ITWS 2110 Quiz 1 - Part 4 Written Questions

Braden Lam

662056967

10/28/25

Question 1: Describe how HTML fits into the broader ecosystem of a website. Contrast the fundamental role of HTML with the primary roles of CSS and JavaScript. (5 points)

HTML is the foundation of every website and defines the content and structure. It tells the browser what elements exist on the page, like headings, paragraphs, images, links, and tables.

CSS is responsible for how the website looks. It controls colors, fonts, layouts, spacing, and visual design. CSS makes the website attractive and visually appealing.

JavaScript adds interactivity and behavior to websites. It allows pages to respond to user actions like clicks and form submissions, update content dynamically, and communicate with servers without reloading the page.

Question 2: Explain the difference between HTML structure and HTML semantics. Why is writing semantic HTML considered a best practice? Provide one example of a semantic HTML element and one example of a non-semantic element. (10 points)

HTML structure refers to how elements are organized and nested in a document. It's about the arrangement of tags and how they create the layout.

HTML semantics refers to the meaning that HTML elements convey. Semantic elements clearly describe what type of content they contain, making it easier for browsers, search engines, and assistive technologies to understand the page.

Writing semantic HTML is a best practice because it improves accessibility for users with disabilities, helps search engines better understand and rank content, and makes code easier to read and maintain for developers.

Example of a semantic element: <header> - This clearly indicates that its content is the header section of a page or section. Anyone reading the code immediately knows what this element represents.

Example of a non-semantic element: - This is a generic container that doesn't tell you anything about its content. It could contain anything and provides no meaningful information about what's inside.

Question 3: What is the "three-tier model" (also known as three-tier architecture) in web systems? Briefly describe the function and responsibility of each of the three tiers. (15 points)

The three-tier model is an architecture pattern that separates a web application into three distinct layers, each with specific responsibilities.

Tier 1: Presentation Tier

Tier 1 is the user interface that people interact with directly. It runs in the web browser and is built with HTML, CSS, and JavaScript. The presentation tier displays information to users and collects their input through forms and buttons. It only handles display and user interaction, not business logic or data storage.

Tier 2: Application Tier

This is the middle layer that contains all the business rules and logic. It processes requests from the presentation tier, performs calculations, validates data, and makes decisions about how the application should behave. It's typically built with server-side languages like Python or Java.

Tier 3: Data Tier

This layer manages all data storage and retrieval. It consists of database systems that store the application's data permanently. The data tier handles saving, retrieving, updating, and deleting information. It ensures data is consistent and secure. Only the application tier communicates with the data tier directly.

The benefit of this architecture is that each tier can be developed and updated independently without affecting the others.

Question 4: Explain what is meant by a Universal Interface in a REST API. (5 points)

A Universal Interface in REST means that all resources are accessed using the same standardized set of operations, regardless of what type of resource you're working with.

In REST APIs, this is achieved through standard HTTP methods: GET (retrieve data), POST (create new data), PUT (update existing data), and DELETE (remove data). These methods work the same way for every resource.

For example, whether you're getting user information or product information, you always use a GET request. To retrieve user 123, you use GET /users/123. To retrieve product 456, you use GET /products/456. The pattern is consistent and predictable, making REST APIs easy to understand and use once you learn the basic principles.

Question 5: Explain how your browser chooses which CSS rule to apply to a tag in the case where there are multiple rules that could apply. (15 points)

When multiple CSS rules could apply to the same element, browsers use a process called the cascade to determine which styles win. This involves three main factors:

1. Specificity

Specificity is a scoring system that determines how specific a CSS selector is. More specific selectors override less specific ones. The hierarchy from most to least specific is:

- Inline styles (written directly in HTML)
- ID selectors (#id)
- Class selectors (.class)
- Element selectors (div, p, h1)

2. Source Order

When two rules have the exact same specificity, the one that appears later in the CSS file wins. This is why it's called "cascading" style sheets. If you write:

```
p { color: blue; }

p { color: red; }
```

The text will be red because that rule comes last.

3. !important Declaration

Lastly, Adding !important to a CSS rule makes it override other rules, even if they're more specific.