Training Day 19 Report

Amrinder Singh
URN: 2302468   CRN: 2315013

10 July 2025

Topic: CSS Inheritance and Specificity

Today's session explored how CSS styles cascade, how properties are inherited by child elements, and how to resolve style conflicts using specificity rules. These concepts are critical for organizing large stylesheets and building scalable design systems.

Key Areas Covered

1. CSS Inheritance

- Inheritance allows certain properties (like `color`, `font`) to be passed from parent to child.
- Learned which properties are naturally inherited and which are not.
- Used `inherit`, `initial`, and `unset` values to manage inheritance behavior.

2. Specificity Rules

- CSS specificity determines which style rule takes precedence.
- Selector strength hierarchy: inline ¿ ID ¿ class ¿ element.
- Calculated specificity scores for various selector combinations.
- Identified conflicting styles and learned resolution strategies.

3. Overriding Styles

- Used `!important` sparingly to force style priorities.
- Organized styles using grouped classes and modular naming.
- Avoided overly complex selectors to keep styles readable.

4. Style Management Tips

- Maintain stylesheet order for logical overrides.
- Style resets to remove browser-default inconsistencies.
- Group selectors by layout role and functionality for clarity.

Hands-On Practice

Refined layout from previous session:

- Created a base style for typography using global selectors.
- Overrode specific elements (headings, buttons) with targeted classes.
- Debugged an issue where inline styles blocked external stylesheet rules.

Key Takeaways

- CSS inheritance saves repetition and simplifies styling flow.
- Specificity must be managed carefully to avoid unexpected results.
- Knowing which rules win ensures predictable, maintainable designs.
- Organized stylesheets lead to faster debugging and cleaner codebases.