**SUBLIME TEXT**

**Multiple Cursors**

* Used to make changes at multiple scattered coordinates.
* **Ctrl + (Click at required locations)**

**Selecting Occurrence**

* It is used to replace a word at multiple occurrences.
* Press Ctrl + D to select the word.

**Column Selector**

* Put cursor at place, and press **Ctrl + Alt + (Up/Down arrow)**.

**Word Wrap**

* Used generally in **HTML** web development.
* Press **Alt + Shift + W** to word wrap around the cursor’s selected area.

**Move Line Vertically**

* Press **Ctrl + Shift + (Up/Down Arrow)**

**Line Manipulation**

**\*Reference to line containing the** **cursor\***

* Create same line below: **Ctrl + Shift + D** (D = duplicate)
* Delete whole line: **Ctrl + Shift + K** (K = Kill)
* Indent line to left: **Ctrl + [**
* Indent line to right: **Ctrl + ]**

**\*With reference to the selected part\***

* Edit -> Line -> Reindent

**Project Manager**

* Just **drag and drop the folder** to your screen in Sublime Text.
* To open a file in new tab, **double click** on it.
* To close, right click on folder opened and choose remove from project.
* Press **Ctrl + P** to open search bar for finding files:
  + Write **file\_name@function\_name** to highlight the particular function.
  + Write **file\_name:line\_number** to jump to a specific line.
  + Write **@** and toggle through arrows to different functions.

**Speedy Tips**

* For seeing unsaved changes:
  + **Right click** anywhere on screen.
  + Select **“Show Unsaved Changes…”**
  + Press **Esc** to close the terminal.

**Saving Sublime Project**

* **For saving:**
  + Select **Project** from head bar.
  + Now save the file with extension of **.sublime-project**
* **For accessing saved project:**
  + Goto **Project**.
  + Select either **Open Recent**, if the save was made recently.
  + Or we can use **Open Project** and toggle through.

**Command Palette**

* Shortcut: **Ctrl + Shift + P**
* Used for giving command to editor.
* You can change **extension** from **lower right corner**.

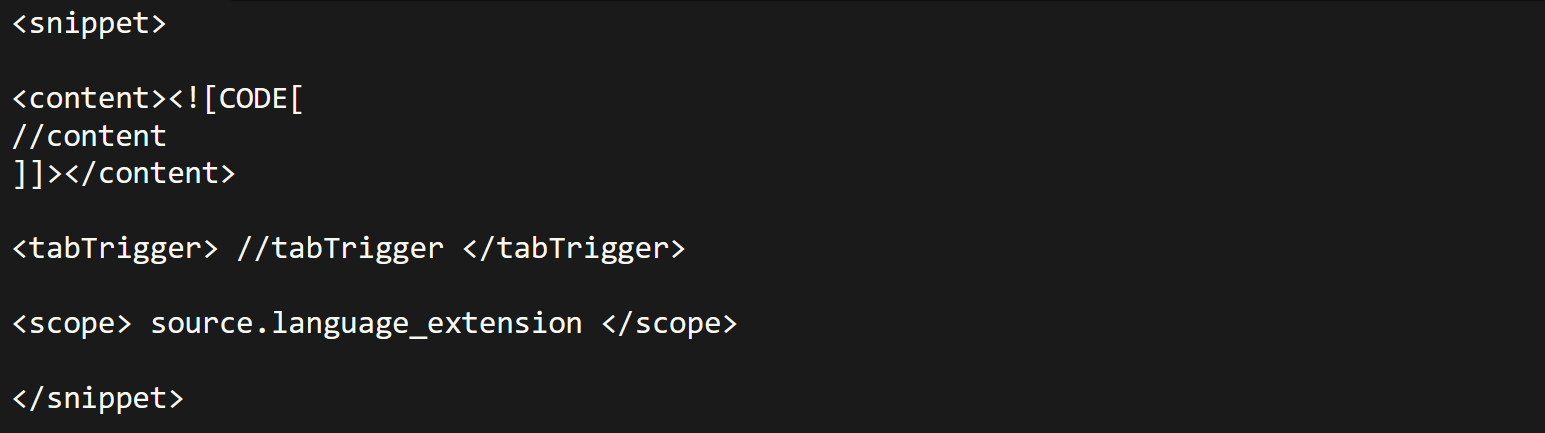
**Package Management**

* Write **“install”** in command palette for choosing package for installation.
* Some extra package features can be accessed in **tools**.

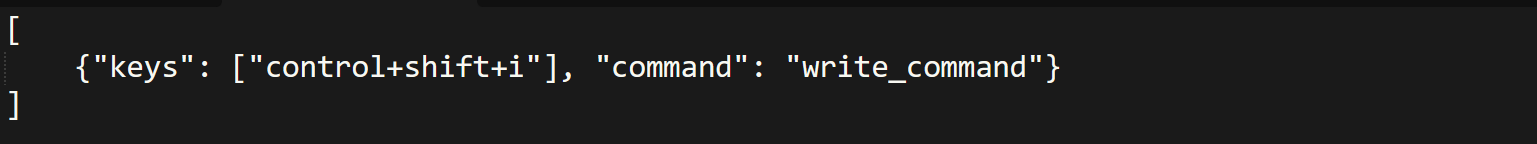
**Snippets**

* Shortcut words to create code blocks with less effort.
* How to know snippets? **Tools -> Snippets…**
* Extension: **sublime-snippet**

**Custom Snippet Creation**



**Key Bindings**



* You can find the command by searching in console.
* To open console: **View -> Show console**
* Extension: **sublime-keymap**

**Macro**

* Similar to snippets.
* **Ctrl + Q** to start recording macro.
* Stopping macro: **Tools -> Save macro…**
* Extension: **sublime-macro**
* Choosing macro: **Tools -> Macros -> (…)**



* Add this, where **“run\_macro\_file”** is command.

**Colour Themes**

* Theme names can be searched at the **command palette**.
* **“Themr”** package can be handy for those who continuously switch themes.
* You can add/customize more at **Preference.sublime-settings**

**Split Layout**

* Goto: **View -> Layout -> (choose)**
* Now you can drag and drop tabs into various window pans.

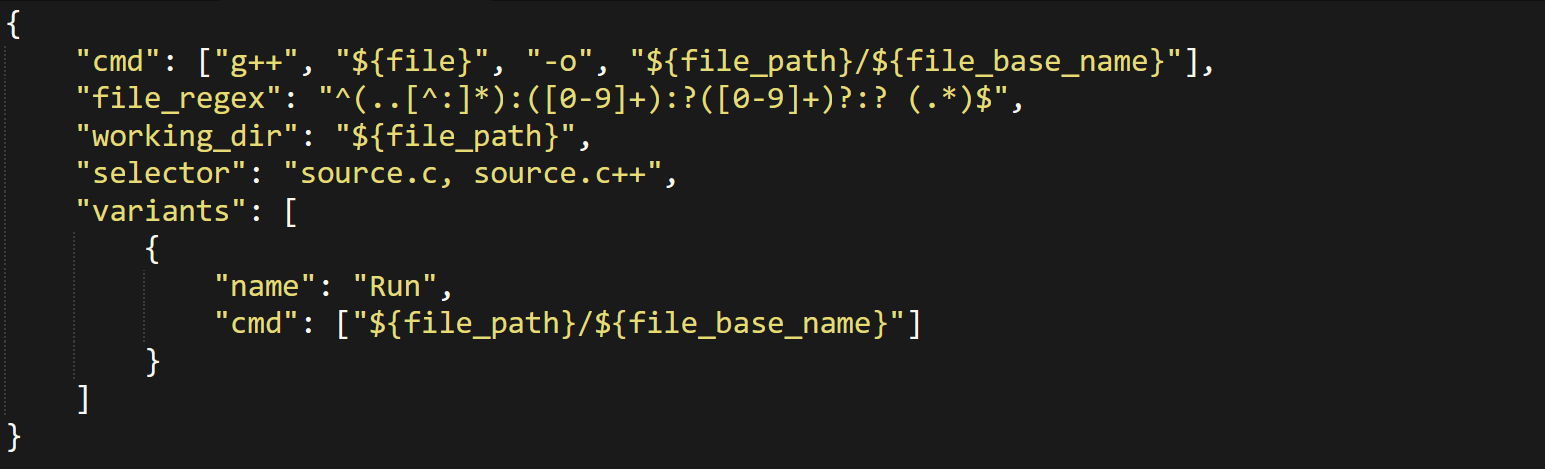
**Creating Build File**

* Kept as **key-value** pairs.
* Are **JSON** files with extension of ***.sublime-build***.
* **JSON:** Java Script Object Notation
* JSON are used as they are **lightweight**, **human-readable** & **easy to parse**.
* Source directory of respective language must have been added to environment variable first.

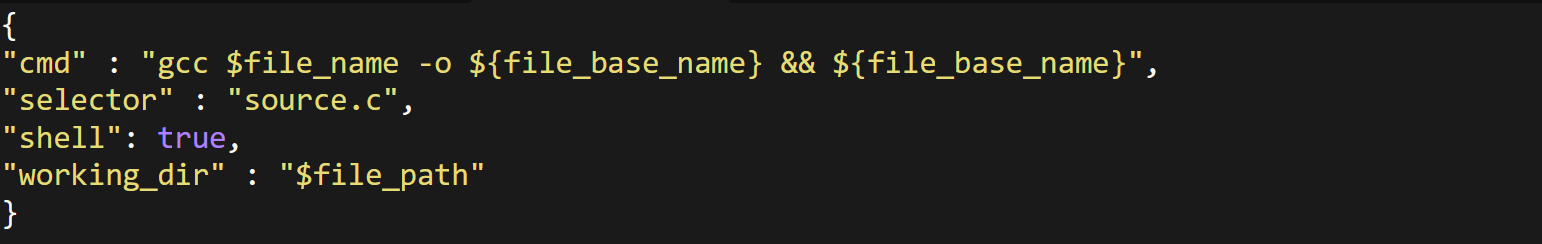
Common keys used:-

* ***cmd*** – Everything marked with ***$*** get replaced with actual asset.
* ***file\_regex*** – Helps Sublime Text identify lines with errors & warnings.
* ***working\_dir*** – In this, ***${file\_path}*** refers to path of running file.
* ***selector*** – Tell for what kind of files the build is.
* ***variants*** – Tells which utility must be used for what kind of job.
* ***shell*** – Optional, but set it to true if adding.

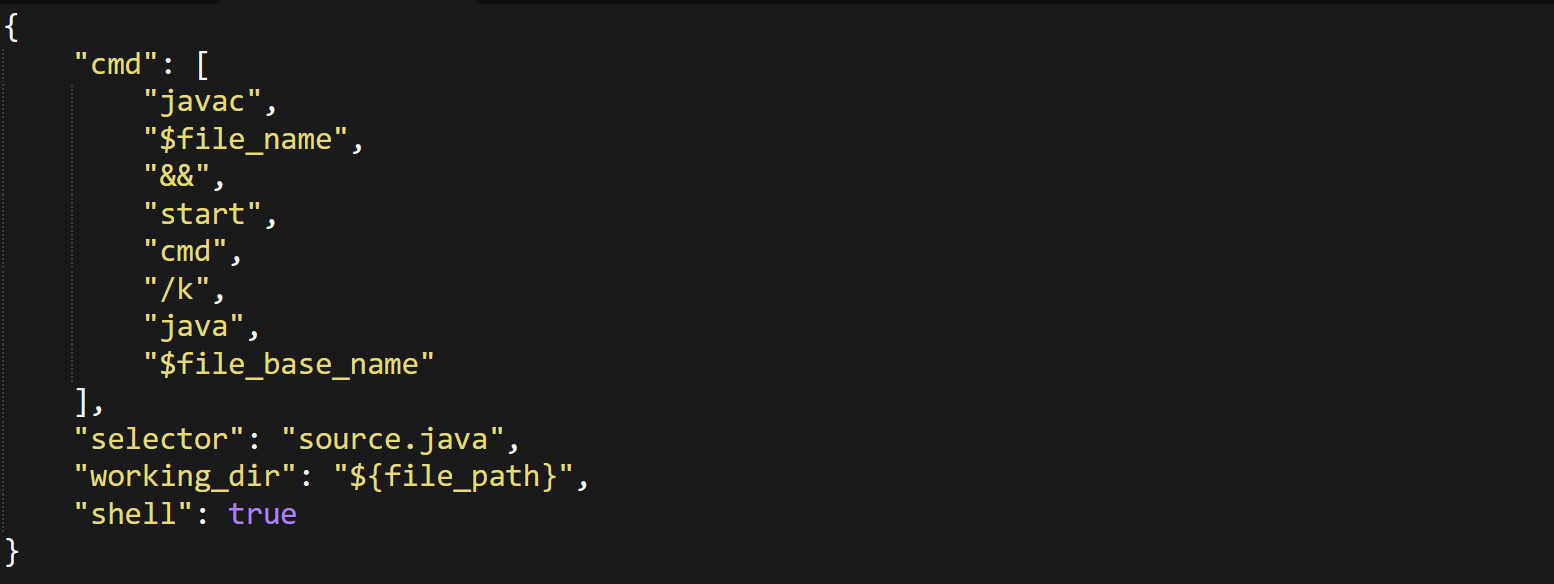
Sample sheet:-



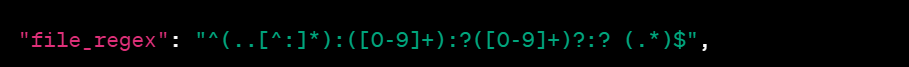
Basic C build file:-



Basic Java JDK build file:-



**Breakdown of file\_regex**



* **^** sticks the code reader to **beginning** of a line.
* **..[^:]\*** reads **first two characters** of line & catches if anything other than colon is found. Commonly referring to **name** of the file.
* **:([0-9]+)** reads digits **immediate** to **colon (:)**. Commonly referring to **line number**.
* **:?([0-9]+)?:?** reads if any second set of such colon & digits are there. Commonly referring to **column number**.
* **(.\*)$** reads rest of the line to **catch errors & warnings** if any.

For example:-

