `<meta>` Element:
The meta element provides information about your page, like the character encoding (how text is displayed). The charset attribute ensures your text displays correctly.

```html
<meta charset="UTF-8">
```

The `<link>` Element:
This is used to link your HTML document to external resources, like stylesheets (CSS) or icons. It's commonly used in the head.

```html
<link rel="stylesheet" type="text/css" href="styles.css">
```

The `<style>` Element:
You can also include your styles (CSS) directly in the HTML document using the style element. It goes in the head.

```html
<style>
  body {
    background-color: lightblue;
  }
</style>
```

The `<script>` Element:
The script element is used to include JavaScript code in your web page. JavaScript makes your page interactive. It's often placed at the end of the body to ensure the page loads faster.

```html
<script>
  function sayHello() {
    alert("Hello, world!");
  }
</script>
```

The `<body>` Element:
This is where you put the visible content of your web page, like text, images, and links. Users see everything inside the body.

```html
<body>
  <h1>Welcome to My Website</h1>
  <p>This is some text on the page.</p>
  <!-- Other content goes here -->
</body>
```

Understanding these basic elements is the foundation of creating any web page in HTML. It's like building blocks where you add the content and structure to make your website look and work the way you want.

Chapter 3: HTML Elements

1. Basic text formatting (headings, paragraphs)
2. Line breaks and horizontal rules
3. Lists (ordered and unordered)
4. Links (`<a>` element)
5. Images (`<img>` element)
6. Comments (`<!-- -->`)

Certainly! Here's Chapter 3: HTML Elements, explained in easy language with examples:

3.1 Basic Text Formatting: Headings and Paragraphs

- HTML lets you structure text. The main ways are headings and paragraphs.
- Headings are like titles for sections. They range from `<h1>` to `<h6>`, with `<h1>` being the biggest.

Example:
```html
<h1>This is a Heading 1</h1>
<h2>This is a Heading 2</h2>
```

- Paragraphs are used for regular text. You enclose text in `<p>` tags.

Example:
```html
<p>This is a paragraph of text. It can be as long as you want.</p>
```

3.2 Line Breaks and Horizontal Rules

- To create a new line without starting a new paragraph, you can use `<br>`.

Example:
```html
<p>This is some text.<br>This is on a new line.</p>
```

- To insert a horizontal line, use `<hr>`.

Example:
```html
<p>Some text before<hr>and some text after the horizontal line.</p>
```

3.3 Lists: Ordered and Unordered

- Lists help organize items. There are two main types: ordered and unordered.
- For ordered lists, use `<ol>`. Each item is `<li>`.

Example:
```html
<ol>
  <li>First item</li>
  <li>Second item</li>
</ol>
```

- For unordered lists, use `<ul>`. Each item is `<li>`.

Example:

```html
<ul>
  <li>Apples</li>
  <li>Bananas</li>
</ul>
```

3.4 Links (`<a>` Element)

- Links are how you navigate to other web pages.
- Use `<a>` with the `href` attribute to specify the URL.

Example:

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

3.5 Images (`<img>` Element)

- To display images, use the `<img>` tag.
- Add the `src` attribute to specify the image file's location.

Example:

```html
<img src="image.jpg" alt="A beautiful image">
```

3.6 Comments (`<!-- -->`)

- Comments are notes for you and others; they don't show on the webpage.
- Wrap comments with `<!--` at the start and `-->` at the end.

Example:

```html
<!-- This is a comment. It won't appear on the page. -->
```

These are fundamental HTML elements for creating content and linking pages. You can use them to build structured and organized web pages.

Chapter 4: HTML Forms

1. Form structure (`<form>` element)
2. Text input fields (`<input type="text">`)
3. Radio buttons and checkboxes
4. Dropdown lists (`<select>` and `<option>`)
5. Text areas (`<textarea>`)
6. Form submission (`<input type="submit">`)

Chapter 4: HTML Forms

HTML forms allow users to input data, and they are an essential part of many websites. In this chapter, we will explore how to create and use HTML forms. Let's break it down into simple language and examples:

Form Structure (`<form>` element):

A form is like a container for various input elements. It's created using the `<form>` element. Here's how you make a basic form:

```html
<form>
  <!-- Input elements go here -->
</form>
```

Text Input Fields (`<input type="text">):

Text input fields are where users can type text. You define them using the `<input>` element with the `type` attribute set to "text". Here's an example:

```html
<form>
  <label for="username">Username:</label>
  <input type="text" id="username" name="username">
</form>
```

In this example, the `label` element provides a label for the input field. The `for` attribute links it to the `id` of the input element for accessibility.

Radio Buttons and Checkboxes:

Radio buttons and checkboxes allow users to choose options. Radio buttons let users select one option from a list, while checkboxes allow multiple selections. Here's an example of radio buttons and checkboxes:

```html
<form>
  <p>Choose your gender:</p>
  <input type="radio" id="male" name="gender" value="male">
  <label for="male">Male</label>
  <input type="radio" id="female" name="gender" value="female">
  <label for="female">Female</label>

  <p>Select your hobbies:</p>
  <input type="checkbox" id="reading" name="hobby" value="reading">
  <label for="reading">Reading</label>
  <input type="checkbox" id="cooking" name="hobby" value="cooking">
  <label for="cooking">Cooking</label>
</form>
```

Dropdown Lists (`<select>` and `<option>`):

Dropdown lists let users choose from a list of options. Use the `<select>` element to create the dropdown and `<option>` elements for each option:

```html
<form>
  <label for="country">Select your country:</label>
  <select id="country" name="country">
    <option value="usa">United States</option>
    <option value="canada">Canada</option>
    <option value="uk">United Kingdom</option>
  </select>
</form>
```

Text Areas (`<textarea>`):

Text areas are for longer text inputs like comments or messages. Here's an example:

```html
<form>
  <label for="comments">Comments:</label>
  <textarea id="comments" name="comments" rows="4" cols="50"></textarea>
</form>
```

Form Submission (`<input type="submit">`):

The submit button allows users to send the form data to a server. Here's how you add it:

```html
<form>
  <!-- Other input elements -->

  <input type="submit" value="Submit Form">
</form>
```

When the user clicks the "Submit Form" button, the data from the form is sent to a server for processing.

That's the basic structure of HTML forms, including various input elements. You can customize them further with attributes and CSS styles to make your forms more appealing and user-friendly.

Chapter 5: HTML Tables

1. Creating tables (`<table>`)
2. Table rows (`<tr>`)
3. Table headers (`<th>`)
4. Table data cells (`<td>`)
5. Table captions (`<caption>`)

Chapter 5: HTML Tables - Creating Structured Data

Creating tables (<table>) : HTML tables are used to display structured data. You can create a table using the `<table>` element. Here's an example:

```html
<table>
  <!-- table content goes here -->
</table>
```

Table rows (<tr>) : To organize data into rows, you use the `<tr>` element inside the table. Each row contains data cells (table headers or table data cells). Example:

```html
<table>
  <tr>
    <!-- row 1 data cells -->
  </tr>
  <tr>
    <!-- row 2 data cells -->
  </tr>
</table>
```

Table headers (<th>) : Table headers are used to label columns or rows. You use the `<th>` element inside a row to create a header cell. Example:

```html
<table>
  <tr>
    <th>Header 1</th>
    <th>Header 2</th>
  </tr>
  <tr>
    <td>Data 1</td>
    <td>Data 2</td>
  </tr>
</table>
```

Table data cells (<td>) : Table data cells contain the actual data within the table. You use the `<td>` element inside a row to create a data cell. Example:

```html
<table>
  <tr>
    <th>Header 1</th>
    <th>Header 2</th>
  </tr>
  <tr>
    <td>Data 1</td>
    <td>Data 2</td>
  </tr>
</table>
```

Table captions (<caption>) : A table caption is used to provide a title or description for the table. It's placed just after the opening `<table>` tag. Example:

```html
<table>
  <caption>Monthly Expenses</caption>
  <tr>
    <th>Category</th>
    <th>Amount</th>
  </tr>
  <tr>
    <td>Rent</td>
    <td>$800</td>
  </tr>
</table>
```

In summary, HTML tables are used to organize data into rows and columns. You create tables using `<table>`, organize data into rows with `<tr>`, label columns or rows using `<th>`, and place actual data in cells with `<td>`. You can also provide a title or description for the table using the `<caption>` element. Tables are a fundamental part of displaying structured information on a web page.

Chapter 6: HTML Semantic Elements

1. Semantic HTML and its importance

2. `<header>`, `<nav>`, `<main>`, `<article>`, `<section>`, `<aside>`, `<footer>`

3. When to use semantic elements

Chapter 6: HTML Semantic Elements - Simplified

In this chapter, we'll learn about semantic HTML elements and why they're important. Semantic elements help search engines and assistive technologies understand your web content better. Let's explore some key elements:

1. Semantic HTML and Its Importance :

- Semantic HTML means using tags that carry meaning. They describe the content within the tags, making it clear and understandable.
- Importance: Search engines use these tags to determine what your page is about. They also help screen readers for people with disabilities.

2. <header> :

- Use `<header>` for the top section of your webpage, typically containing your site's title or logo.
- Example:

```html
<header>
  <h1>Welcome to My Website</h1>
</header>
```

3. <nav> :

- Wrap your site's navigation links in the `<nav>` element.
- Example:

```html
<nav>
  <ul>
    <li><a href="/">Home</a></li>
    <li><a href="/about">About</a></li>
    <li><a href="/contact">Contact</a></li>
  </ul>
</nav>
```

4. <main> :

- `<main>` is where the primary content of your webpage goes. You should only have one per page.
- Example:

```html
<main>
  <h1>Welcome to My Blog</h1>
  <p>Check out my latest posts here.</p>
</main>
```

5. <article> :

- Use `<article>` for a self-contained piece of content that can stand alone, like a news article or blog post.
- Example:

```html
<article>
  <h2>How to Bake the Perfect Cake</h2>
  <p>Follow these steps to make a delicious cake.</p>
</article>
```

6. <section> :

- `<section>` is used to group related content together.
- Example:

```html
<section>
  <h2>Our Services</h2>
  <p>Explore what we offer to our customers.</p>
</section>
```

7. <aside> :

- `<aside>` is for content that is tangentially related to the main content.
- Example:

```html
<aside>
  <h3>Related Links</h3>
  <ul>
    <li><a href="/news">Latest News</a></li>
    <li><a href="/events">Upcoming Events</a></li>
  </ul>
</aside>
```

8. <footer> :

- `<footer>` typically contains copyright information, contact details, and other footer-related content.
- Example:

```html
<footer>
  <p>&copy; 2023 MyWebsite. Contact us at info@mywebsite.com</p>
</footer>
```

9. When to Use Semantic Elements :

- Use semantic elements whenever they accurately represent the structure of your content. If you have a heading, use `<header>`. If you have navigation links, use `<nav>`. If it's the main content, use `<main>`, and so on.

By using these elements, you make your webpage more accessible, understandable, and search engine-friendly. It's like speaking a language that both humans and computers can easily understand.

Chapter 7: HTML Links and Multimedia

1. Hyperlinks (internal and external)

2. Linking to email addresses

3. Linking to files (e.g., PDFs)

4. Embedding audio and video

5. Embedding iframes

Chapter 7: HTML Links and Multimedia

In this chapter, we'll learn how to connect web pages and add different kinds of content to our websites. We'll cover:

1. Hyperlinks (internal and external):

Hyperlinks are like magic buttons that can take you from one webpage to another. They can be used to connect pages within your website (internal) or link to other websites (external).

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

2. Linking to email addresses:

You can create links to open email programs with pre-filled email addresses. When someone clicks, it opens their email with the recipient's address already there.

Example:
```html
<a href="mailto:contact@example.com">Email Us</a>
```

3. Linking to files (e.g., PDFs):

You can link to all kinds of files, like PDF documents. When a visitor clicks the link, the file will open or download.

Example:
```html
<a href="example.pdf">Download PDF</a>
```

4. Embedding audio and video:

You can add audio and video files to your webpage so that visitors can play them without leaving the page.

Example for video:
```html
<video width="320" height="240" controls>
  <source src="example.mp4" type="video/mp4">
  Your browser does not support the video tag.
</video>
```

5. Embedding iframes:

An iframe is like a little window on your webpage that can display content from another website, like Google Maps.

Example:
```html
<iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d2530...">
  Your browser does not support iframes.
</iframe>
```

These features help make your website interactive and engaging. Hyperlinks connect information, and multimedia content like audio, video, and iframes bring your web pages to life. It's like adding the finishing touches to your website to make it informative and fun for your visitors.

Chapter 8: HTML Forms and Input Elements

1. Form structure (`<form>`)
2. Text input fields (`<input type="text">`)
3. Radio buttons and checkboxes
4. Dropdown lists (`<select>` and `<option>`)
5. Text areas (`<textarea>`)
6. Form submission (`<input type="submit">`)
7. Form validation
8. Hidden fields

Chapter 8: HTML Forms and Input Elements

HTML forms are a fundamental part of web development as they allow users to input and submit data on websites. In this chapter, we will explore the key elements related to forms in an easy-to-understand way, along with examples.

1. Form Structure (<form>)
  - A form is like a container that holds various input elements. It's created using the `<form>` tag.

Example:

```html
<form>
   <!-- Input elements go here -->
</form>
```

2. Text Input Fields (<input type="text">)

- These fields are used for single-line text input, like names or email addresses.

Example:

```html
<label for="name">Name:</label>
<input type="text" id="name" name="name">
```

3. Radio Buttons and Checkboxes

- Radio buttons are used when you want the user to select only one option. Checkboxes allow multiple selections.

Example:

```html
<input type="radio" id="male" name="gender" value="male">
<label for="male">Male</label>
<input type="radio" id="female" name="gender" value="female">
<label for="female">Female</label>

<input type="checkbox" id="subscribe" name="subscription" value="yes">
<label for="subscribe">Subscribe to newsletter</label>
```

4. Dropdown Lists (<select> and <option>)

- Dropdown lists provide a list of options for the user to choose from.

Example:

```html
<label for="country">Country:</label>
<select id="country" name="country">
   <option value="usa">United States</option>
   <option value="canada">Canada</option>
   <option value="uk">United Kingdom</option>
</select>
```

5. Text Areas (<textarea>)
- Text areas allow users to input multiple lines of text, like comments or messages.

Example:
```html
<label for="comments">Comments:</label>
<textarea id="comments" name="comments" rows="4" cols="50"></textarea>
```

6. Form Submission (<input type="submit">)
- The submit button lets users send their input data to the server for processing.

Example:
```html
<input type="submit" value="Submit">
```

7. Form Validation
- You can add validation rules to ensure that the data entered is correct. For example, requiring an email format in an email field.

8. Hidden Fields
- Hidden fields are not visible to the user but can store information that is sent along with the form when it's submitted. They are often used to store session data or other hidden values.

Example:
```html
<input type="hidden" name="session_id" value="12345">
```

Understanding and using these form elements is essential for building interactive and user-friendly websites. With forms, you can collect information from users and process it, making websites more dynamic and useful.

Chapter 9: HTML Lists and CSS Styling

1. Ordered and unordered lists
2. Nested lists
3. Basic CSS styling
4. Inline CSS vs. External CSS
5. CSS classes and IDs

Chapter 9: HTML Lists and CSS Styling

In this chapter, we'll learn about creating lists in HTML and how to make them look nice with CSS (Cascading Style Sheets). We'll explore different types of lists, add some style to them, and understand the difference between inline CSS and external CSS. We'll also discover how to use CSS classes and IDs for more specific styling.

Ordered and Unordered Lists:

- Ordered lists are used when you want to list items in a specific order, such as step-by-step instructions. You create them using the `<ol>` tag. Here's an example:

```html
<ol>
  <li>First item</li>
  <li>Second item</li>
  <li>Third item</li>
</ol>
```

- Unordered lists are for items that don't have a particular order. You create them with the `<ul>` tag. Here's an example:

```html
<ul>
  <li>Apples</li>
  <li>Bananas</li>
  <li>Oranges</li>
</ul>
```

Nested Lists:

You can also nest lists inside other lists. For instance, you can have an ordered list within an unordered list or vice versa. Here's a simple example:

```html
<ul>
  <li>Fruits
    <ol>
      <li>Apples</li>
      <li>Bananas</li>
    </ol>
  </li>
  <li>Vegetables
    <ul>
      <li>Carrots</li>
      <li>Broccoli</li>
    </ul>
  </li>
</ul>
```

Basic CSS Styling:

You can make your lists look more appealing by applying some basic CSS styles. For example, you can change the font, color, or spacing of your list items.

```html
<style>
  ul {
    list-style-type: square;
    color: blue;
  }
</style>
```

Inline CSS vs. External CSS:

- Inline CSS is applied directly to the HTML elements. For instance:

```html
<p style="color: red;">This is a red paragraph.</p>
```

- External CSS is stored in a separate CSS file and linked to the HTML document. It helps maintain consistency and makes your code cleaner.

```html
<!-- In your HTML file -->
<link rel="stylesheet" type="text/css" href="styles.css">
```

CSS Classes and IDs:

CSS classes and IDs allow you to target specific elements for styling. You define them in your HTML and then apply styles in your CSS file.

HTML:

```html
<p class="important">This is an important paragraph.</p>
```

CSS:

```css
.important {
  font-weight: bold;
  color: red;
}
```

IDs are similar but unique to a single element. You define them with `id` and apply styles using `#`.

HTML:

```html
<p id="unique-paragraph">This is a unique paragraph.</p>
```

CSS:

```css
#unique-paragraph {
  font-style: italic;
  color: blue;
}
```

This chapter helps you make your lists more appealing and gives you the power to style your web pages the way you want. You can either include styles directly in your HTML or keep them separate in a CSS file for better organization and maintainability.

Questions 👍

Certainly! Here are some questions for each of the chapters from 1 to 9 to help you test your knowledge and understanding of HTML:

Chapter 1: Introduction to HTML

1. What is HTML, and what role does it play in web development?
2. Explain the purpose of the `<!DOCTYPE>` declaration in an HTML document.
3. What are the basic elements of an HTML document structure?
4. How does the web work in relation to HTML?

Chapter 2: HTML Document Structure

5. Describe the structure of an HTML document, including the purpose of the `<head>` and `<body>` elements.
6. What does the `<meta>` element do, and why is it important?
7. How do you link an external CSS file to an HTML document?
8. What is the function of the `<script>` element in HTML?

Chapter 3: HTML Elements

9. Explain the different types of headings in HTML and their significance.
10. How do you create an ordered list, and what is its purpose?
11. Describe the use of the `<a>` element in creating hyperlinks.
12. How can you add comments in HTML?

Chapter 4: HTML Forms

13. What is the purpose of HTML forms, and how are they structured?
14. Explain the difference between text input fields and text areas.
15. How can you create a dropdown list in an HTML form?
16. What is the role of the submit button in a form?

Chapter 5: HTML Tables

17. How do you create a table in HTML, and what elements are involved?
18. What are table rows, table headers, and table data cells?
19. How can you add a caption to an HTML table?

Chapter 6: HTML Semantic Elements

20. What is the importance of semantic HTML elements in web development?
21. Explain the purpose of `<header>`, `<nav>`, and `<footer>` elements.
22. When should you use semantic elements in your HTML document?

Chapter 7: HTML Links and Multimedia

23. How do you create hyperlinks to external websites in HTML?
24. What are the different ways to embed multimedia, such as audio and video, in HTML?
25. What is an iframe, and how can it be used in HTML?

Chapter 8: HTML Forms and Input Elements

26. How can you validate user input in an HTML form?
27. Explain the concept of hidden fields in HTML forms.
28. Describe the difference between GET and POST methods in form submission.

Chapter 9: HTML Lists and CSS Styling

29. What is the difference between ordered and unordered lists in HTML?
30. How can you create nested lists in HTML?
31. Give an example of applying basic CSS styling to a list.
32. Explain the difference inline CSS and external CSS.
33. How do CSS classes and IDs help in styling HTML elements?

These questions cover a range of topics from the chapters and should help reinforce your understanding of HTML.