Shortcut Commands in Visual Studio Code

- Open a folder
  - **File** > **Open Folder** (`Ctrl+K Ctrl+O`)
- File Explorer
  - **View** > **Explorer** (`Ctrl+Shift+E`)
- Search view
  - **View** > **Search** (`Ctrl+Shift+F`)
- Source Control
  - **View** > **Source Control (SCM)** (`Ctrl+Shift+G`)
- Run and Debug
  - **View** > **Run** (`Ctrl+Shift+D`)
- Extensions view
  - **View** > **Extensions** (`Ctrl+Shift+X`)
- Open the Command Palette(Find features and keyboard shortcuts)
  - **View** > **Command Palette...** (`Ctrl+Shift+P`)
- Output panel
  - **View** > **Output** (`Ctrl+Shift+U`)
- Debug Console
  - **View** > **Debug Console** (`Ctrl+Shift+Y`)
- Problems panel
  - **View** > **Problems** (`Ctrl+Shift+M`)
- Terminal
  - **View** > **Terminal** (`Ctrl+\``) ← Display the Integrated Terminal.
  - **Split Terminal** (`Ctrl+Shift+5`) ← Split the terminal.
  - **Create New Terminal** (`Ctrl+Shift+\``) ← Create new terminal.
- Create a new file
  - **File** > **New File** (`Ctrl+N`)
- Save a file
  - **File** > **Save** (`Ctrl+S`)
- Auto Save
  - **File** > **Auto Save**
- Run
  - **Run** > **Start Debugging** (`F5`)
- Programming language extensions
  - Python - IntelliSense, linting, debugging, code formatting, refactoring, and more.
  - Live Preview - Hosts a local server to preview your webpages.
- Zoom
  - Zoom out (`Ctrl+-`)
  - Zoom in (`Ctrl+=`)
- Customize your editor with color themes.
  - **File** > **Preferences** > **Theme** > **Color Theme** (`Ctrl+K Ctrl+T`)

- Automatically format the source code.
  - **Format Document** command (`Shift+Alt+F`)
- Quick Open recent files or search by filename
  - `Ctrl+P` to show the Quick Open dropdown
- Go to Line in a file
  - type `filename:line number`
- Go to Symbol in a file
  - type `filename@symbol name`
- View Quick Open options
  - type `?`
- Quick Open multiple files
  - press `Right Arrow` to open the selected file but leave the dropdown available
- Multi-cursor selection
  - `Alt+Click` to add a new cursor
  - `Ctrl+Alt+Up` `Ctrl+Alt+Down` to add a new cursor above or below the current position
  - `Ctrl+Shift+L` to add cursors to all matches of the current selection
- Auto Save changes
  - **File** > **Auto Save**
- Open the Settings editor
  - **File** > **Preferences** > **Settings** (`Ctrl+,`)
- Set Format On Type
  - check **Editor: Format on Type**
- Set Format On Paste
  - check **Editor: Format on Paste**
- IntelliSense smart code completions
  - trigger IntelliSense with `Ctrl+Space`
- Install a new Color Theme from the VS Code Extension Marketplace.
  - Change your Color Theme
  - Change your File Icon Theme
- Find and Install extensions from the VS Code Extension Marketplace.
  - Look at all extensions without specifying anything in search
  - Type `@` and look at the most popular ones sorted by installs
  - Type `@category` and here you can see specific extensions for virtually anything
  - Looking at all extensions that fall in the category of themes by typing `category:themes` in search
  - See what features are added via the **Features Contributions** tab or Command Palette (`Ctrl+Shift+P`).
  - See recommendations for other extensions.

Keyboard Shortcuts

- To see all shortcut commands, go to file -> preferences -> click Keyboard Shortcuts. Here you can use search for specific command you are looking.
- Another way, go to command palette and type and select "Preferences: Open Keyboard Shortcuts (JSON)", we can see list of shortcuts.

- Find a keyboard shortcut in the Command Palette
- Learn about common keyboard shortcuts
- Change a keyboard shortcut using the keyboard shortcut editor
- Use a JSON file to customize keyboard shortcuts
- Learn about keyboard shortcut extensions
- Customize your settings using the settings editor
- Use a JSON file to customize settings
- Configure language specific settings

How to use Git version Control in VS Code

Source control, version control, backing up whatever you choose to call it. It's all about keeping track of code and file overtime.

- Git is a program for managing source code
- GitHub hosts a copy of that code and repository online.
- Visual Studio Code makes the process of using Git and GitHub easy.

Visual Studio Code with Git:

- Step 1: Initialization
  - Initialize a folder to reflect your repository.
    - Open up an existing folder with some files in it.
  - Then go to source control( ) (`Ctrl+Shift+G`) and click "initialize repository".
  - Now notice at the bottom left-hand corner that the branch is named main. But if we need to rename it, we can simply open the command palette (`Ctrl+Shift+P`) and type rename and select "Git:Rename Branch" from the options and type the name you want to call(it will reflect in below(from name main to new name)) but let it be as main/master which is familiar for initial.
- Step 2: Stage or add files to the repository.
  - Now after initialization, we'll see that in the source control panel that any new file shows up with the letter 'U' beside it and 'U' stands for untracked file, meaning that a file that is new or changed but has not been added to the repository. So to track it, just click the plus(+) sign next to the file and now it is what referred to as stage. Now once added, the letter next to the file will change to 'A' which represents a new file has been added to the repository.
- Step 3: Commit and publish
  - Next step would be to commit it. All we need to do is add a message such as 'First Commit'. And in general, it will just describe whatever changes that we make. It's for informational purposes. And to submit it, just hit the check/tick mark right above it.
  - Now, we're on our main branch, but if this is a real application and we need to add new code, new features, for example: we want to create a separate branch to do

that by going to the command palette. In command palette, we will see the option here to create branch or we can just start typing in 'create branch'. And give a name like 'New Features'. And the bottom left hand corner reflects that now.

- After committing, if we add 2 lines of code that will reflects in branch('New Features'). One for warning and one for danger. Green line for adding new line, shaded blue line for changes in line, and red arrow for deleting a line(we can put it back).
- Now, right under the word changes, we'll see the file shows up with the letter 'M' in the source control panel. We can see some icons here to open the file, discard changes (undo), stage changes.
- Now, Visual Studio Code has the ability to perform a diff on files to compare files side-by-side. Typically we would need to have a separate tool to do this, but it's built in and to see our changes side-by-side, we just need to click on our file. But if we want to see all the changes in one file, go to the upper right hand corner three dots and choose inline view, it consolidates them. And within here, we can also make changes so we add some if we wanted to. Now stage the changes by clicking the plus(+) sign and a message and commit it.
- Now remember that, at the bottom left-hand corner it still in a branch(here, 'New Features') and that's the only branch that's affected so switch to main branch by clicking main in command palette(like creating new branch). And in the main, we can see the original code i.e., without changes. Now, if we want the new code to be reflected in the main branch, we need to merge by going to three dots in source control panel and select Branch and then select Merge Branch and then the command palette will open, from there select out branch 'New Features' and now our main branch has the reflected changes. So now we actually want to publish this to GitHub. All we need to do is go to "Publish Branch". It's going to prompt us that some extension GitHub wants to sign in using GitHub. We'll hit allow and then an authentication process will start. We'll hit continue at "github.login prompt in website. And over there, give permission for GitHub to open up Visual Studio Code. And now that will ask us for the permission for this extension to open URL. So we will click open. And now we can select whether we want a public or private repository in a palette. After selecting public/private repository, it begins the publishing process to GitHub and then prompts us to open up and there we could see the file that we've been working on.

- Step 4: Clone
  - If we want to clone the repo, copy the repo's HTTPS URL and open up the command Palette, and enter 'clone' and select 'Git:Clone' and paste the URL and clone from this URL. And now we need to do is specify the location where we wanted clone repo. After selecting the location as the repository location, we will see a notification prompt at right-hand bottom corner in VS Code by asking "Would you like to open the clone repository?" and we will say yes. That's it.

Most Common ways developers use Git:

- Creating a repository and branch
- Staging and Committing changes
- Pushing code up and merging branches
- Cloning existing projects/repos.

Summary to use Git version Control in VS Code:

- Install Git
  - https://git-scm.com
- Open Folder
  - **File** > **Open Folder** (`Ctrl+K Ctrl+O`)
- Source Control
  - **View** > **Source Control (SCM)** (`Ctrl+Shift+G`)
- Initialize repository
  - `main` is the default branch
- Open the Command Palette.
  - **View** > **Command Palette** (`Ctrl+Shift+P`)
- Rename a branch
  - **Git: Rename Branch**
- File version control status
  - **U** - Untracked file
  - **A** - Added file
  - **M** - Modified file
- Commit file
  - **Commit** ☑ (check mark) button
- Create a branch
  - **Git: Create Branch**
- Diff editor
  - **Inline View** button
- Stage changes
  - **Stage Changes** ➕ button
- Switch branches
  - Status bar branch item (lower left)
- Merge branch
  - **Views and More Actions** (**...**) > **Branch** > **Merge Branch**
- Publish branch to GitHub
- Clone repository
  - **Git: Clone** > **Clone from URL**