

Top 20 VS Code Extensions Every Developer Must Use

Subtitle: Boost your productivity, code quality, and debugging speed with these essential VS Code extensions.

Website Name: haas.dev

Website Link: <https://dev-roast-app.vercel.app>

Introduction

VS Code is the most popular code editor for developers, but its **real power comes from extensions**. This guide introduces 20 essential extensions that every beginner and intermediate developer should use to **write better code, debug faster, and save time**.

Step 1: Productivity & Code Formatting Extensions

1. Prettier – Code Formatter

- Automatically formats code on save
- Supports JS, HTML, CSS, Python, etc.
- Mini tip: Use `.prettierrc` to customize rules

2. Bracket Pair Colorizer 2

- Highlights matching brackets with colors
- Makes reading nested code easier

3. Path Intellisense

- Auto-completes file paths in imports

4. Auto Rename Tag

- Automatically renames HTML/JSX paired tags

5. Trailing Spaces

- Highlights unnecessary trailing spaces
-

Step 2: Debugging & Testing Extensions

6. Debugger for Chrome

6. **Debugger for Chrome**

- Debug JavaScript in Chrome directly from VS Code

7. **ESLint**

- Detects coding errors and enforces style rules
- Works well with Prettier

8. **REST Client**

- Test APIs inside VS Code without Postman
- Send GET/POST requests and view responses

9. **Code Runner**

- Run code snippets in multiple languages quickly

10. **Live Server**

- Launch local development server with live reload for HTML/JS projects
-

Step 3: Git & Version Control Extensions

11. **GitLens**

- Shows git blame info inline
- View commit history for each line of code

12. **GitHub Pull Requests and Issues**

- Manage PRs and issues directly from VS Code

13. **Git History**

- Visualize branch and commit history
-

Step 4: Snippets & Templates

14. **JavaScript (ES6) Code Snippets**

- Predefined JS/React code snippets for faster coding

15. **HTML CSS Support**

- Autocomplete for HTML classes and IDs from CSS files

16. **React Developer Tools**

16. React Developer Tools

- React-specific snippets and IntelliSense

17. Python (if applicable)

- Linting, IntelliSense, and snippets for Python projects
-

Step 5: UI / Theme / Accessibility Extensions

18. Material Icon Theme

- Shows icons for different file types and folders

19. One Dark Pro

- Popular dark theme for coding comfort

20. Peacock

- Color-code VS Code workspaces for easy project differentiation
-

Step 6: Bonus Tips for Using Extensions

- Use **Workspace-specific extensions** to avoid overload
 - Disable extensions you don't use regularly to save memory
 - Learn keyboard shortcuts for extension commands
-

Step 7: Mini Project Idea

- Setup a new web project with HTML, CSS, and JS
 - Install **Live Server**, **Prettier**, **Path Intellisense**, and **ESLint**
 - Try importing a module from another folder using **Path Intellisense**
 - Debug a small JS bug using **Debugger for Chrome**
-

Step 8: Key Takeaways

- Extensions improve coding efficiency, reduce errors, and save time
- Focus on **formatting, debugging, Git, snippets, and UI**
- Customize VS Code workflow to your projects
- Practice using extensions while building mini-projects to see real benefit

- Practice using extensions while building mini-projects to see real benefit
-

Visit **haas.dev** for step-by-step guides, recommended tools, and beginner-friendly developer resources.

Website Name: haas.dev

Website Link: <https://dev-roast-app.vercel.app>
