Assignment Code: FSD-AG-002

CSS | Assignments

Instructions: Carefully read each question. Use Google Docs, Microsoft Word, or a similar tool to create a document where you type out each question along with its answer. Save the document as a PDF, and then upload it to the LMS. Please do not zip or archive the files before uploading them. Each question carries 20 marks.

Total Marks: 160

Question 1 : What are semantic HTML elements? Why is using them important for web development?

Answer:

Semantic HTML elements are tags that describe the meaning and purpose of the content they contain, such as <header>, <nav>, <main>, <article>, <section>, and <footer>.

Importance in web development:

- Improves SEO, since search engines better understand page structure.
- Enhances accessibility, helping screen readers and assistive tools.
- Makes code more readable and maintainable for developers.

Question 2: You're designing a blog page. Which semantic elements would you use to structure the page, and why?

Answer:

These semantic elements give the page clear structure, improve readability, SEO, and accessibility.

- <header> → For the blog title and navigation.
- <nav> → To group links (menu, categories).
- <main> → To hold the main blog content.
- <article> → For each blog post (self-contained).
- <section> → To divide content within posts (intro, body, comments).
- <aside> → For sidebars (ads, related posts, author bio).
- <footer> → For copyright, contact info, and links.

Question 3: How can you make an HTML form more accessible to users with disabilities?

Answer:

1. Use <label> for every input
   - Associate labels with form controls using the for attribute.
2. Provide meaningful placeholder/help text
   - Use placeholder for hints, but don't rely on it alone—always have a label.
   - Add instructions outside placeholders for clarity.
3. Provide clear error messages
   - Use text (not just color) to explain errors.
   - Example: "Password must be at least 8 characters".
4. Group related fields
   - Use <fieldset> and <legend> for grouped controls (e.g., radio buttons).

Question 4: Identify and correct the errors in the following CSS code:

```
p {
  font-size: 16;
  color: #333
  margin-top 10px;
}
```

Answer:

font-size: 16; → Missing a unit (must be 16px, 16em, etc.).
color: #333 → Missing a semicolon (;) at the end.
margin-top 10px; → Missing a colon (:) between property and value.

Corrected CSS:

```
p {
  font-size: 16px;
  color: #333;
  margin-top: 10px;
}
```

Question 5: Write CSS rules to style all **<h2>** elements inside a **<section>** with a blue color and center alignment.

Answer:

```
section h2 {
  color: blue;            /* Text color blue */
  text-align: center;     /* Center the text */
}
```

Question 6: Explain the CSS box model and its components.

Answer:

The CSS box model is a way to understand how elements are displayed and how their size and spacing are calculated on a webpage. Every HTML element is treated as a rectangular box, which consists of four main components:

1. Content
- The innermost part of the box where text, images, or other content appears.
- Controlled by properties like width and height.

2. Padding
- The space between the content and the border.
- Makes the content not touch the edges of the box.
- Example: padding: 10px; adds 10px of space inside the box.

3. Border
- The line surrounding the padding and content.
- Controlled with properties like border-width, border-style, border-color.
- Example: border: 2px solid black.

4. Margin
- The outermost space that separates the element from other elements.
- Creates spacing between elements without affecting the element itself.
- Example: margin: 20px.

Question 7: How do the relative, absolute, and fixed positioning properties differ in CSS?

Answer:

In CSS, positioning determines how an element is placed on a page. The three commonly used types relative, absolute, and fixed work differently:

1. Relative (position: relative)
- The element is positioned relative to its normal position in the document flow.
- Other elements still behave as if it's in its original place.
- You can use top, right, bottom, left to move it slightly from its original spot.

2. Absolute (position: absolute)
- The element is removed from the normal document flow.
- Positioned relative to its nearest positioned ancestor (relative, absolute, or fixed).
- Other elements behave as if it doesn't exist, so it can overlap them.

3. Fixed (position: fixed)
- The element is removed from the normal flow and fixed relative to the viewport.
- It stays in place even when you scroll the page.

Question 8: Write a CSS rule to set a background image for a ** ** with the class ***.banner***, ensuring the image covers the entire area without repeating.

Answer:

```
.banner {
  background-image: url("./dog.jpeg");        /* Set your image path */
  background-size: cover;                     /* Makes image cover entire div */
  background-repeat: no-repeat;               /* Prevents the image from repeating */
  background-position: center;                /* Centers the image in the div */
}
```