**Q/A Summary of JavaScript and Web Development**

**CSS Media Queries**

* Q: What are media queries used for in responsive web design?
  + A: Media queries allow you to define different styles for a website based on the characteristics of the device it's being viewed on (screen width, orientation, etc.). This ensures an optimal viewing experience across various devices.
* Q: What are some examples of media queries used for mobile-friendly websites?
  + A: Examples include targeting specific screen widths for layout changes, optimizing image sizes, adjusting font sizes, and disabling features unsuitable for mobile.
* Q: What is an example of a poorly designed media query?
  + A: A media query targeting only iPads in portrait mode is bad practice. It's better to use wider screen width ranges and rely on CSS frameworks for a more comprehensive approach.
* Q: What does the viewport meta tag do in responsive design?
  + A: It sets the initial viewport size and scaling mode for the webpage content. This ensures it scales appropriately within the viewport (visible area of the browser window) on different devices.
* Q: How does the viewport meta tag improve user experience?
  + A: By setting the viewport appropriately, users don't need to zoom in or out manually to see the content properly on their device.

**CSS Reset and Normalization**

* Q: What is a CSS reset?
  + A: A CSS reset is a collection of rules that neutralizes default styles browsers apply to HTML elements, creating a blank slate for web developers.
* Q: Is a CSS reset mandatory for every website?
  + A: Not anymore. CSS normalization is a more common approach today, which adjusts browser defaults for consistency while preserving some useful styles.
* Q: What are the benefits of using CSS normalization?
  + A: It reduces inconsistencies across browsers, improves maintainability, and provides a predictable foundation for your website's design.
* Q: What are common bugs found in a browser's CSS defaults?
  + A: Inconsistent box models, font rendering discrepancies, margin and padding issues, and vendor prefixes. These can be addressed through a CSS reset or proper CSS normalization.
* Q: What are extra things I would need to add to my code after doing a CSS reset for website development?
  + A: Base styles, layout styles, component styles, typography styles, responsiveness, and animations and transitions need to be added.

**JavaScript Debugging**

* Q: What are common steps involved in debugging JavaScript code?
  + A: Identify the error, reproduce it, use browser developer tools (breakpoints, console logs, variable inspection), utilize debugging techniques (logical reasoning, rubber duck debugging), fix the error, and test to verify.
* Q: How can I use breakpoints in the browser?
  + A: Set breakpoints in the developer tools at specific lines of code. When execution reaches that line, the code pauses, allowing you to inspect variables and the call stack.
* Q: Can I implement breakpoints directly in JavaScript code?
  + A: No, but you can use the debugger statement to pause execution at a specific line or use console logs to track code flow and identify errors.
* Q: What are useful techniques to achieve code debugging in JavaScript?
  + A: Leverage browser developer tools (breakpoints, console logs), analyze error messages, use logical reasoning and debugging techniques, consider using debugging tools and extensions, test thoroughly, and isolate the problem.
* Q: What are the best ways to avoid bugs when coding with JavaScript?
  + A: Write clean code (meaningful names, proper formatting), handle errors (validation, try...catch), write unit tests, test comprehensively, use linters and code formatters, leverage libraries, write modular code, and use version control.