25-Jul-2023 RCW
vscode variables per ChatGPT:

1. ${workspaceFolder}: The absolute path of the folder opened in Visual Studio Code.
2. ${workspaceFolderBasename}: The name of the workspace folder opened in Visual Studio Code (the last segment of the workspace folder path).
3. ${file}: The current opened file.
4. ${fileWorkspaceFolder}: The absolute path of the folder containing the currently opened file.
5. ${fileBasename}: The base name of the currently opened file (i.e., without the directory path).
6. ${fileBasenameNoExtension}: The base name of the currently opened file without the file extension.
7. ${fileDirname}: The directory path of the currently opened file.
8. ${fileExtname}: The extension of the currently opened file (including the dot ".").
9. ${cwd}: The current working directory (i.e., the directory from which Code was started).
10. ${env:VARIABLE_NAME}: Access environment variables, where VARIABLE_NAME is the name of the environment variable

Task properties:

label: (required): The display name of the task. It is used to identify the task in the Visual Studio Code user interface.

type: (required): The task type. It specifies the task's execution provider. Common types include:

"shell": Execute a shell command.
"process": Run an external executable as the task.
"npm": Execute an npm script.
"grunt": Execute a Grunt task.
"gulp": Execute a Gulp task.
"msbuild": Run MSBuild tasks (C# projects).

command: (required): The command to execute for the task. This can be a shell command, an executable name, or a script.

args: An array of arguments to pass to the command or script specified by the command property.

options: Additional options for task execution, like setting the current working directory or environment variables.

group: An optional grouping object that can have the following properties:

kind: The kind of group (e.g., "build", "test").
isDefault: Specifies whether the task is the default task for the group.
presentation: Defines the task presentation in the user interface. It can have properties like:

reveal: Controls how the output is shown (e.g., "always", "silent", "never").

echo: Controls how the command is displayed in the output channel (e.g., "always", "silent", "never").

problemMatcher: Specifies the problem matcher to use to capture errors or warnings from the task output. This helps to navigate between errors in the source code and the task's output.

dependsOn: An optional array of task labels on which this task depends. This allows defining task dependencies and ensures that dependent tasks are executed before the current task.

isBackground: Specifies whether the task runs in the background. If true, the task runs as a background task, and its output is not shown in the terminal.

runOptions: Provides options that control how the task is run, such as running the task in the terminal or running it silently.