

Top 10 Developer tools you can't live without

In a developer's toolkit, having the right tools can significantly boost productivity, **improve code quality, and streamline workflows.**



Here's a list of the top 10 developer tools every programmer should know and use in 2025



Tool 1

Visual Studio Code



Why use

Purpose: A lightweight yet powerful code editor with extensive plugin support.

Features:

- IntelliSense for smart completions.
- Integrated Git support.
- Debugging and live previews.

Facts:

- Over 50% of developers use VS Code (Stack Overflow Developer Survey 2023).
- Supports Python, JavaScript, TypeScript, and more.

Pros

Free, open-source

Highly customizable

Cons

Can slow down with too many plugins

Not ideal for large-scale IDE needs

Fun fact: VS Code's marketplace has over 38,000 extensions for customization.



Tool 2

GitHub



Why use

Purpose: A platform for version control and collaboration.

Features:

- GitHub Actions for CI/CD pipelines.
- Pull request reviews and discussions.
- GitHub Copilot for AI-assisted coding.

Facts:

- Hosts over **83 million** repositories globally.
- Integrates seamlessly with tools like Jira, Slack, and VS Code.

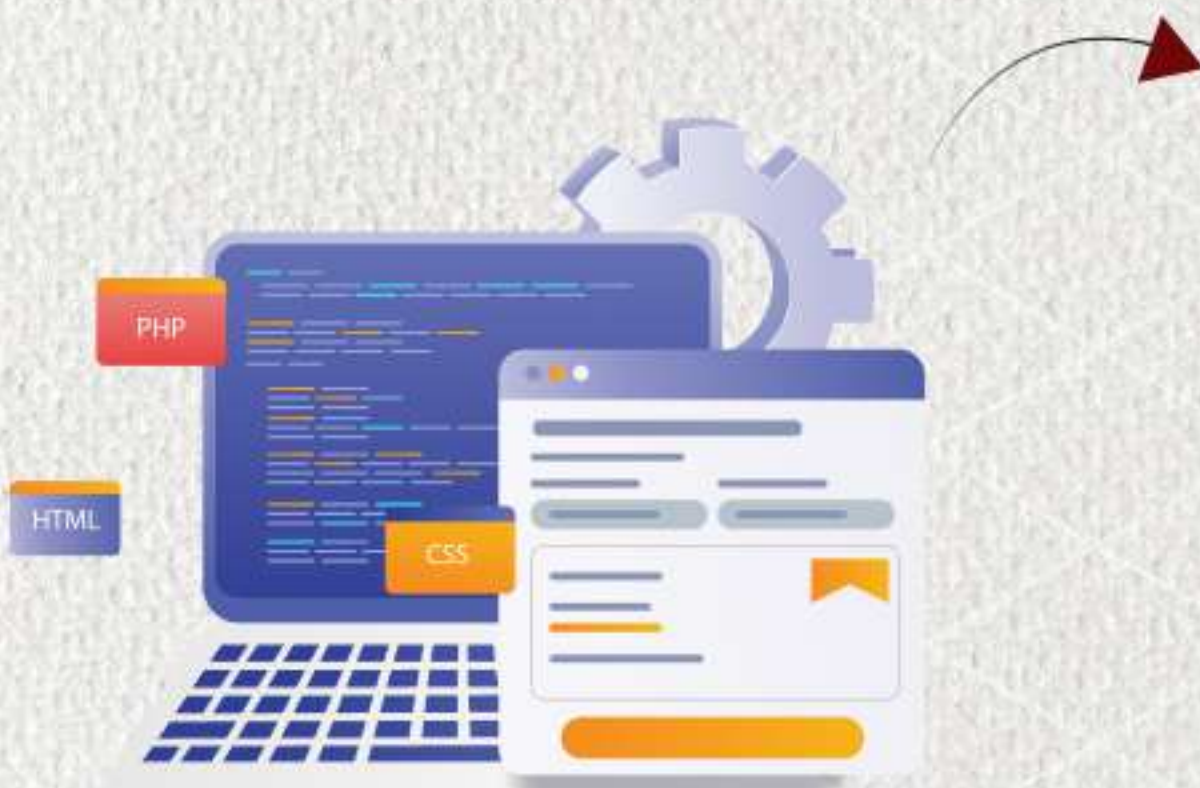
Pros	Cons
Great for collaboration	Some features require paid plans
Strong community support	Dependency on internet connection

Did you know? GitHub sponsors thousands of open-source developers worldwide.



Pros	Cons
Lightweight and portable	Steep learning curve for beginners
Cross-platform support	Resource-heavy for large images

Big player alert: Companies like **Netflix** and **Airbnb** rely heavily on Docker.



Tool 4

Git

Why use

Purpose: A distributed version control system for managing source code.

Features:

- Track changes and revert to previous versions.
- Collaborative branching and merging.
- Supports multiple workflows (e.g., Gitflow).

Facts:

- Created by Linus Torvalds, the creator of Linux.
- Git is used by **87% of developers globally.**

Pros	Cons
Free and open-source	Can be complex for large repositories
Supports offline work	Merge conflicts can be tricky

Pro tip: Master Git commands like rebase, stash, and cherry-pick for advanced control.



Tool 5

Postman

Why use

Purpose: API testing and development made simple.

Features:

- Build and test REST and GraphQL APIs.
- Automate API testing with collections.
- Monitor API performance.

Facts:

- Used by **25 million developers** worldwide.
- Allows exporting API tests as scripts for CI/CD.

Pros	Cons
Intuitive interface	Limited features in free plan
Supports automation testing	Can feel slow for large-scale projects

Fun fact: Postman's API Network includes thousands of public APIs for learning and testing.



Tool 6

Jenkins



Why use

Purpose: Automate build and deployment pipelines with CI/CD.

Features:

- Extensive plugin ecosystem.
- Configurable with Groovy scripts.
- Scalable for large teams and projects.

Facts:

- Jenkins supports over 1,800 plugins.
- Open-source and free to use.

Tool 7



Why use

Purpose: Collaborative design and prototyping for UI/UX projects.

Features:

- Real-time collaboration for teams.
- Easy integration with design systems.
- Plugin support for added functionality.

Facts:

- Used by top companies like Google, Uber, and Microsoft.
- Preferred tool for front-end developers and designers.

Tool 8

IntelliJ IDEA



Why use

Purpose: IDE optimized for Java and other JVM languages.

Features:

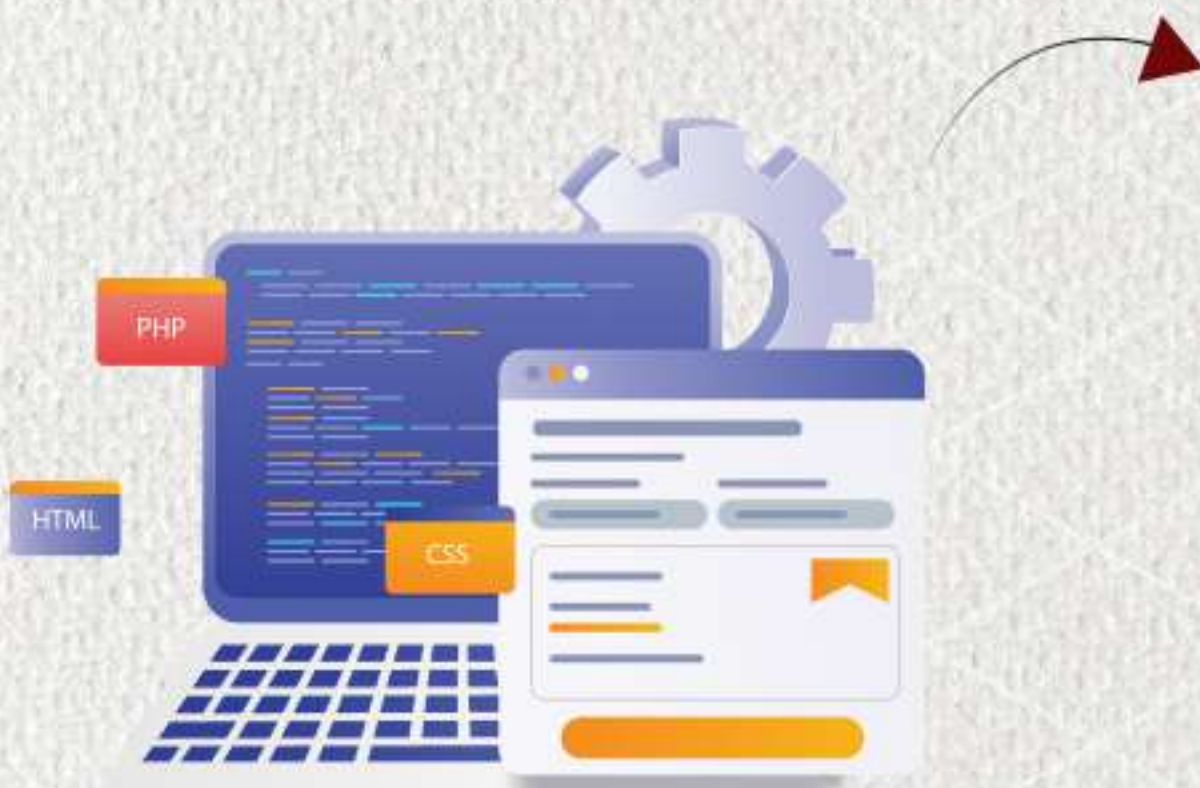
- Advanced code refactoring tools.
- Built-in support for Spring, Hibernate, and Kotlin.
- Smart debugging and profiling tools.

Facts:

- Used by **70% of Java developers** (JetBrains Survey 2024).
- Available as both free (Community) and paid (Ultimate) editions.

Pros	Cons
Feature-rich for Java dev	Can feel heavy on resources
Strong plugin ecosystem	Paid version is expensive

Big user: IntelliJ IDEA powers development at Google and Spotify.



Tool 9



Why use

Purpose: Team communication and collaboration.

Features:

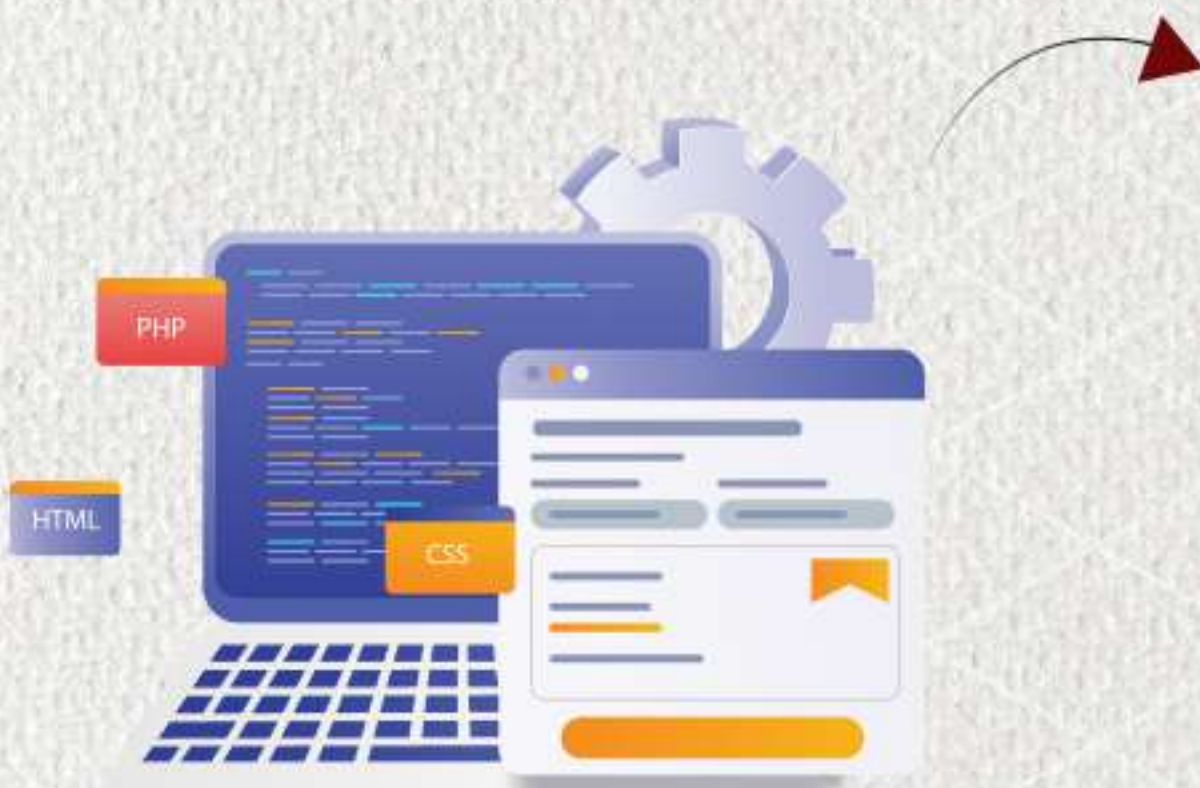
- Integration with GitHub, Jira, Jenkins, and more.
- Supports threaded discussions and searchable history.
- Slack bots for automation.

Facts:

- Over **12 million** daily active users.
- Popular among remote and distributed teams.

Pros	Cons
Great for team updates	Expensive for large teams
Integrates with dev tools	Can lead to notification fatigue

Fun fact: Slack was originally designed as an internal tool for game developers.



Tool 10

Notion

The Notion logo, which consists of a black cube with a white letter 'N' on its front face.

Why use

Purpose: An all-in-one workspace for project management, documentation, and team collaboration.

Features:

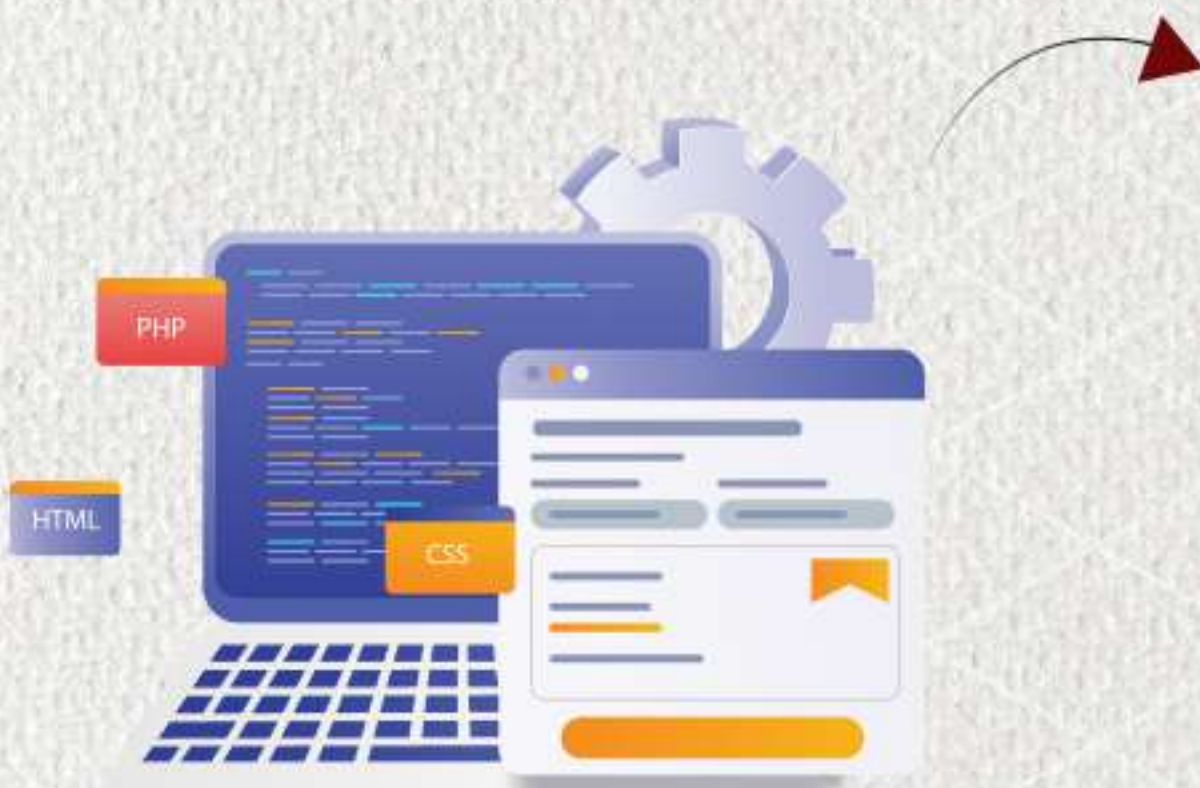
- Kanban boards, task tracking, and calendars.
- Supports markdown for documentation.
- Easily shareable templates.

Facts:

- Used by **20 million+** users globally.
- Popular for managing personal and team productivity.

Pros	Cons
Highly versatile	Free plan has limitations
Easy to set up	Lacks advanced PM features

Interesting trend: Many developers use Notion for personal coding wikis



Conclusion

These top 10 tools are essential for boosting your productivity, improving collaboration, and streamlining your workflows in 2025. Whether you're a front-end developer, a back-end expert, or a DevOps engineer, these tools have something for everyone.

Which of these tools do you use the most? Comment below and share your favorites!

Upskill with



India's most trusted

Program For **Working Professional**

Data Structure Algorithms & System Design


With **Gen-AI** For Software Developers

Program electives:

 GenAI

 Product management

 DevOps

 Data engineering



Get Certification from :



WOLFF UNIVERSITY

