

# CSCI 350 Sublime Text 3 Guide

Written by: Stephen Tsung-Han Sher  
June 10, 2016

## Introduction

It's time you upgrade your text editor. Back in CSCI 104, you might have been told not to use *gedit* and use something like *emacs*, *vim*, or *Sublime Text*. When I first switched to *Sublime Text*, I used it the exact same way as I used *gedit*, which meant there was essentially no difference in my productivity nor efficiency. Halfway through the semester I decided to take a look into what *Sublime Text* had to offer, and it changed the way I used text editors forever.

So, I am here to show you the most used functionality of *Sublime Text* that I use. It's going to take a bit to get used to and memorize the hotkeys, so at the bottom of this guide I will give you a "cheat sheet" for you to use.

If you're curious more about this text editor, please do visit their website here: [www.sublimetext.com](http://www.sublimetext.com).

## Overview

The point of using a powerful text editor like *Sublime Text* is to keep your hands away from your mouse and more time on the keyboard, allowing you to navigate your code, files, and options as efficiently as possible. The rule of thumb is: **Keep your hands on your keyboard as often as possible.**

This rule of thumb is the same in *emacs*, *vim*, and other coding-centric text editors. I've seen someone use *vim* so efficiently he didn't need to touch the mouse at all, and coded faster than my eyes can follow; it honestly looked like a stereotypical hacking scene you see in blockbuster movies.

Without further ado, let's get started.



# Contents

<b>1 Essential Commands</b>	<b>3</b>
1.1 Customizing Your Sublime Window . . . . .	3
1.2 Menu-ing from Your Keyboard . . . . .	4
1.3 More Customization . . . . .	5
<b>2 Calling Sublime Text from Terminal</b>	<b>7</b>
<b>3 Navigating Your Project in Sublime Text</b>	<b>9</b>
<b>4 Navigating the File</b>	<b>12</b>
4.1 Find . . . . .	12
4.2 Find in Files . . . . .	13
4.3 Goto Line . . . . .	14
4.4 Find and Replace . . . . .	15
<b>5 Editing the File</b>	<b>15</b>
5.1 Moving the Cursor . . . . .	15
5.2 Scrolling . . . . .	15
5.3 Moving Lines . . . . .	16
5.4 Selecting Lines . . . . .	16
5.5 Multiple Cursors . . . . .	17
<b>6 Miscellaneous</b>	<b>19</b>
<b>7 Sublime Text Cheat Sheet</b>	<b>19</b>

# 1 Essential Commands

Some simple commands first:

- **Ctrl + S** save file
- **Ctrl + A** select all
- **Ctrl + W** close file
- **Ctrl + Shift + W** close all files in current window
- **Ctrl + N** new file as tab
- **Ctrl + Shift + N** new file as window
- **Ctrl + Tab** next tab
- **Ctrl + /** Comment / uncomment current line or highlighted lines
- **Ctrl + [** Tab back current line or highlighted lines
- **Ctrl + ]** Tab forward current line or highlighted lines

Basically it works as you would expect in a typical word processor or text editor. There are a few special commands, but we'll get into that later.

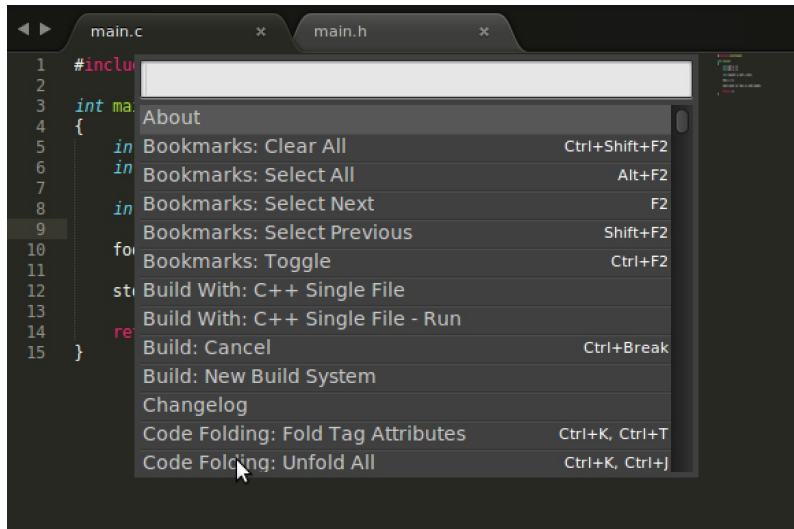
## 1.1 Customizing Your Sublime Window

Here's a list of commands you can use to change the look of *Sublime Text* to something you prefer:

- **Ctrl + =** zoom in
- **Ctrl + -** zoom out
- **Alt + Shift + 1** Have a 1 column window
- **Alt + Shift + 2** Have a 2 column window
- **Alt + Shift + 3** Have a 3 column window
- **Alt + Shift + 4** Have a 4 column window

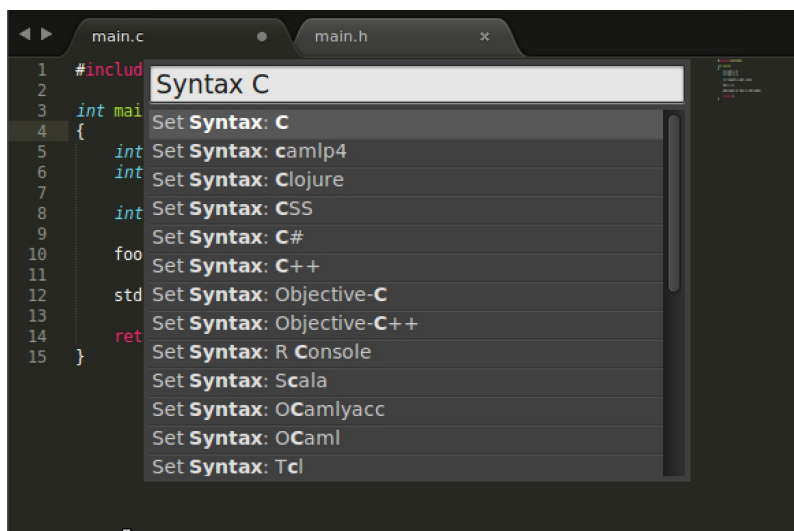
## 1.2 Menu-ing from Your Keyboard

A handy trick to keep in mind is that you can access most of your menu items right from your keyboard. If you press **Ctrl + Shift + P**, you will bring up a query for you to type in:



You don't have to search up the exact menu item name; a substring or keywords will work just fine<sup>1</sup>. There's a lot to explore here, but here's a small list of menu items you'll want to know:

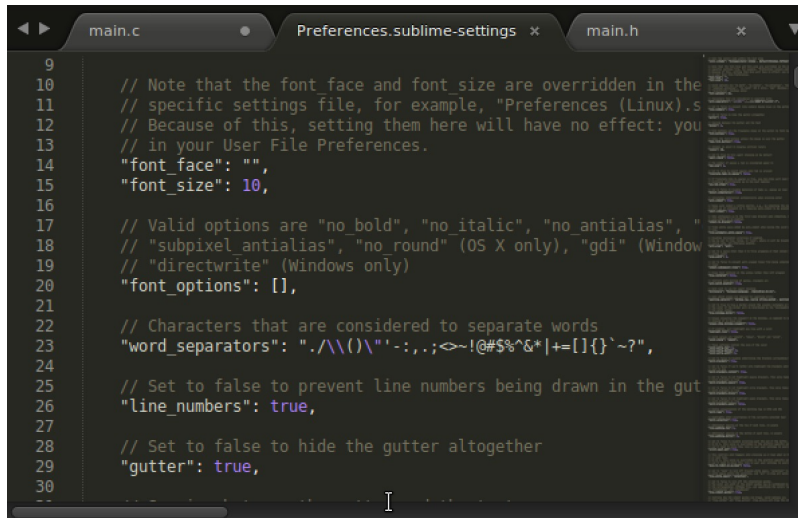
- Set Syntax: C                                Sets the syntax coloring of the file to C<sup>2</sup>
- Indentation: Convert to Spaces        Converts all tabs in the file to spaces
- Indentation: Convert to Tabs        Converts all four consecutive white spaces to tabs
- Preferences: Settings - Default       Look at default settings
- Preferences: Settings - User        Look at user settings



<sup>1</sup>Sublime Text uses something called Fuzzy Searching. More on this in a later section

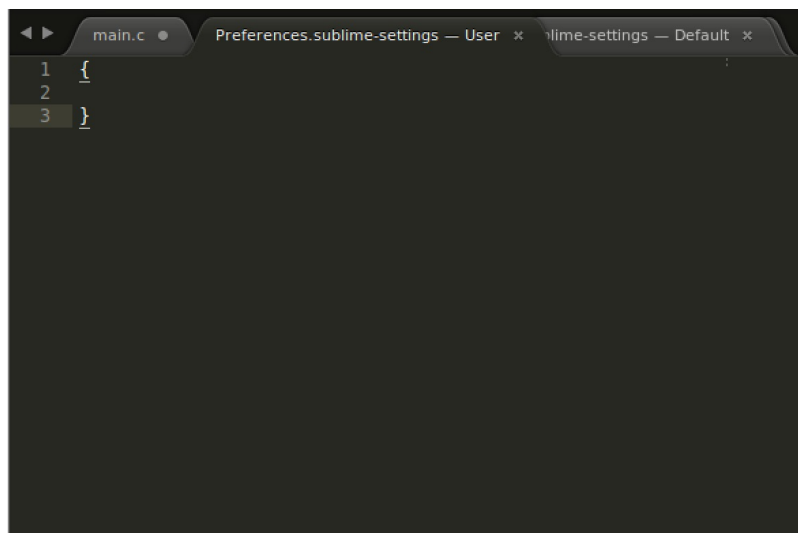
### 1.3 More Customization

If you look at **Preferences: Settings - Default**, you will see a huge number of settings *Sublime Text* offers you. Feel free to go through these and see what you can customize in your own time.



```
9
10 // Note that the font face and font_size are overridden in the
11 // specific settings file, for example, "Preferences (Linux).s
12 // Because of this, setting them here will have no effect: you
13 // in your User File Preferences.
14 "font_face": "",
15 "font_size": 10,
16
17 // Valid options are "no_bold", "no_italic", "no_antialias", "
18 // "subpixel_antialias", "no_round" (OS X only), "gdi" (Window
19 // "directwrite" (Windows only)
20 "font_options": [],
21
22 // Characters that are considered to separate words
23 "word_separators": ".\\(\\)\"'-:;=<~!@#%&*|+=[]{}`~?*",
24
25 // Set to false to prevent line numbers being drawn in the gut
26 "line_numbers": true,
27
28 // Set to false to hide the gutter altogether
29 "gutter": true,
30
--
```

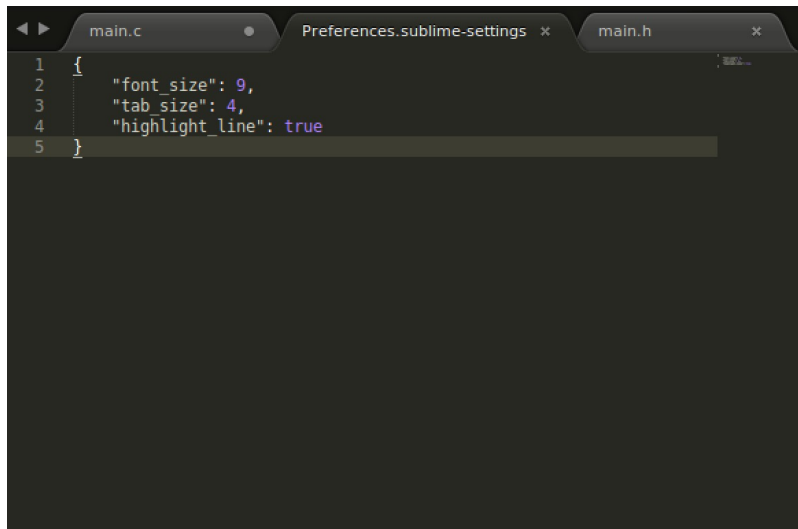
This document cannot be edited in any way; this is because this is the default settings. If there is anything you do want to change, you'll need to add it to **Preferences: Settings - User**:



```
1 {
2
3 }
```

You will notice that this document is empty at first; by default you have no user settings. You can copy and paste anything from **Preferences: Settings - Default** to this document and change the parameters. Some items that you might find useful are:

- "font\_size": Number                      Changes font size
- "tab\_size": Number                        Changes the number of white spaces in one tab
- "highlight\_line": true / false        Highlights the line the cursor is currently on



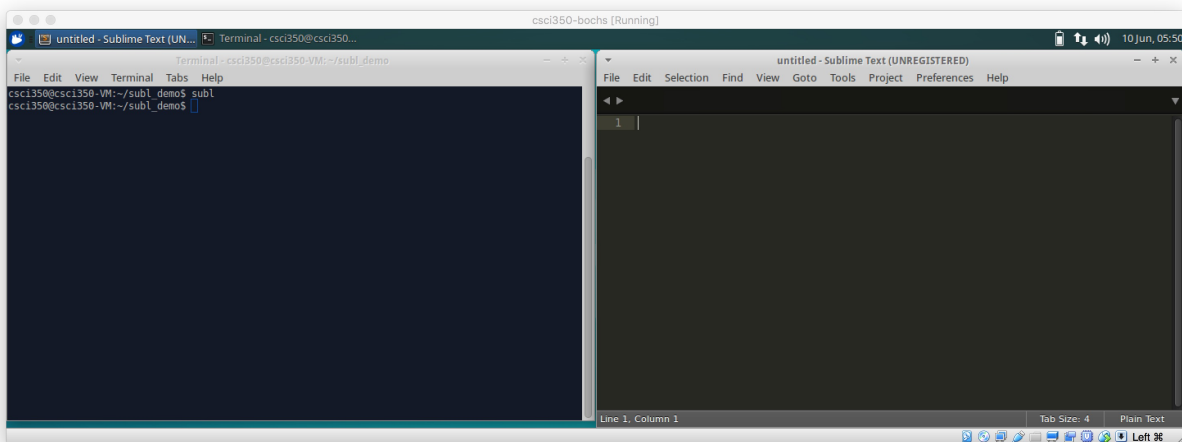
```
1 {
2   "font_size": 9,
3   "tab_size": 4,
4   "highlight_line": true
5 }
```

## 2 Calling Sublime Text from Terminal

In the VM that we have provided you, there is already an alias set up in the bash shell to call *Sublime Text* up directly. Note that you can do the same thing on a Macintosh, however you'll need to do a bit of setup. You can find a guide on how to do this at the end of this document.

To launch *Sublime Text* from the terminal, type:

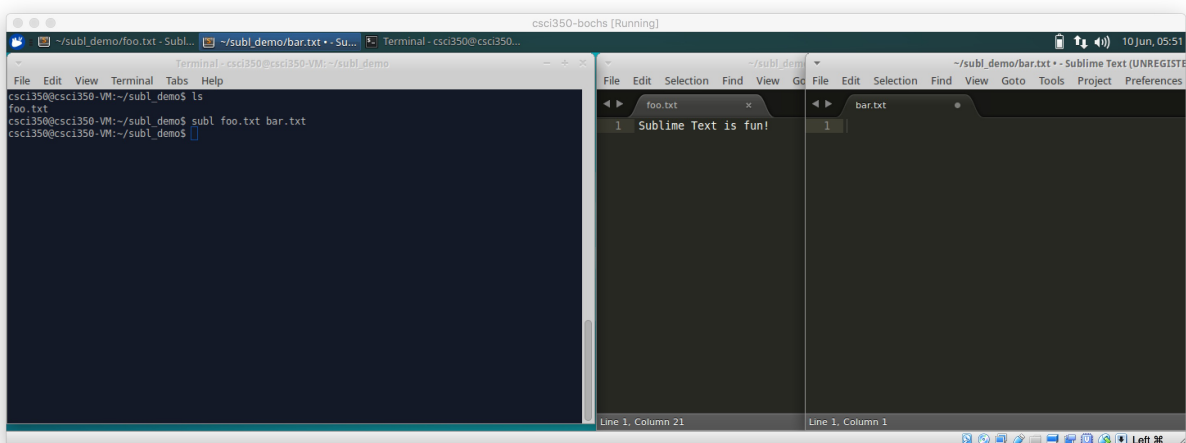
```
$csci350@csci350-VM:~$ subl
```



This will launch a new *Sublime Text* window<sup>3</sup>.

You can also launch *Sublime Text* designating an existing or new file by appending filenames as arguments:

```
$csci350@csci350-VM:~$ subl <filename1> <filename 2> <...>
```

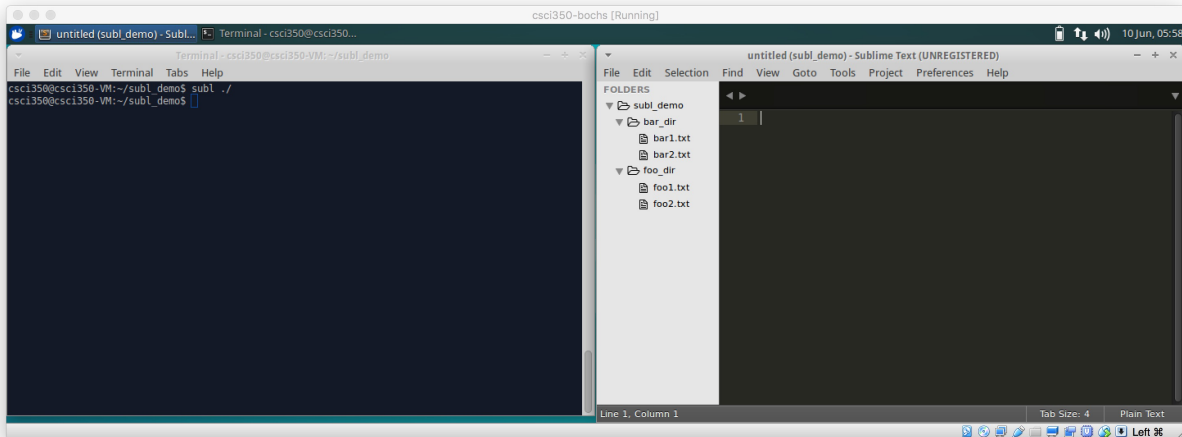


<sup>3</sup>If you quit out of *Sublime Text* without closing the windows, this launch the *Sublime Text* application with last session's unclosed windows

The most useful, by far, is launching *Sublime Text* on a directory <sup>4</sup>:

```
$csci350@csci350-VM:~$ subl <directory 1> <directory 2> <...>
```

*Note:* “./” is used to open the present working directory



This will open *Sublime Text* with the designated directory as a project. This is going to be immensely useful for *Pintos* since there might be close to 20 files you will need to work on at the same time. Opening a directory as a project in *Sublime Text* allows you to navigate all the files **inside** *Sublime Text*, avoiding the need to click around the file manager to find the file you want and opening them one by one.

---

<sup>4</sup>“./” is used to open the present working directory. You can even use “../” as you would navigating

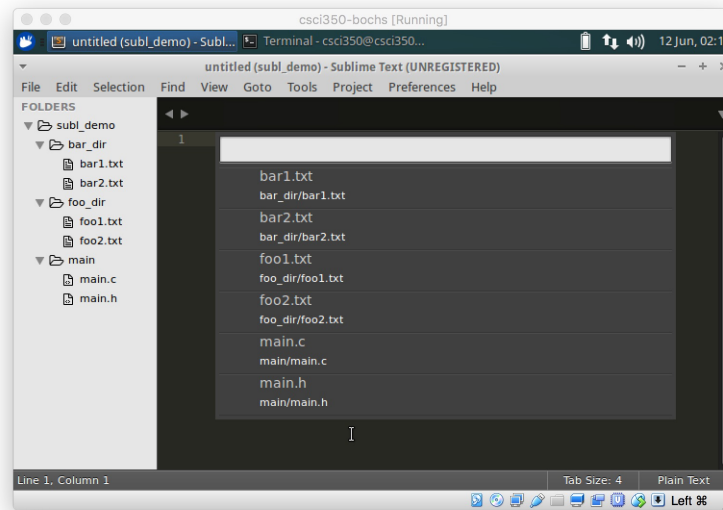


### 3 Navigating Your Project in Sublime Text

This is going to be your best friend working on Pintos.

Once you have a project opened in *Sublime Text*, the left hand has a side bar that shows the directories and files in your project. You can click on the directories to expand or contract to show the contents of the directories. However clicking isn't much fun is it? Remember, **Keep your hands on your keyboard as often as possible**.

You can open up your project navigator <sup>5</sup> with **Ctrl + P**:



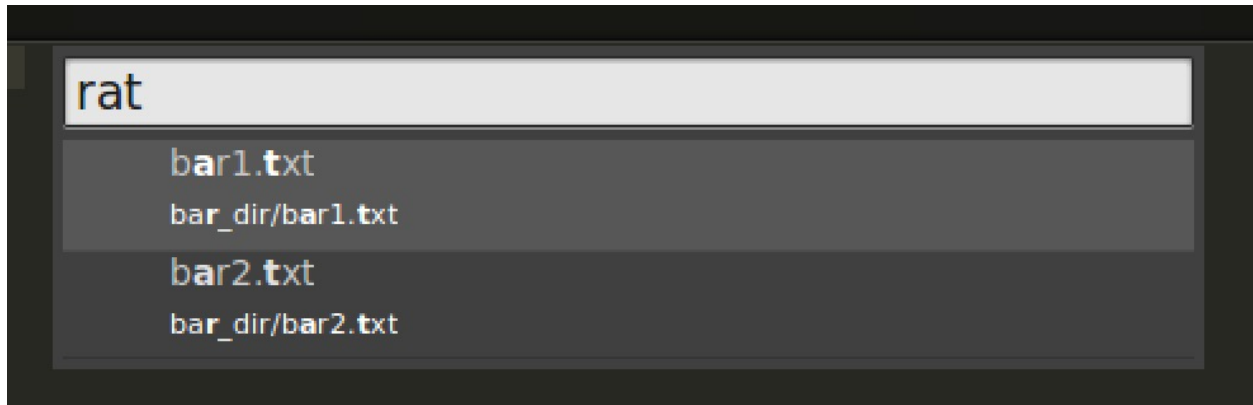
As you can see, this opens up a navigator that you can type into. Press **esc** to exit the navigator. The word you type will undergo what's called *fuzzy search*, meaning *Sublime Text* isn't going to just search the exact string or a substring, it's going to search any ordering of the letters you type in in any amount of arbitrary spacing. For example **foo** will return a search positive for:

- **foo**
- **ofo**
- **oof**
- This string will also **work** since the bolded letters matched up with the letters in **foo**

---

<sup>5</sup>Sublime Text calls this "Goto Anything", but it doesn't flow well

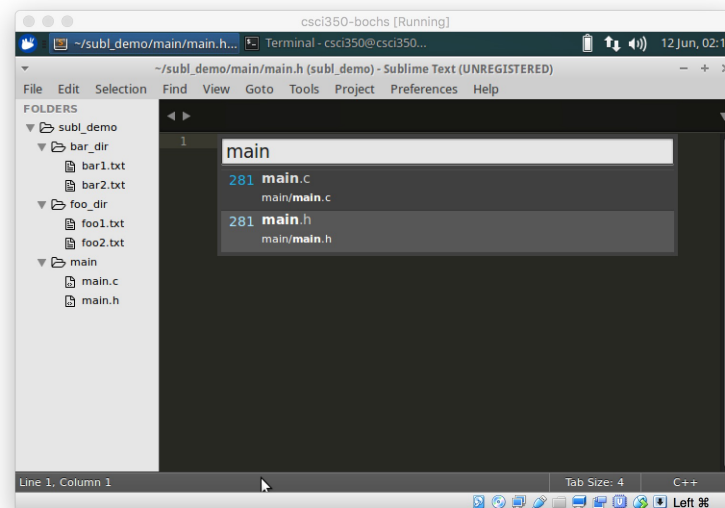
Whatever you type in will be searched with not only the file name, but with the entire file path including the directories.



Notice the word "rat" matched with these files due to the letters that are bolded white in the results. "r" matched with the "r" in the directory, "a" matched with the filename, and "t" matched with the extension. The results of fuzzy searching in *Sublime Text* will prioritize the best matched first (full strings, then substrings, then arbitrary ordering).

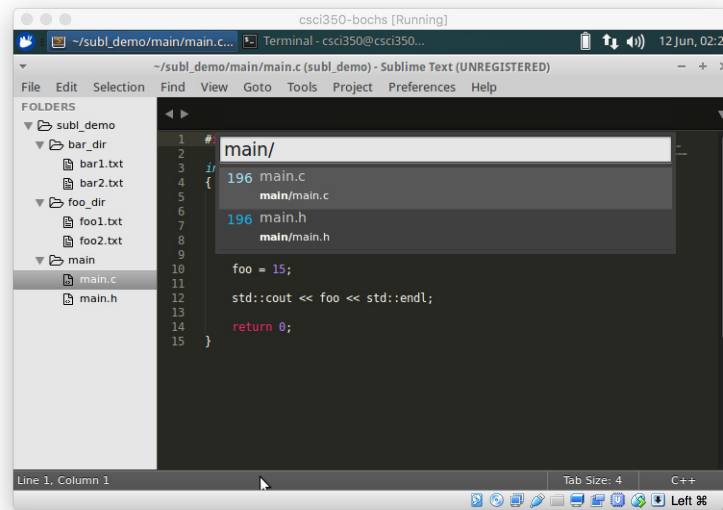
Once you have the results, you can use the up and down arrow keys to select through the results. For the file you want to open, you can press **Enter** to open the file. If you do so, the file will open and the navigator will close. If you want to open multiple files with one search result, use the right arrow key to open the selected file and remain in the navigator.

This is especially useful if you want to open both the `.c` and `.h` file. Use the navigator to search the name of the file.



Once the results show up, the `.c` and `.h` files should be the top two results. Simply press the right arrow key to open the first file, switch to the second file with the down arrow key, and press **Enter** to open the second file and exit the navigator, or use the right arrow key again to open the file and remain in the navigator and search for more files.

Note that if you don't know the filename, you can always search the directory name, and the results will show all the files for the given directory, allowing you to choose which file you would like to open. Note that this time the directory is bolded in the search results.



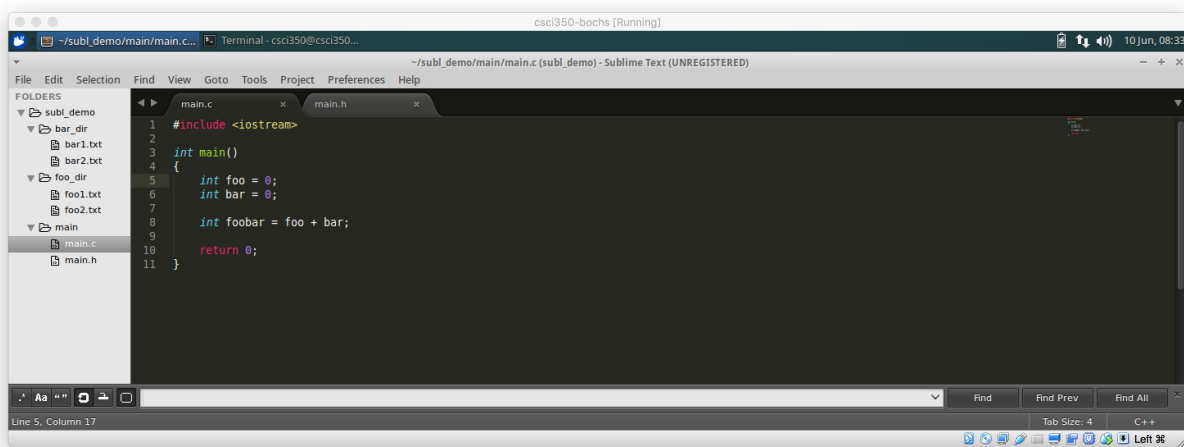
And just like that, you opened both the `.c` and `.h` file in less than five seconds. Much better than clicking around, and best of all, **your hands remain on the keyboard.**

## 4 Navigating the File

Once we can open any file we want with ease, let's now talk about how to navigate the contents of a file efficiently.

### 4.1 Find

Once you have a file open, you can start coding away. There are some pretty handy tools *Sublime Text* offers you. First is the good old trusty *find* function. As you would expect, **Ctrl + F** is going to give you the *find* prompt on the bottom of the window

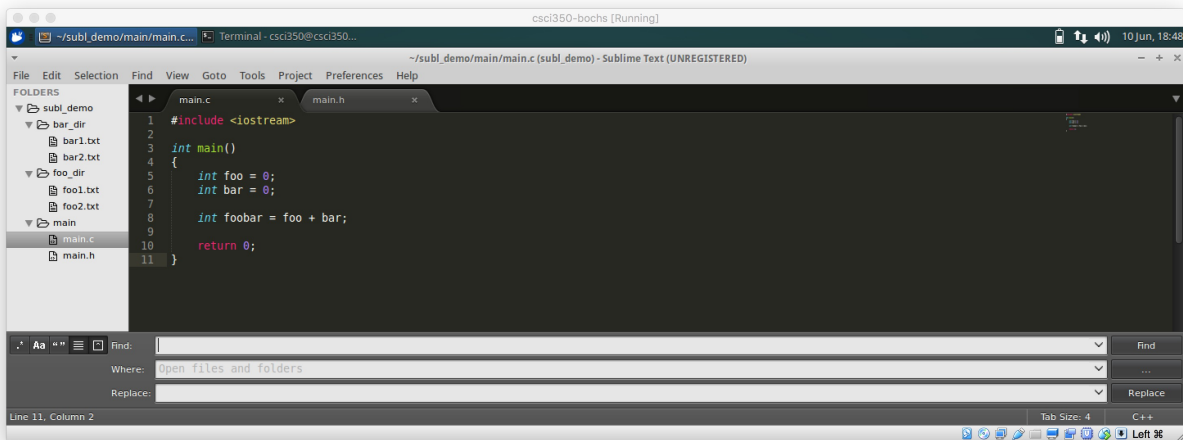


Note the bottom left hand corner first. The **[.\*]** is the regular expression options, where you can search strings using regular expressions (toggled with **Alt + R**). The next button **[Aa]** toggles case sensitivity (toggled with **Alt + C**), and the **[ " ]** toggles matching entire words (toggled with **Alt + W**). By default, none of these options are toggled.

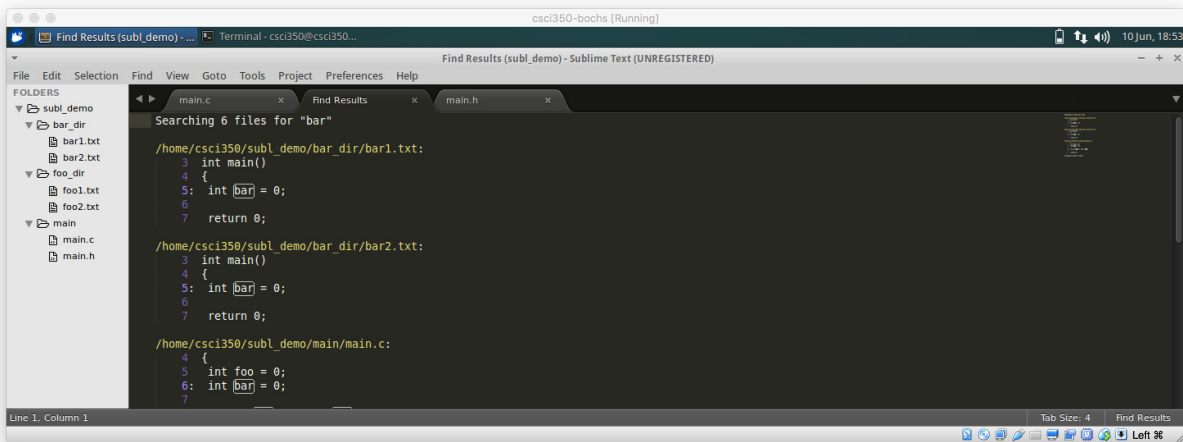
Using *find* is as you would expect; pressing **enter** will cycle through the document and highlight every instance of the query.

## 4.2 Find in Files

One of the most powerful functions in *Sublime Text* is *Find in Files*. This will let you use the find function to search across all files in your project. Press **Ctrl + Shift + F** to open the *Find in Files* prompt.



You can see in addition to the string query, you can also specify specifically where you want to search. I usually leave this blank and search the entire project. Once you enter your search, *Sublime Text* will generate a new temporary document with the find results called "*Find Results*".



You can see on the top of the document it indicates what you searched. If you make multiple *Find in Files* searches, it will concatenate each search result at the bottom of this file.

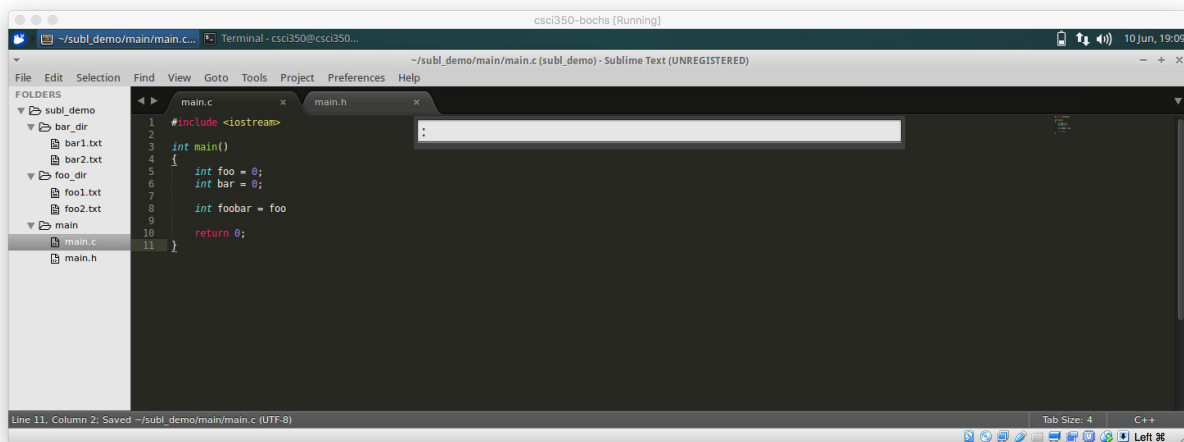
You can now see this file shows every line where it your search query returned positive, as well as a few lines above and below to give you some context. It also shows you the name of the file corresponding to the result. From here, you can scroll through your results to find which instance you want, and you can now:

1. Double click the filename and *Sublime Text* will take you directly to the corresponding file.
2. Double click the line containing the word searched for and *Sublime Text* will open the corresponding file and place the cursor at that specific line.

I find this especially helpful when I need to find instances in which I use a specific function. Searching the filename will let me see every instance of this function, and by double clicking the line I can go to that instance immediately. This is one of the very few time when you do need to use the mouse, however I find this much worth the extra time spent.

### 4.3 Goto Line

You also have the option of going directly to a desired line. By pressing **Ctrl + G**, you will have the same query input as if you were searching for a file in the project, except with a colon already inputted in the query<sup>6</sup>:

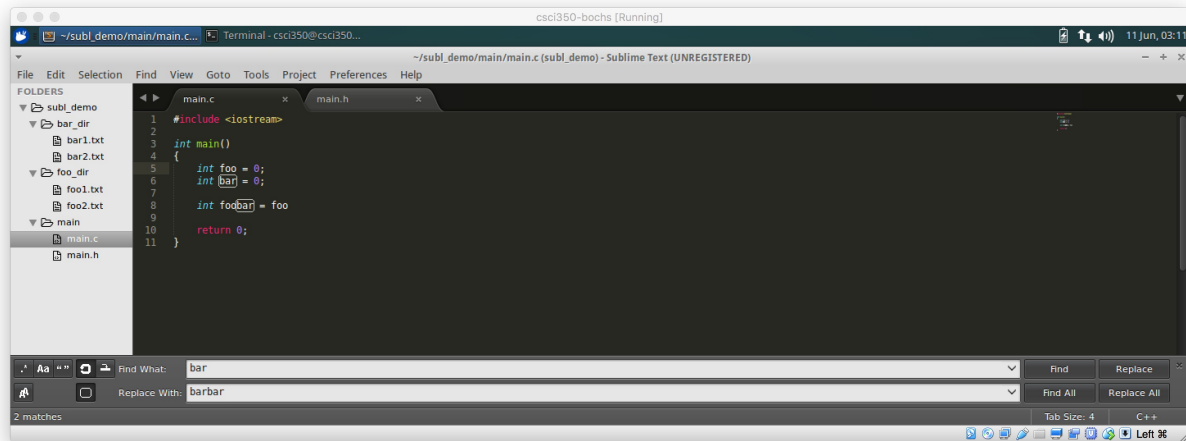


From here you can put in a line number, and your cursor will be taken directly to that line in the currently opened file. This is especially useful for compile errors when the compiler points out something wrong with a specific line in a specific file, you can directly go to that exact line with a **Ctrl + P** followed by a **Ctrl + G** command.

<sup>6</sup>Alternatively you can use **Ctrl + P**, type in a colon followed by a line number

## 4.4 Find and Replace

Much like a typical word processor or text editor, you can use the find and replace command. Invoke this function with **Ctrl + H**:



In this query, simply pressing **Enter** will perform a simple find function for you to go through and find the actual instance you want to replace. Once the instance you want to replace is highlighted, you can press **Ctrl + Shift + H** again to perform the replacement.

## 5 Editing the File

The next skill to learn after navigating the file is to modify the file efficiently. This section is perhaps going to give you some of the most powerful text editing tools

### 5.1 Moving the Cursor

Using the left and right arrow keys moves the cursor one character at a time. If you have an especially long line of code, you might want to move the cursor to the middle of the line. Instead of one character at a time, you can move one word at a time (in other words, moving to the next whitespace) by using **Ctrl + (← or →)**.

### 5.2 Scrolling

You can always scroll using your mouse or trackpad, however if you want to keep your hands on the keyboard as often as possible, keyboard scrolling might come in handy. The up and down arrow keys moves your cursor to the line above and below respectively. You can scroll one line at a time upwards and downwards by using **Ctrl + (↑ or ↓)**.

### 5.3 Moving Lines

If you want to move a line or lines of code to a different position in the file, cutting and pasting is definitely an option. However if the number of lines you want to move isn't significant, you can use **Ctrl + Shift + (↑ or ↓)** in the following two ways:

1. If you have your cursor on one line, using **Ctrl + Shift + (↑ or ↓)** will move the line you're on upwards or downwards respectively.
2. If you have multiple lines selected (not necessarily in consecutive order), using **Ctrl + Shift + (↑ or ↓)** will move the selected set of lines upwards or downwards respectively.

### 5.4 Selecting Lines

Rather than double-clicking a line to select it, you can select the line the cursor is on by using **Ctrl + L**. Once you have one line selected, pressing **Ctrl + L** in succession will select the next line each time you press this combination of keys.

This is powerful used in conjunction with **Ctrl + Shift + (↑ or ↓)** as it allows you quickly reposition small sections of code with ease.

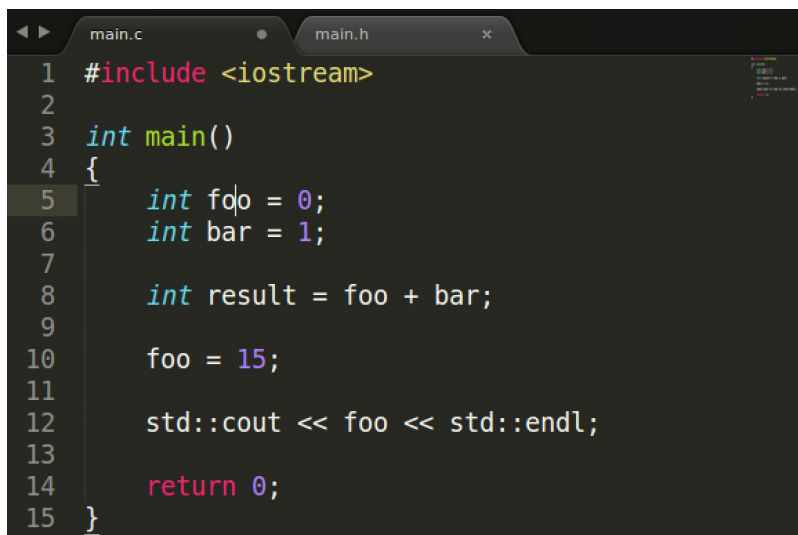


## 5.5 Multiple Cursors

This is a really fun one and by far my favorite feature about *Sublime Text*. If you have your cursor on a word, you can press **Ctrl + D** to highlight the word.

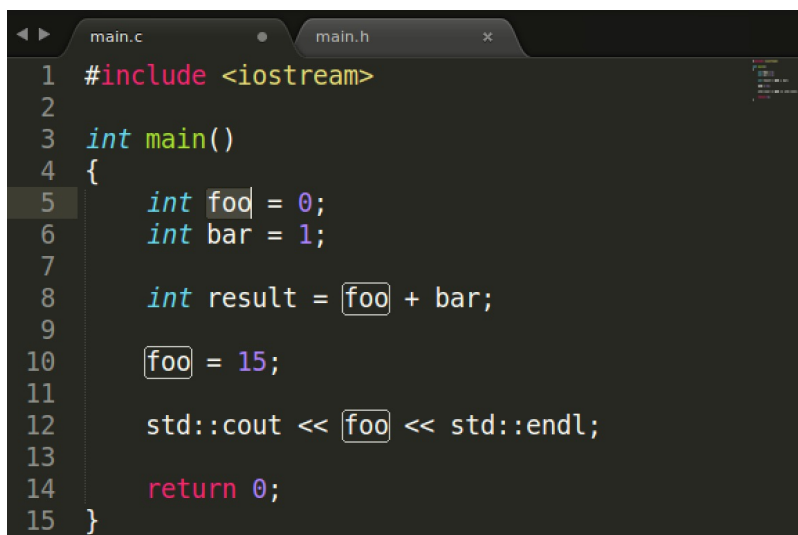
Once you have the first instance of this word highlighted, pressing **Ctrl + D** successively will highlight the next instance of this word. Once you have multiple words highlighted, you effectively have one cursor in each instance highlighted. In you proceed to type or use the arrow keys, you will see that you are simultaneously modifying at every instance highlighted.

This is a tad confusing, so I'll go through an example. Say we begin with this file:



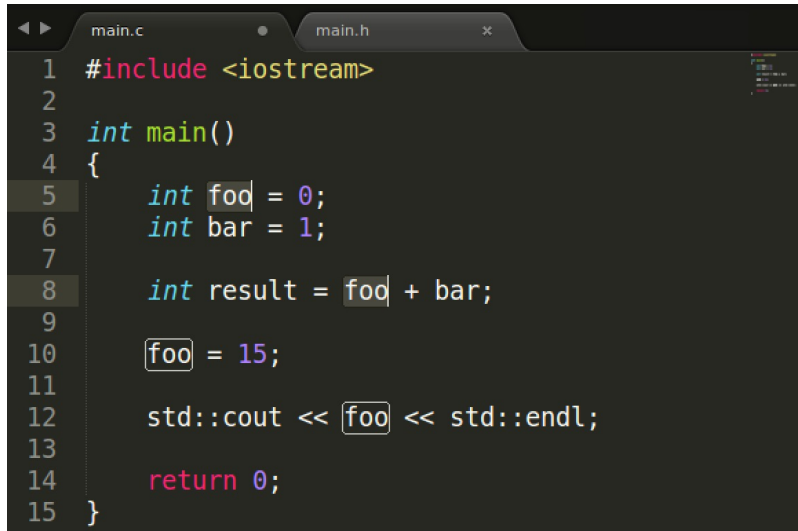
```
1 #include <iostream>
2
3 int main()
4 {
5     int foo = 0;
6     int bar = 1;
7
8     int result = foo + bar;
9
10    foo = 15;
11
12    std::cout << foo << std::endl;
13
14    return 0;
15 }
```

Notice how my cursor is on the word "foo" at this point. If I press **Ctrl + D**, *Sublime Text* will highlight "foo".



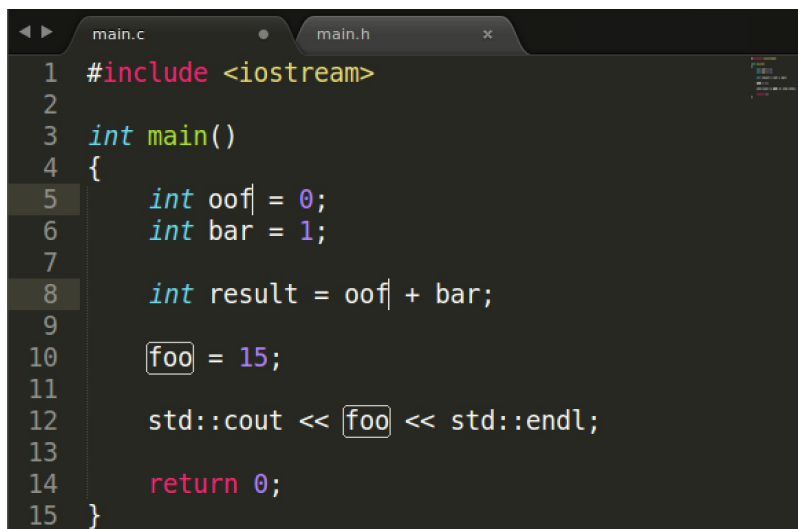
```
1 #include <iostream>
2
3 int main()
4 {
5     int foo = 0;
6     int bar = 1;
7
8     int result = foo + bar;
9
10    foo = 15;
11
12    std::cout << foo << std::endl;
13
14    return 0;
15 }
```

Now notice that "foo" is highlighted, and every single other instance of "foo" is boxed. The words that are boxed shows you which words are recognized as the same instance of the word you just highlighted. If we press **Ctrl + D** again, the next instance of the word "foo", which is on line 8, will be highlighted next.



```
1 #include <iostream>
2
3 int main()
4 {
5     int foo = 0;
6     int bar = 1;
7
8     int result = foo + bar;
9
10    foo = 15;
11
12    std::cout << foo << std::endl;
13
14    return 0;
15 }
```

This means that I have two cursors on line 5 and line 8, but not on 10 and 12 as I have not selected them yet. If I were to type "oof" on my keyboard, only lines 5 and 8 will be modified:



```
1 #include <iostream>
2
3 int main()
4 {
5     int oof = 0;
6     int bar = 1;
7
8     int result = oof + bar;
9
10    foo = 15;
11
12    std::cout << foo << std::endl;
13
14    return 0;
15 }
```

Note that I have two cursors on line 5 and 8, allowing me to freely move these two cursors simultaneously and make other modifications on these two lines at the same time. It does get tricky at times keeping track of all these cursors, but generally, it's not too bad.

## 6 Miscellaneous

There is still a lot to *Sublime Text* than what's covered here. If you want to know anything more, feel free to search online or ask me via email or in person.

I really hope this guide will help you improve your coding efficiency and help with managing your Pintos project this semester.

## 7 Sublime Text Cheat Sheet

Hotkey	Usage
Simple Commands	
<b>Ctrl + S</b>	save file
<b>Ctrl + A</b>	select all
<b>Ctrl + W</b>	close file
<b>Ctrl + Shift + W</b>	close all files in current window
<b>Ctrl + N</b>	new file as tab
<b>Ctrl + Shift + N</b>	new file as window
<b>Ctrl + Tab</b>	next tab
<b>Ctrl + /</b>	Comment / uncomment current line or highlighted lines
<b>Ctrl + [</b>	Tab back current line or highlighted lines
<b>Ctrl + ]</b>	Tab forward current line or highlighted lines
View Commands	
<b>Ctrl + =</b>	zoom in
<b>Ctrl + -</b>	zoom out
<b>Alt + Shift + 1</b>	Have a 1 column window
<b>Alt + Shift + 2</b>	Have a 2 column window
<b>Alt + Shift + 3</b>	Have a 3 column window
<b>Alt + Shift + 4</b>	Have a 4 column window
Project & File Commands	
<b>Ctrl + Shift + P</b>	Keyboard menu
<b>Ctrl + P</b>	Project navigator
<b>Ctrl + F</b>	Find
<b>Ctrl + Shift + F</b>	Find in files
<b>Ctrl + G</b>	Goto line
<b>Ctrl + H</b>	Find & Replace
<b>Ctrl + (← or →)</b>	Move cursor word by word
<b>Ctrl + (↑ or ↓)</b>	Scroll
<b>Ctrl + Shift + (↑ or ↓)</b>	Move lines
<b>Ctrl + L</b>	Select current line / next line
<b>Ctrl + D</b>	Select current word / subsequent instances