

# Running Petite Chez Scheme in Sublime Text 3

Posted to Piazza in Spring, 2016 by Austin Derrow-Pinion

Being a fan of sublime text, I went ahead and found a good way of programming Petite Chez Scheme in Sublime Text:

1. If you don't already have "Package Control" installed, I don't know why you are using Sublime Text, and should probably do that soon. [Here is a link](#) for instructions if you want it.
2. Once Package Control is installed, go to "Preferences -> Package Control". Then select "Package Control: Install Package". Type "Scheme" and install the package.
3. Once that is installed, by doing the same process, install the "SublimeREPL" package.
4. If you didn't select to add Petite Chez Scheme to your PATH environment variables during installation, make sure you do that so Sublime knows it exists.
5. If you reset Sublime, you can then go to "Tools -> SublimeREPL -> Scheme -> Petite Chez Scheme". This will open up a REPL window within Sublime so that you can program that way.
6. To build a program, you will have to make a new build system by going to "Tools -> Build System -> New Build System...". This one I have made here will give you two options when building. Both builds will open the Command Prompt, but the default one will automatically execute "petite file-path" in the command prompt, which is how you run the program. The second one will open the Command Prompt but will not run a command. This is useful if you do not wish to execute the default build's command, but instead may want to pass arguments. If you haven't used Sublime much, first make sure that Scheme (or whatever you named your new build system) is selected at "Tools -> Build System -> Scheme", then if you press `Ctrl + Shift + b`, it will pop up a little window for you to select which build to use. After selecting that build, you will only have to press `Ctrl + b` to build future programs in the same way.

```
{
  "cmd": ["start", "cmd", "/k", "petite", "$file"],
  "file_regex": "^[ ]*File \"(...*?)\"", line ([0-9]*),
  "shell": true,

  "variants": [
    {
      "cmd": ["start", "cmd"],
      "name": "args",
      "shell": true,
    }
  ]
}
```

I think a useful way to set up sublime would be to make two columns (by pressing `Alt + Shift + 2`), one that has the Petite Chez Scheme REPL loaded in, and the other a text editor for writing programs.

Since this is the first time I'm learning Scheme, this may not be exactly what we want. I am pretty sure, as I have tested and used this for small programs. If there is any tips or suggestions let me know and I can try to figure it out. Also just to mention, there is a package named "Glue" that runs the command prompt inside sublime instead of opening a new window. You may prefer this.

After reading through the first few sections in chapter 2, I noticed that we will probably be using the (load *string*) command a lot. So I went ahead and wrote a plugin and a key-bind to set your clipboard to that exact command, but *string* is replaced with the current file you are in. Just something little I did and thought I would share on here since I already stated other ways to use this language in Sublime. To add a new Plugin, click "Tools -> New Plugin...". Then copy and paste my following code in and save the file.

```
import sublime, sublime_plugin, os

class load_scheme_clipboard(sublime_plugin.TextCommand):
    def run(self, edit):
        view = sublime.Window.active_view(sublime.active_window())
        head, tail = os.path.split(view.file_name())
        sublime.set_clipboard("(load \"" + tail + "\"")
```

Then after that file is saved, go to "Preferences -> Key Bindings-User" and paste this code inside the brackets, then save:

```
{ "keys": ["ctrl+alt+c"], "command": "load_scheme_clipboard" }
```

After restarting sublime, if you go to a file and press `ctrl + alt + c`, your clipboard will now be that load command. With this, you can paste in the Petite Chez Scheme REPL to load the file and use it.