Key Term

Visual Studio Code - A free, open-source, lightweight code editor with support for debugging, version control, and extensions. It can be used to develop web, desktop, and cloud applications across many languages.

Extensions - Add-on tools and features that allow you to customize and enhance Visual Studio Code for your development needs. Some common extensions include linters, debuggers, snippets, and language support.

Debugging - The process of detecting and removing errors from computer hardware or software. Visual Studio Code has built-in debugging support, allowing you to set breakpoints, step through code, and inspect variables.

Version Control - The management of changes to code over time. Visual Studio Code integrates with Git and other version control systems to track revisions, review history, and collaborate with teams.

Workspaces - A Visual Studio Code concept that allows you to group one or more folders that contain code. Useful for organizing multi-file and multi-language projects into a single editable space.

Visual Studio Code

the place that → can write code

has features beyond just writing text files
⌐ Debug Code , run code

↗△ It's not similarly to VIM that be text editor.

- IDE (Integrated Development Environment)
- Code completion → *help with editing code*
- Debugging
- Built-in support JavaScript, TypeScript, Node.js
- Plugins for other languages

This lesson provided an overview of Visual Studio Code, including what it is, key features, and how to get started using it for development. Some top takeaways are:

- Visual Studio Code is a free, customizable code editor for web/cloud development
- It offers built-in support for debugging code, version control, extensions, and more
- You can customize VS Code to your needs via themes, keyboard shortcuts, and extensions
- VS Code uses workspaces to organize multi-file and multi-language projects
- The Explorer, Search, Source Control and other panels help navigate and manage code

Reflection Questions:

1. What are some benefits of using Visual Studio Code over another code editor?
2. What are extensions and how can they help improve your workflow?
3. How might you setup VS Code to best suit your personal coding preferences?
4. What are some key debugging features offered in Visual Studio Code?
5. How can you use workspaces to organize large, complex projects in VS Code?

Challenges:

1. Install Visual Studio Code and test out the basic features covered in this lesson.
2. Configure your own user settings, keyboard shortcuts, and color theme in VS Code.
3. Browse and install 1-2 extensions that are useful for your development needs.
4. Practice debugging a simple web or Rust application in VS Code.
5. Create a multi-language workspace in VS Code for a full-stack project.