PCET's Pimpri Chinchwad University
Learn | Grow | Achieve

PIMPRI CHINCHWAD UNIVERSITY
School of Engineering and Technology

Academic Year: 2024-2025 **Sem:** I

Course: WEB PROGRAMMING **Course code**: **UBTFY113/SEC**

Name: **Roll No:**

Practical No. 1

Title:

A. Create HTML document with formatting as: bold, italics, underline, colors, heading, title, font, background, paragraph etc.

B. Build a personal Bio Data using HTML and CSS. Create an HTML document with appropriate elements such as headings, paragraphs, and lists Use CSS to style the page, including fonts, colors, and layout Add images or icons to enhance the visual appeal

Aim/ Objective:

Part A:

Aim: The primary aim of this task is to demonstrate the basic formatting capabilities of HTML and CSS. This includes applying various text styles, colors, and structuring content to create a visually appealing document.

Objectives:

1. **Understanding HTML Structure: Familiarize users with the basic structure of an HTML document, including elements like `<head>`, `<body>`, and various formatting tags.**

2. **Text Formatting: Illustrate how to use HTML tags for bold, italics, underline, and color changes to enhance text presentation.**
3. **Utilizing CSS: Show how CSS can be used to style HTML elements, control layout, and improve overall aesthetics.**
4. **Content Organization: Teach the importance of organizing content using headings, paragraphs, and lists to make information clear and accessible.**

Part B:

B. Aim and Objective of the Personal Bio Data Using HTML and CSS

Aim: The aim of creating a personal bio data document is to showcase an individual's information in a structured and visually appealing format using HTML and CSS.

Objectives:

1. **Personal Presentation: Provide a clear and organized way to present personal information, education, skills, and hobbies.**
2. **Effective Use of Headings and Lists: Highlight the importance of using appropriate HTML elements like headings and lists for better readability and organization.**
3. **Styling with CSS: Demonstrate how to use CSS for styling, including background colors, font choices, and layout design, to create an engaging visual presentation.**
4. **Image Integration: Show how to incorporate images to enhance the visual appeal and personal touch of the bio data.**
5. **Responsive Design: Introduce the concept of responsive design by using flexible layouts that work well on different devices (though basic in this example).**

Introduction:

Part A:

A. Tags Used for HTML Document Formatting

1. **Document Structure Tags:**
    - **`<!DOCTYPE html>`: Declares the document type.**
    - **`<html>`: Root element of the HTML document.**
    - **`<head>`: Contains meta-information about the document (e.g., title, links to styles).**
    - **`<body>`: Contains the content of the document that is displayed in the browser.**
2. **Text Formatting Tags:**
    - **Bold Text: `<strong>` or `<b>` (though `<strong>` is preferred for semantic meaning).**
    - **Italic Text: `<em>` or `<i>` (again, `<em>` is preferred for semantic meaning).**
    - **Underline Text: `<u>`: Underlines the text.**
3. **Coloring Text:**
    - **This is typically done with CSS, but you can also use the `<font>` tag (deprecated) for setting color, like `<font color="red">Text</font>`. Preferred method is via CSS.**

Example CSS:

```
.highlight { color: red; }
```

4. **Headings:**
    - **`<h1>` to `<h6>`: Define headings, with `<h1>` being the largest and `<h6>` the smallest.**
5. **Title:**
    - **`<title>`: Sets the title of the document, displayed in the browser tab.**
6. **Font:**
    - **The `<font>` tag is deprecated. Use CSS for font properties (e.g., `font-family`, `font-size`).**

Example CSS:

```
body { font-family: Arial, sans-serif; }
```

-

7. **Background:**
    - **Set using CSS for the body or specific elements.**

Example CSS:

```
body { background-color: #f4f4f4; }
```

8. **Paragraph:**
    - **`<p>`: Defines a paragraph of text.**
9. **Lists:**
    - **`<ul>`: Unordered list.**
    - **`<ol>`: Ordered list.**
    - **`<li>`: List item.**

Part B:

HTML Tags Used for Personal Bio Data

1. **Document Structure Tags:**
    - **`<!DOCTYPE html>`: Declares the document type.**
    - **`<html>`: Root element of the HTML document.**
    - **`<head>`: Contains meta-information about the document.**
    - **`<body>`: Contains the content that is displayed in the browser.**
2. **Headings:**
    - **`<h1>`: Main heading (e.g., title of the bio data).**
    - **`<h2>`, `<h3>`: Subheadings for sections like "Personal Information," "Education," etc.**
3. **Paragraphs:**
    - **`<p>`: Defines paragraphs for text content.**

4. **Lists:**
    - `<ul>`: Unordered list (e.g., for hobbies or skills).
    - `<ol>`: Ordered list (if there is a specific order).
    - `<li>`: List item.
5. **Images:**
    - `<img>`: Used to include images, such as a profile picture.
    - Example: `<img src="profile.jpg" alt="Profile Image">`
6. **Links:**
    - `<a>`: Defines hyperlinks to connect to other pages or resources (e.g., email link).
    - Example: `<a href="mailto:john.doe@example.com">Email Me</a>`
7. **Divisions and Spans:**
    - `<div>`: Used for sectioning content or creating layout divisions.
    - `<span>`: Used to style specific inline elements or portions of text.

CSS Properties Used for Styling

1. **Fonts:**
    - `font-family`: Sets the font type (e.g., `font-family: Arial, sans-serif;`).
    - `font-size`: Sets the size of the font.
2. **Colors:**
    - `color`: Sets the text color.
    - `background-color`: Sets the background color of elements.
3. **Layout:**
    - `margin`: Adds space outside an element.
    - `padding`: Adds space inside an element.
    - `border`: Defines the border properties of elements.

- **`max-width`: Restricts the width of elements for better layout.**
- **`text-align`: Aligns text (e.g., `text-align: center;`).**

4. **Images:**
   - **`width` and `height`: Controls the size of images.**
   - **`border-radius`: Creates rounded corners for images.**

Procedure/ Steps/Algorithm/Flowchart:(any one)

Procedure/Steps/Algorithm

1. **Set Up Environment:**
   - **Open a text editor (e.g., VS Code, Notepad, Sublime Text) or an integrated development environment (IDE).**
   - **Create a new file and save it with a `.html` extension (e.g., `formatted-document.html`).**
2. **Declare Document Type:**
   - **Start with `<!DOCTYPE html>` to define the document type.**
3. **Create HTML Structure:**
   - **Add the `<html>` tag to begin the HTML document.**
   - **Inside `<html>`, add the `<head>` section:**
     - **Include `<meta charset="UTF-8">` for character encoding.**
     - **Include `<meta name="viewport" content="width=device-width, initial-scale=1.0">` for responsive design.**
     - **Add a `<title>` tag to set the document title.**
4. **Add CSS Styles:**
   - **Inside the `<head>`, add a `<style>` section to include CSS for formatting:**
     - **Define `body` styles (background color, font family, etc.).**

        - Define styles for headings, paragraphs, and text formatting (bold, italics, underline, colors).
5. Create Body Content:
    - Open the `<body>` tag.
    - Add headings using `<h1>`, `<h2>`, etc.
    - Write paragraphs using the `<p>` tag.
    - Use formatting tags (`<strong>`, `<em>`, `<u>`, `<span>`) for text formatting.
    - Create lists using `<ul>` or `<ol>` with `<li>` for list items.
6. Close HTML Tags:
    - Make sure to properly close all tags (`</body>`, `</html>`).
7. Save and Open in Browser:
    - Save the HTML file.
    - Open the file in a web browser to view the formatted document.
8. Debug and Adjust:
    - Check the formatting and appearance.
    - Make any necessary adjustments to HTML or CSS.

OR

Flowchart

Here's a simple flowchart to visualize the process:

Create New HTML File

Add <!DOCTYPE html>

Add <html> and <head>

Add <meta> and <title>

Add <style> for CSS

Add <body>

Add Headings and Paragraphs

Add Formatting (bold, italics, etc.)

Add Lists

Close Tags (</body>, </html>)

Save File

Open in Browser

Check Formatting

Adjust as Needed

End

Input:Write code on Notepad or any text editor,Visual studio

Output:Display on web Browser

Testing: System Testing

System testing is a critical phase in the software development lifecycle (SDLC) that involves evaluating the complete and integrated software system to ensure it meets specified requirements. Here's an overview of system testing, including its purpose, types, process, and best practices.

Conclusion:

Hence , All tags studies and executed successfully.

References:

https://www.w3schools.com/html/

Questions:

1. **What is Html and Its structure?**
2. **Explain Anatomy of HTML.**
3. **Write down syntax related to CSS Text properties.**

4. **Display code related to Inline,internal and External CSS.**
5. **Difference between web Browser and web server.**

Date of Performance:	**Faculty Sign:**
Date of Submission:	
Marks:-	