Module 6 – HTML, CSS and JS in PHP

1. Introduction to C++

[THEORY EXERCISE]

1. What is HTML? Explain its structure.

➔ HTML (Hypertext Markup Language) is the language used to create web page documents.
➔ HTML stands for Hyper Text Markup Language
➔ An HTML file is a text file containing small markup tags <> <h1> </h1>
➔ The markup tags tell the Web browser how to display the page
➔ An HTML file must have an .htm or .html file extension
➔ An HTML file can be created using a simple text editor

➔ **The structure of an HTML :**

```
<html>

        <head>

                <title> hello world </title>

        </head>

<body>

        This is my first homepage.<b> This text is bold</b>

</body>

</html>
```

➔ HTML Element: **<html> </html>** The root element that contains all other elements.
The first tag in your HTML document is <html>. This tag tells your browser that this is the start of an HTML document. The last tag in your document is </html>. This tag tells your browser that this is end of the HTML document.
➔ Head Element: **<head> </head>** Contains meta-information about the document, such as the title, character set, and links
The text between the <head> tag and the </head> tag is header information. Header information is not displayed in the browser window.
➔ Body Element: **<body> </body>** Contains the content of the document, such as text, images, links, etc.
Title tag & favicon & external files The text between the <body> tags is the text that will be displayed in your browser.

2. Describe the purpose of HTML tags and provide examples of commonly used tags.

➔ HTML tags are used to mark-up HTML elements
  HTML tags are surrounded by the two characters < and >
  The surrounding characters are called angle brackets
  HTML tags normally come in pairs like <b> and </b>
  The first tag in a pair is the start tag, the second tag is the end tag
  The text between the start and end tags is the element content
  HTML tags are not case sensitive, <b> means the same as <B>

  <head>/ <HEAD>

  HTML tags are the building blocks of HTML. They define the structure and content of a webpage. Each tag typically consists of an opening tag, content, and a closing tag.

  Some tags are self-closing.

```
<h1>Main Title</h1>

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

<a href="https://www.google.com"

<img src="image.jpg" alt="Description“

 <div>

    <p>Content inside a div.</p>

 </div>
```

3. What are the differences between block-level and inline elements? Give examples of each.

➔ **Block-level elements :** are elements that take up the full width available and start on a new line. They create a "block" of content.

- <div>

  <h1>, <h2>, <h3>, etc.

  <p>

  <ul>, <ol>, <li>

➔ **Inline elements :** on the other hand, only take up as much width as necessary and do not start on a new line. They can be placed within block-level elements.

- <span>

  <a>

  <img>

  <strong>, <em>

4. Explain the concept of semantic HTML and why it is important.

Semantic HTML refers to the use of HTML markup that conveys meaning about the content contained within the tags. Instead of using generic tags like <div> and <span>, semantic HTML uses tags that describe the role of the content, such as <header>, <footer>, <article>, <section>, and <nav>.

➔ **Accessibility:** Screen readers and other assistive technologies can better interpret the content, making it more accessible to users with disabilities.

➔ **SEO (Search Engine Optimization):** Search engines can better understand the structure and context of the content, which can improve search rankings.

➔ **Maintainability:** Code is easier to read and maintain when it uses meaningful tags, making it clearer for developers to understand the structure and purpose of the content.

➔ **Future-proofing:** As web standards evolve, using semantic HTML can help ensure that content remains relevant and usable across different platforms and devices.

1. CSS Fundamentals

[THEORY EXERCISE]

1. What is CSS? How does it differ from HTML?

CSS, or Cascading Style Sheets, is a stylesheet language used to describe the presentation and layout of a document written in HTML or XML. While HTML provides the structure and content of a webpage, CSS is responsible for the visual appearance, including colors, fonts, spacing, and positioning of elements.

Differences between CSS and HTML :

- Purpose : HTML is used to create the structure and content of a webpage, while CSS is used to style and format that content.
- Syntax : HTML uses tags to define elements, whereas CSS uses selectors and properties to apply styles.
- Functionality : HTML is focused on the semantic meaning of content, while CSS focuses on the visual representation and layout.

2. Explain the three ways to apply CSS to a web page

1. Inline CSS :

CSS styles are applied directly within an HTML element using the `style` attribute. This method is useful for quick, one-off styles but is not recommended for larger projects due to maintainability issues.

Html

```
<h1 style="color: blue; font-size: 24px;">Hello, World!</h1>
```

2. Internal CSS :

CSS styles are defined within a `<style>` tag in the `<head>` section of the HTML document. This method is suitable for single-page styles and allows for easier management than inline styles.

```
  <head>
     <style>
        h1 {
           color: blue;
           font-size: 24px; }
```

```
    </style>
  </head>
```

3. External CSS :

CSS styles are written in a separate `.css` file and linked to the HTML document using a `<link>` tag in the `<head>`. This method is the most efficient for larger projects, as it allows for consistent styling across multiple pages.

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

3. What are CSS selectors? List and describe the different types of selectors.

CSS selectors are patterns used to select the elements you want to style. They determine which HTML elements the CSS rules apply to. Different types of selectors include:

1. Universal Selector (`*`) : Selects all elements on the page.

```
  * {
    margin: 0;
    padding: 0;
  }
```

2. Type Selector : Selects all elements of a specific type (e.g., all `<p>` elements).

```
  p {
    color: red;
  }
```

3. Class Selector (`.`) : Selects elements with a specific class attribute. Classes are reusable.

```
  .my-class {
    font-size: 16px;
  }
```

4. ID Selector (`#`) : Selects a single element with a specific ID attribute. IDs should be unique within a page.

```
  #my-id {
    background-color: yellow; }
```

5. Attribute Selector : Selects elements based on the presence or value of an attribute.

```
a[target="_blank"] {
   color: green;
}
```

6. Descendant Selector : Selects elements that are descendants of a specified element.

```
div p {
   color: blue;
}
```

7. Child Selec+tor (`>`) : Selects elements that are direct children of a specified element.

```
ul > li {
   list-style-type: square;
}
```

8. Adjacent Sibling Selector (`+`) : Selects an element that is immediately following a specified element.

```
h1 + p {
   margin-top: 10px;
}
```

9. General Sibling Selector (`~`) : Selects all siblings of a specified element.

```
h1 ~ p {
   color: gray;
}
```

4. What is the box model in CSS? Explain its components.

The CSS box model is a fundamental concept that describes how elements are structured and how their dimensions are calculated. Every element on a webpage is represented as a rectangular box, which consists of several components:

1. Content : The actual content of the box, such as text, images, or other media. The size of the content area can be controlled using the `width` and `height` properties.

2. Padding : The space between the content and the border of the box. Padding creates space inside the box and can be set using the `padding` property. It can be specified for each side (top, right, bottom, left) or as a shorthand.

```
padding: 10px; /* All sides */
padding: 10px 20px; /* Vertical | Horizontal */
```

3. Border : A line that surrounds the padding (if any) and content. The border can be styled using the `border` property, which allows you to set the width, style, and color.

```
border: 2px solid black;
```

4. Margin : The space outside the border, creating distance between the box and other elements. Margins can also be set for each side or as shorthand.

```
margin: 20px; /* All sides */
margin: 10px 15px; /* Vertical | Horizontal */
```

The total width and height of an element can be calculated

- Total Width : `width + padding-left + padding-right + border-left + border-right + margin-left + margin-right`

- Total Height : `height + padding-top + padding-bottom + border-top + border-bottom + margin-top + margin-bottom`

3. Responsive Web Design

[THEORY EXERCISE]

1. What isresponsive web design? Why is it important?

Responsive web design (RWD) is an approach to web design that aims to create websites that provide an optimal viewing experience across a wide range of devices, from desktop computers to mobile phones. This is achieved by using flexible layouts, images, and CSS media queries to adapt the website's appearance based on the screen size and orientation of the device being used.

Importance of Responsive Web Design :
1. Improved User Experience : RWD ensures that users have a seamless experience regardless of the device they are using. This leads to higher user satisfaction and engagement.

2. Increased Mobile Traffic : With the growing use of mobile devices for browsing, having a responsive design allows websites to cater to mobile users effectively, capturing a larger audience.

3. SEO Benefits : Search engines, like Google, prioritize mobile-friendly websites in their rankings. A responsive design can improve a site's visibility and search engine optimization (SEO).

4. Cost-Effectiveness : Maintaining a single responsive website is more cost-effective than creating separate versions for different devices. It simplifies updates and maintenance.

5. Future-Proofing : RWD prepares websites for future devices and screen sizes, ensuring they remain functional and visually appealing as technology evolves.

2. Explain the use of media queriesin CSS. Provide an example.

Media queries are a feature of CSS that allow developers to apply different styles to a webpage based on the characteristics of the device, such as its width, height, resolution, and orientation. Media queries enable responsive design by allowing specific styles to be applied only when certain conditions are met.

```
/* Default styles for all devices */
body {
   font-size: 16px;
   background-color: white;
}

/* Styles for devices with a maximum width of 768px (tablets and mobile devices) */
@media (max-width: 768px) {
   body {
      font-size: 14px;
      background-color: lightgray;
   }
}

/* Styles for devices with a minimum width of 769px (desktops) */
@media (min-width: 769px) {
   body {
      font-size: 18px;
      background-color: white;
   }
}
```

In this example, the default styles apply to all devices. When the viewport width is 768 pixels or less, the font size decreases, and the background color changes to light gray. For devices wider than 769 pixels, the font size increases, and the background color remains white.

3. What are the benefits of using a mobile-first approach in web design?

A mobile-first approach is a design strategy that prioritizes designing for mobile devices before scaling up to larger screens. This approach has several benefits:

1. Enhanced Performance : Mobile-first designs often lead to lighter, faster-loading websites, as they focus on essential content and features for smaller screens. This can improve load times and overall performance.

2. Better User Experience : By starting with mobile users in mind, designers can create a more intuitive and user-friendly experience. This ensures that essential features are easily accessible on smaller screens.

3. Progressive Enhancement : A mobile-first approach encourages the use of progressive enhancement, where the core functionality is available on all devices, and additional features are added for larger screens. This ensures that all users have a functional experience, regardless of their device.

4. Improved SEO : Search engines favor mobile-friendly websites, and a mobile-first design can help improve search rankings. Google's mobile-first indexing means that the mobile version of a site is considered the primary version for ranking purposes.

5. Future-Proofing : As mobile usage continues to grow, designing with mobile in mind prepares websites for future trends and technologies, ensuring they remain relevant and functional.

6. Cost-Effectiveness : By focusing on mobile first, developers can avoid unnecessary complexity in the design process, leading to more efficient development and maintenance.

4. PHP Integration

[THEORY EXERCISE]

1. How can PHP be used to dynamically generate HTML content? Provide examples.

PHP (Hypertext Preprocessor) is a server-side scripting language that can be used to dynamically generate HTML content based on various conditions, such as user input, database queries, or other server-side logic. This allows developers to create interactive and personalized web pages.

- **Example 1: Basic Dynamic HTML Generation**

-
  ```
  <?php
  $name = “John Doe”;
  Echo “<h1>Welcome, $name!</h1>”;
  ?>
  ```

- **Example 2: Generating a List from an Array**

  -
    ```
    <?php
    $fruits = [“Apple”, “Banana”, “Cherry”];
    Echo “<ul>”;
    Foreach ($fruits as $fruit)
    {
    Echo “<li>$fruit</li>”;
    }
    Echo “</ul>”;
    ?>
    ```

- **Example 3: Generating HTML Based on User Input**

  -
    ```
    <?php
    If ($_SERVER[“REQUEST_METHOD”] == “POST”)
    {
       $username = htmlspecialchars($_POST[‘username’]);
       Echo “<h2>Hello, $username!</h2>”;
    }


    <form method=”post” action=””>
       <label for=”username”>Enter your name:</label>
       <input type=”text” id=”username” name=”username”>
       <input type=”submit” value=”Submit”>
    </form>
    ```

2. Explain how to include CSS files in a PHP-generated HTML page.

To include CSS files in a PHP-generated HTML page, you can use the standard HTML `<link>` tag within the `<head>` section of your PHP script. This is done similarly to how you would include CSS in a regular HTML document.

Example: Including a CSS File in PHP

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>My PHP Page</title>
   <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
   <h1>Welcome to My PHP Page</h1>
   <p>This page uses an external CSS file for styling.</p>
</body>
</html>
```

In this example, the `<link>` tag is used to include an external CSS file named `styles.css`.

4. What are the advantages of using PHP to manage HTML forms?

Using PHP to manage HTML forms offers several advantages:

1. **Server-Side Processing :** PHP allows for server-side processing of form data, enabling validation, sanitization, and storage in databases. This ensures that data is handled securely and efficiently.
2. **Dynamic Content Generation :** PHP can generate dynamic content based on user input. For example, after form submission, PHP can display personalized messages or redirect users to different pages based on their input.
3. **Data Validation and Error Handling :** PHP can validate form data before processing it. This helps ensure that the data submitted is in the correct format and meets specific criteria, reducing the risk of errors and improving user experience.
4. **Database Integration :** PHP can easily connect to databases (e.g., MySQL) to store or retrieve form data. This allows for the creation of dynamic applications, such as user registration systems, feedback forms, and content management systems.
5. **Session Management :** PHP can manage user sessions, allowing for the retention of user data across multiple pages. This is useful for applications that require user authentication or tracking user activity.
6. **Security Features :** PHP provides various functions to help secure form data, such as escaping special characters to prevent SQL injection and cross-site scripting (XSS) attacks.
7. **Ease of Use :** PHP is relatively easy to learn and integrate with HTML, making it accessible for developers who want to create interactive web applications without extensive knowledge of other programming languages.