WEB PROGRAMMIG PRACTICAL

Title: Build a personal Bio data using HTML and CSS. Create an HTML document with appropriate elements such as headings paragraph and lists use CSS to style the page, including fonts, colors and layout Add images or icons to enhance the visual appeal.

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

Objective:
- Personal presentation: provide a clear or organised way to present personal information education skills and hobbies.
- Effective use of headings and lists: Highlight the importance of using appropriate HTML elements like headings and lists for better readability and organisation
- Styling with CSS: Demostrate how to use CSS for styling including background colors, font choices, layout design.
- Image Integration: show how to incorporate images to enhance the visual studios.
- Responsive Design: Introduce the concept of responsive by using flexible layout that work well on different devices.

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
<body> : contains the content that is displayed

2) Heading: <h1> Main Heading
<h2>, <h3> subheadings for sections like personal information Education etc.

3) Paragraph: <p> defines paragraph

4) Lists : <ul> : unordered list
 : ordered list.
 : list item.

5) Images: <img> used to include images, such as a profile photo.
Example: <img src="profile.jpg" alt="profile Image">

6) Links : <a> defines hyperlinks to connect to other pages or resources

7) Division and span: Used for sectioning content or creating layout. division
 used to style specific inline elements or portions of text.

CSS properties used for styling:-

Fonts: Font family: sets the font type.
Font size: sets the size of the font.

Colors: colors: sets the text color
background colour: sets the background color of element.

Layout: Margin: Adds space outside an element
- padding: adds space inside an element.
- border: Defines the border properties of element.
- Max width: Restricts the width of element for better layout.
- Text-align: Aligns Text.

Images: Width and height: control the size of image.
- Border radius: Creates rounded corners for images.

Procedure: 1) Step up Environment:-
- open a text editor or an integrated development environment.
- Create a new file and save it with a .html extension.

2) Declare Document Type: start with <!DOCTYPE html> to define the document type.

3) Creating 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 style. (background color, font family)
- Define styles for heading, paragraphs, and text formatting (bold, italic, underline, color)

5) Create body content:-
- open the <body> tag
- add heading using the <h> tag.
- Use formatting tags (, , <u>) for text formatting.
- create lists using <ul> and <ol> with <li> 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

Conclusion:- Hence, All Tags studied are executed successfully.

References:- https://www.w3school.com/html/.