

EasyRoute Style Rules

JavaScript Style Rules

Naming Conventions:

Use camelCase for identifiers (variables and functions).

Use PascalCase for class names.

Prefix boolean variables with is, has, can, etc. (e.g., isVisible, hasEntries).

Use clear, descriptive names; avoid abbreviations.

Syntax:

Use semicolons at the end of statements.

Prefer template literals over concatenation when dealing with variables.

Use arrow functions for anonymous functions when context is suitable.

Spacing and Indentation:

Use 2 spaces for indentation.

Add spaces around operators (e.g., x + y).

Place an empty line before return statements, unless it's the only statement in the block.

Brackets:

Use curly braces with all multi-line blocks.

Place opening braces on the same line as the statement (if, function, etc.) with one space before the opening brace.

Best Practices:

Use const for declaring variables that will not be re-assigned, and let for variables that will.

Avoid using var to declare variables.

Use array methods like map, filter, and reduce over loops when it enhances clarity.

Keep functions short and focused; each should do one thing only.

CSS Style Rules

Naming Conventions:

Use kebab-case for class names (e.g., .main-container).

Use meaningful or generic class names that reflect the purpose of the element rather than its content.

Syntax:

Use a space after the colon in property declarations.

End all declarations with a semicolon.

Group related CSS rules and separate groups with a blank line.

Organization:

Organize sections of your stylesheet logically (e.g., reset, typography, layout, components).

Use comments to define sections.

Selectors:

Keep selectors short and avoid excessive specificity.

Prefer class selectors over tag selectors for maintainability and performance.

HTML Style Rules

Syntax:

Use lowercase for element names, attributes, and values (unless text/CDATA).

Use double quotes around attribute values.

Include a doctype declaration and character encoding (typically UTF-8).

Indentation and Spacing:

Use 2 spaces for indentation.

Semantics:

Use HTML5 semantic elements (like <header>, <footer>, <article>, <section>) appropriately.

Use <nav> for navigation blocks and <main> for the primary content.

Accessibility:

Ensure that forms are accessible with properly associated <label> tags.