22

There are three primary methods to embed CSS into an HTML document:

1. **Inline CSS:**
   Inline CSS involves using the `style` attribute directly within HTML elements. Each element that needs styling will have its CSS rules defined within the `style` attribute.
   - Quick to apply: Ideal for making quick changes or testing specific styles without editing separate CSS files.
   - Specificity: Applies styles directly to the element, which can override other styles easily.
   - Maintenance difficulty: Hard to manage and update styles, especially in larger projects, since styles are scattered across many HTML elements.
   - Poor separation of concerns: Mixes content (HTML) and presentation (CSS), making the code harder to read and maintain.
   - Redundancy: Can lead to repetitive code if the same styles are applied to multiple elements.
   - Suitable for small projects or single elements needing unique styles.
   - Useful for quick fixes or testing new styles.

2. **Internal (Embedded) CSS:**
   Internal CSS places CSS rules within a `<style>` tag in the `<head>` section of an HTML document. These styles apply to the entire document, allowing for centralized style management.
   - Centralized control: Keeps all styles for the document in one place, making it easier to manage and update.
   - Isolation: Keeps styles within the HTML document, avoiding the need for external files.
   - Limited reusability: Styles are only applicable to that specific HTML document, making it difficult to reuse styles across multiple pages.
   - Increased page size: Embedding styles within the HTML can increase the size of the document, potentially affecting load times.
   - Suitable for single-page applications or documents.
   - Useful when prototyping or during the initial development phase before styles are moved to an external stylesheet.

3. **External CSS:**
   External CSS involves linking to an external stylesheet using the `<link>` tag in the `<head>` section of the HTML document. The stylesheet contains CSS rules that can be applied to multiple HTML documents.
   - Reusability: Styles can be reused across multiple HTML documents, promoting consistency and reducing redundancy.
   - Separation of concerns: Keeps HTML and CSS separate, making the codebase easier to maintain and read.
   - Reduced page size: The HTML document is kept lean as styles are loaded from an external file.
   - Additional HTTP request: Requires an additional request to load the stylesheet, which can affect page load times, although this can be mitigated by caching.
   - Dependency: Requires the presence of the external stylesheet file, so if it's missing or incorrectly linked, styles won't be applied.
   - Ideal for larger projects with multiple pages needing consistent styling.
   - Beneficial for maintaining a clean separation between HTML content and CSS styling.

Here is a PHP program to check whether a number is prime or not:

```php
<?php
function isPrime($number)
{
    if ($number < 2) {
        return false;
    }
    for ($i = 2; $i <= sqrt($number); $i++) {
        if ($number % $i == 0) {
            return false;
        }
    }
    return true;
}
```

14. Explain tag in HTML: A tag in HTML is a syntactic construct that defines the structure and presentation of content on a web page. Tags typically consist of an opening tag and a closing tag, with content in between, or a self-closing tag. Tags are enclosed in angle brackets and can include attributes that provide additional information about the element. Tags tell the browser how to display the content and can represent elements like headings, paragraphs, links, images, and more.

15. Discuss comparison and assignment operators in JavaScript:Comparison Operators: These operators compare two values and return a boolean result (true or false). Common comparison operators include:== (equal to)=== (strict equal to, checks both value and type)!= (not equal to)!== (strict not equal to, checks both value and type)< (less than)> (greater than)<= (less than or equal to)>= (greater than or equal to)Assignment Operators: These operators assign values to variables. Common assignment operators include:= (simple assignment)+= (addition assignment)-= (subtraction assignment)*= (multiplication assignment)/= (division assignment)%= (modulus assignment)

16. Explain different Decision Control Statements in JavaScript:if statement: Executes a block of code if a specified condition is true.if...else statement: Executes one block of code if a specified condition is true, and another block if the condition is false.else if statement: Specifies a new condition to test if the first condition is false.switch statement: Evaluates an expression, matches the expression's value to a case clause, and executes the associated block of code.

17. Explain any three display statements available in PHP:echo: Outputs one or more strings. It is the most commonly used way to output data to the browser.print: Outputs a string. It is similar to echo but always returns 1, so it can be used in expressions.print_r: Outputs information about a variable in a human-readable form. It is particularly useful for displaying arrays and objects.

13. Four Key Concepts of HTML
1. **Elements and Tags**: HTML uses tags to define elements. Elements are the building blocks of HTML pages and are represented by tags, usually in pairs like `<tag>` and `</tag>`.
2. **Attributes**: Attributes provide additional information about elements and are included in the opening tag. They typically consist of a name and a value.
3. **Document Structure**: HTML documents have a defined structure with a `<!DOCTYPE>` declaration, and elements like `<html>`, `<head>`, and `<body>` that outline the document's layout.
4. **Hyperlinks**: HTML enables linking to other documents or resources using the `<a>` tag, allowing navigation across different web pages or sites.

14. Attributes of Tag
Attributes in HTML tags provide additional information about elements and define their properties. Attributes are written within the opening tag and are usually in a name-value pair format. They help in specifying behaviors, styling, and other settings of HTML elements.

15. Five Pseudo Classes in CSS
1. **:hover**: Applies styles to an element when the mouse pointer is over it.
2. **:active**: Applies styles to an element when it is being activated, for example, a button being clicked.
3. **:focus**: Applies styles to an element when it has received focus, such as an input field when clicked or navigated to with the keyboard.
4. **:nth-child(n)**: Selects the nth child of a parent element, allowing for styling specific items in a list or set.
5. **:first-child**: Applies styles to the first child of a parent element.

16. Form Validation and Its Importance
Form validation is the process of ensuring that user input is correct and meets the necessary criteria before it is processed. It can be done on the client side using JavaScript or on the server side using server-side languages. Validation helps prevent incorrect or malicious data from being submitted, improving data integrity and security.

17. Features of PHP
- **Server-side Scripting**: PHP runs on the server and generates dynamic content for web pages.
- **Cross-Platform**: PHP works on various operating systems like Windows, Linux, and macOS.
- **Embedded in HTML**: PHP code can be embedded within HTML, making it easy to integrate with web pages.
- **Database Support**: PHP has built-in support for various databases like MySQL, PostgreSQL, and SQLite.
- **Open Source**: PHP is free to use and has a large community providing support and resources.

18. Multidimensional Array in PHP
A multidimensional array is an array containing one or more arrays within it. It allows for storing data in a tabular form. It is declared by nesting arrays within another array and accessed using multiple indices.

19. Five String Handling Functions in PHP
1. **strlen()**: Returns the length of a string.
2. **str_replace()**: Replaces all occurrences of a search string with a replacement string within another string.
3. **strpos()**: Finds the position of the first occurrence of a substring within a string.
4. **substr()**: Returns a portion of a string specified by start and length parameters.
5. **strtolower()**: Converts all characters in a string to lowercase.

20. Error Levels Used in PHP
- **E_ERROR**: Fatal run-time errors that cause script termination.
- **E_WARNING**: Run-time warnings (non-fatal errors) that do not halt script execution.
- **E_PARSE**: Compile-time parse errors.
- **E_NOTICE**: Run-time notices indicating possible issues in the code.
- **E_STRICT**: Suggests changes to improve code compatibility and interoperability.

21. Use of mysql_fetch_array()
`mysql_fetch_array()` is used to fetch a result row as an associative array, a numeric array, or both from a result set obtained by a `mysql_query()`. It allows for flexible retrieval of data from a MySQL database.
Example:
```php
<?php
$result = mysql_query("SELECT * FROM users");
while ($row = mysql_fetch_array($result)) {
    echo $row['username']; // Access by column name
    echo $row[0]; // Access by column index
}
?>
```

18. Write a PHP program to find the factorial of a number:
```php
<?php
function factorial($number) {
    if ($number <= 1) {
        return 1;
    } else {
        return $number * factorial($number - 1);
    }
}

// Example usage
$number = 5;
echo "The factorial of $number is " . factorial($number);
?>
```

13. Text Formatting Tags in HTML
Text formatting tags in HTML are used to define the appearance and structure of text. Common tags include:
- `<b>` for bold text
- `<i>` for italic text
- `<u>` for underlined text
- `<strong>` for important text (usually bold)
- `<em>` for emphasized text (usually italic)
- `<mark>` for highlighted text
- `<small>` for smaller text
- `<del>` for strikethrough text
- `<ins>` for inserted text, usually underlined
- `<sub>` for subscript text
- `<sup>` for superscript text

14. How to Create an HTML Form
To create an HTML form, you use the `<form>` tag along with various input elements. The basic structure includes:
- `<form>`: Container for form elements.
- `<input>`: Collects user input (e.g., text, radio, checkbox).
- `<label>`: Describes the input elements.
- `<textarea>`: For multi-line text input.
- `<select>`: Drop-down list.
- `<button>`: Submit or reset actions.

15. CSS Text Properties
CSS text properties control the appearance and layout of text. Key properties include:
- `color`: Sets the color of the text.
- `font-family`: Specifies the font for the text.
- `font-size`: Defines the size of the text.
- `font-style`: Applies italic style to text.
- `font-weight`: Sets the thickness of the text.
- `text-align`: Aligns text left, right, center, or justify.
- `text-decoration`: Adds decorations like underline, overline, or line-through.
- `text-transform`: Controls capitalization (uppercase, lowercase, capitalize).
- `letter-spacing`: Adjusts space between characters.
- `line-height`: Sets the height between lines of text.

16. Confirm Box vs. Prompt Box in JavaScript
- **Confirm Box**: Displays a dialog box with a message and "OK" and "Cancel" buttons. It returns `true` if "OK" is clicked and `false` if "Cancel" is clicked.
- **Prompt Box**: Displays a dialog box that prompts the user to enter input. It has a message, a text input field, and "OK" and "Cancel" buttons. It returns the input value if "OK" is clicked and `null` if "Cancel" is clicked.

17. Comments in PHP
Comments in PHP are used to leave notes or explanations within the code and are ignored during execution. Types of comments include:
- Single-line comments: `//` or `#`
- Multi-line comments: `/* */`

19. Commonly Used Array Functions in PHP
1. `array_merge()`: Combines one or more arrays.
2. `array_push()`: Adds one or more elements to the end of an array.
3. `array_pop()`: Removes and returns the last element of an array.
4. `array_shift()`: Removes and returns the first element of an array.
5. `array_unshift()`: Adds one or more elements to the beginning of an array.

20. PHP Session Management
PHP manages sessions using a unique session ID for each user. Sessions allow data to persist across multiple pages. Key functions include:
- `session_start()`: Begins a session or resumes an existing one.
- `$_SESSION`: Superglobal array used to store session variables.
- `session_destroy()`: Ends the session and clears session data.

21. The SELECT Statement and Basic Querying Techniques
The `SELECT` statement in SQL is used to retrieve data from a database. Basic querying techniques include:
- Selecting specific columns: `SELECT column1, column2 FROM table_name;`
- Selecting all columns: `SELECT * FROM table_name;`
- Using `WHERE` clause to filter results: `SELECT * FROM table_name WHERE condition;`
- Sorting results with `ORDER BY`: `SELECT * FROM table_name ORDER BY column;`
- Limiting results with `LIMIT`: `SELECT * FROM table_name LIMIT number;`
- Joining tables with `JOIN`: `SELECT * FROM table1 JOIN table2 ON table1.column = table2.column;`

19. How PHP handles checkboxes and radio buttons in a form? Explain with example: In PHP, checkboxes and radio buttons are handled by checking if they are set in the $_POST or $_GET superglobals. Each checkbox or radio button should have a name attribute. When the form is submitted, PHP can check if a checkbox or radio button was selected by using isset() or checking if the value exists in the $_POST or $_GET arrays.Example:
```php
<form method="post" action="process_form.php">
    <input type="checkbox" name="option1" value="1"> Option 1<br>
    <input type="checkbox" name="option2" value="2"> Option 2<br>
    <input type="radio" name="radio_option" value="A"> Radio A<br>
    <input type="radio" name="radio_option" value="B"> Radio B<br>
    <input type="submit" value="Submit">
</form><?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (isset($_POST['option1'])) {
        echo "Option 1 is checked.";
    }
    if (isset($_POST['option2'])) {
        echo "Option 2 is checked.";
    }
    if (isset($_POST['radio_option'])) {
        echo "Selected radio option: " . $_POST['radio_option'];
    }
}
?>
```