Sub[ime_ Productivity:

Code Like a Pro
with Today's Premier Text Editor]

Sublime Productivity

Code Like a Pro with Today's Premier Text Editor

Josh Earl

This book is for sale at http://leanpub.com/sublime-productivity

This version was published on 2015-09-16

Tweet This Book!

Please help Josh Earl by spreading the word about this book on Twitter!

The suggested hashtag for this book is #sublimetext.

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

https://twitter.com/search?q=#sublimetext

Contents

Contact Me

Sublime Productivity is a work in progress. If you'd like to let me know about an error (I'm sure there are a few), or if you just have a question or want to offer some constructive feedback, email me at josh@sublimeproductivity.com.

Free Weekly Productivity Tips

Want a weekly dose of Sublime Text tips to improve your productivity and streamline your workflow?

Head on over to http://sublimetexttips.com/newsletter[1] and drop your email address in the signup form.

---

[1]http://sublimetexttips.com/newsletter

1. Editing Text

Paste and Indent

Pasted code does its best to mess up your carefully formatted file, especially if you're embedding it into an existing block of code. Typically the leading white space from the copied code is added to the white space from the current code block.

For example, here's some JavaScript I'm working on:

```
1  $(function() {
2    function growBeets() {
3      //TODO implement beet growing algorithm
4    };
5  });
```

and in another code file I have some logic that I'm moving into the `growBeets` function:

```
1  $(function() {
2    function runSchruteFarm(bulbs) {
3      for (var i = 0; i < bulbs.length; i++) {
4        plant(bulbs[i]);
5      }
6    };
7  });
```

But when I position the cursor inside the `growBeets` function and paste the `for` loop, disaster ensues:

```
1  $(function() {
2    function growBeets() {
3        for (var i = 0; i < bulbs.length; i++) { // double the white space!!!
4        plant(bulbs[i]);
5      }
6    };
7  });
```

Paste and Indent handles this scenario correctly, calculating the appropriate amount of indentation for first line of the pasted code.

To perform a **Paste and Indent**:

1. Copy the desired lines of code to the clipboard.
2. Position the cursor at the correct indentation level.
3. Click **Edit | Paste and Indent** from the menu.
   Or press **Shift+Command+V** on Mac OS X or **Ctrl+Shift+V** on Windows and Linux.

You can remap the default **Paste** shortcut to perform a **Paste and Indent**, which should always do the right thing. Refer to the Key Bindings section for a how-to.

Indent, Unindent, Reindent

When I'm shuffling code around, I frequently need to adjust the indentation level. For example, removing an `if` block might leave a few lines indented one tab width too far.

The **Indent** and **Unindent** commands move lines in or out by one tab width.

The **Reindent** command attempts to reset the lines relative to the indentation level of the surrounding code. This is useful if you have multiple lines at several different indentation levels. **Reindent** does seem to get confused by comments, at least in JavaScript, but it's a good first-pass tool that reduces the amount of manual formatting required on a code block.

There's no keyboard shortcut for **Reindent** by default, but you can map one yourself if you use it frequently. Refer to the Key Bindings section for a how-to.

To use the **Indent** command:

1. Position the cursor on the line to be formatted, or select a block of lines.
2. Click **Edit | Line | Indent**.
   Or press **Command+]** on Mac OS X or **Ctrl+]** on Windows and Linux.

To use the **Unindent** command:

1. Position the cursor on the line to be formatted, or select a block of lines.
2. Click **Edit | Line | Unindent**.
   Or press **Command+[** on Mac OS X or **Ctrl+[** on Windows and Linux.

To use the **Reindent** command:

1. Position the cursor on the line to be formatted, or select a block of lines.
2. Click **Edit | Line | Reindent**.

Insert Line Before/Insert Line After

Sometimes it's necessary to open up space between the line I'm editing and the lines immediately before or after it. For example, I might be writing a line of jQuery code and decide that I should insert a comment above it, or maybe I want to set the line off from the code that follows with a line of white space.

Doing this manually involves jumping to the end of the line with **End** and then pressing **Return**, or pressing **Up** and then inserting an extra line.

The **Insert Line Before** and **Insert Line After** commands make this a one-step operation, inserting a blank line above or below the current line. It's similar to moving the cursor to the end of the line and pressing **Return**, but much faster. The cursor jumps to the new line so you can immediately begin typing.

Insert Line After will also indent the next line correctly. If, for example, the previous line defines a function, the inserted line will be indented one tab width.

To insert a blank line below the current line:

- Click **Edit | Text | Insert Line Before**.
  Or press **Command+Return** on Mac OS X or **Ctrl+Enter** on Windows and Linux.

To insert a blank line above the current line:

- Click **Edit | Text | Insert Line Before**.
  Or press **Shift+Command+Return** on Mac OS X or **Ctrl+Shift+Enter** on Windows and Linux.

Delete to End/Delete to Beginning

Often it's necessary to prune the beginning or end of a line, such as when you're removing part of a chained jQuery statement while keeping the selector, or removing the start or end tags on an HTML element. The usual way to do this is to highlight the text with the mouse or use a combination of the arrow keys **Shift** and **Home** or **End**, but Sublime's **Delete to Beginning** and **Delete to End** commands are a better option.

Delete to Beginning deletes all the text and white space preceding the cursor, while **Delete to End** removes everything that follows the cursor.

To use the **Delete to Beginning** command:

- Click **Edit | Text | Delete to Beginning**.
  Or press **Command+Delete** on Mac OS X or **Ctrl+K, Ctrl+Backspace** on Windows and Linux.

To use the **Delete to End** command:

- Click **Edit | Text | Delete to End**.
  Or press **Ctrl+K** on Mac OS X or **Ctrl+K, Ctrl+K** on Windows and Linux.

> The **Delete to End** and **Delete Line** shortcuts make a nice pair of editing tools, and Sublime recognizes this by making the keyboard shortcuts similar.

Delete Word Forward/Delete Word Backward

When you need to surgically remove a few words from the middle of a line, the **Delete Word Forward** and **Delete Word Backward** commands are a handy alternative to highlighting text with the mouse or **Shift** and arrow keys.

The **Delete Word Forward** command deletes from the cursor's current position to the next space. **Delete Word Backward** performs the same function in the opposite direction. Both will also delete partial words if the cursor is positioned inside of a word.

> There are no keyboard shortcuts mapped for these commands on Mac OS X by default, but it's simple to set them up yourself. Refer to the Key Bindings section for a how-to.

To use the **Delete Word Forward** command:

- Click **Edit | Text | Delete Word Forward**.
  Or press **Ctrl+Delete** on Windows or Linux.

To use the **Delete Word Backward** command:

- Click **Edit | Text | Delete Word Backward**.
  Or press **Ctrl+Backspace** on Windows or Linux.

Soft Undo/Soft Redo

Undo and redo are extremely useful, but I frequently find that I'll perform an undo only to realize that the action occurred off screen and I have no idea what, if anything, actually happened.

Sublime's **Soft Undo** and **Soft Redo** address this issue by considering cursor movements and text selections as operations that can be reversed. They are rewind and fast forward buttons for your editing session. This ensures that your attention is focused on any text changes that occur.

The ability to restore lost selections can be a lifesaver as well. It's frustrating when an errant click or button press cancels the selection of some text I've painstakingly highlighted, maybe a multi-screen block of text, or part of a three-screens wide piece of unwrapped code. In cases like these, executing a **Soft Undo** will restore the selection.

To perform a **Soft Undo**:

- Click the undo menu item under **Edit | Undo Selection**. (The text of the menu item changes to indicate what will happen when the action is invoked.) Or press **Command+U** on Mac OS X or **Ctrl+U** on Windows and Linux.

To perform a **Soft Redo**:

- Click the redo menu item under **Edit | Undo Selection**. (The text of the menu item changes to indicate what will happen when the action is invoked.) Or press **Shift+Command+U** on Mac OS X or **Ctrl+Shift+U** on Windows and Linux.

Transposing Letters and Words

Typing letters or words out of order usually necessitates a clumsy dance with the **Shift** and arrow keys, followed by a copy and paste and maybe fixing up the spaces on either side.

The **Transpose** command makes this a painless operation by reversing the position of a pair of letters or words.

> There's no keyboard shortcut for **Transpose** by default on Mac OS X, but it's easy to set one up. Refer to the Key Bindings section for a how-to.

To use **Transpose**:

1. Position the cursor between the letters or words you'd like to reverse.
2. Click **Edit | Text | Transpose**.
   Or press **Ctrl+T** on Windows and Linux.

Changing Capitalization

For sheer tedium, few tasks can rival deleting words and retyping them to convert ALL CAPS to lowercase. Sublime supports several features for dealing with capitalization chores.

The **Swap Case** option inverts the casing of all selected text, while **Title Case** capitalizes the first letter of each word. **Upper** capitalizes all selected letters, while **Lower** does the opposite.

To use the **Swap Case** command:

- Click **Edit | Convert Case | Swap Case**.

To use the **Title Case** command:

- Click **Edit | Convert Case | Title Case**.

To use the **Upper** command:

- Click **Edit | Convert Case | Upper**.
  Or press **Command+K**, **Command+U** on Mac OS X or **Ctrl+K**, **Ctrl+U** on Windows and Linux.

To use the **Lower** command:

- Click **Edit | Convert Case | Lower**.
  Or press **Command+K**, **Command+L** on Mac OS X or **Ctrl+K**, **Ctrl+L** on Windows and Linux.

To define keyboard shortcuts for **Swap Case** or **Title Case**, refer to the Key Bindings section.

Joining Lines

I've probably wasted more hours of my life than I care to admit deleting extraneous white spaces when merging two lines of indented code. When I learned how to join lines, I couldn't believe I'd been pounding **Delete** unnecessarily.

Sublime employs the **Join Lines** command to join the line below your cursor to the one you're currently editing.

To use the **Join Lines** command:

1. Position the cursor on the first of the two lines.
2. Click **Edit | Line | Join Lines**.
   Or press **Command+J** on Mac OS X or **Ctrl+J** on Windows and Linux.

Swapping Lines

Another common line operation is swapping the position of two lines, or shifting a block of lines up or down in the file. Copy and paste is one option, but if you'll only be moving the lines a short distance, the **Swap Line Up** and **Swap Line Down** commands are a better alternative.

To use the **Swap Line Up** command:

1. Position the cursor on the line to be moved, or highlight one or more lines.
2. Click **Edit | Line | Swap Line Up**.
   Or press **Ctrl+Command+Up** on Mac OS X or **Ctrl+Shift+Up** on Windows and Linux.

To use the **Swap Line Down** command:

1. Position the cursor on the line to be moved, or highlight one or more lines.
2. Click **Edit | Line | Swap Line Down**.
   Or press **Ctrl+Command+Down** on Mac OS X or **Ctrl+Shift+Down** on Windows and Linux.

Deleting Lines

Deleting an entire line without highlighting it first is simple using the **Delete Line** command.

To use the **Delete Line** command:

1. Position the cursor on the line to be deleted.
2. Click **Edit | Line | Delete Line**.
   Or press **Ctrl+Shift+K**.

Duplicating Lines

Duplicating an entire line or series of lines without copying and pasting is possible with the **Duplicate Line** command.

Duplicate Line inserts a copy of the current line directly below the current line. If multiple lines are highlighted, it performs the same action with the entire block.

To use the **Duplicate Line** command:

1. Position the cursor on the line to be copied, or highlight one or more lines.
2. Click **Edit | Line | Duplicate Line**.
   Or press **Shift+Command+D** on Mac OS X or **Ctrl+Shift+D** on Windows or Linux.

Wrapping Paragraphs

I like to keep my lines fairly short, so I have a custom ruler set at column 100. This is great for manually wrapping lines, but sometimes when I'm writing prose, I'll edit a paragraph and find that my original wrapping is no longer suitable.

Sublime makes it painless to fix this with the **Wrap Paragraph at Ruler** command.

To use the **Wrap Paragraph at Ruler** command:

1. Select a block of text.
2. Click **Edit | Wrap | Wrap Paragraph at Ruler**.
   Or press **Option+Command+Q** on Mac OS X or **Alt+Q** on Windows or Linux.

Sublime also supports wrapping to predefined column widths of 70, 78, 80, 100 and 120 characters.

To wrap at a width of 80 characters:

1. Select a block of text.
2. Click **Edit | Wrap | Wrap paragraph at 80 characters**.

Commenting and Uncommenting

Sublime gracefully handles both inline and block comments in a variety of languages with the **Toggle Comment** and **Toggle Block Comment** commands. The behavior of these commands differs slightly depending on the language's support for inline and block comments.

CSS and HTML, for example, only support block comments, so the **Toggle Comment** command will wrap the current line or selected text in block comments, or remove the block comment as appropriate. **Toggle Block Comment** will insert an empty block comment if nothing is selected, or it will wrap the selected text in a block comment, or remove the block comment if the code is already commented out.

JavaScript supports line comments, so **Toggle Comment** will add or remove line comments, and **Toggle Block Comment** will use block comments.

To use the **Toggle Comment** command:

1. Position the cursor on the desired line, or select a block of lines.
2. Click **Edit | Comment | Toggle Comment**.
   Or press **Command+/** on Mac OS X, or **Ctrl+/** on Windows or Linux.

To use the **Toggle Block Comment** command:

1. Position the cursor on the desired line, or select a block of lines.
2. Click **Edit | Comment | Toggle Block Comment**.
   Or press **Option+Command+/** on Mac OS X, or **Ctrl+Shift+/** on Windows or Linux.

Sorting and Reordering Lines

In the past, I've occasionally found myself copying lines of text into a spreadsheet program to sort them. Sublime has this covered, though, with several simple sorting algorithms:

- **Sort Lines** performs a simple alpha sort.
- **Sort Lines (Case Sensitive)** does an alpha sort, starting with capitals.
- **Reverse** sorts in reverse alphabetical order, starting with lowercase letters.
- **Shuffle** does a pseudo-random sort.
- **Unique** removes duplicated lines.

To use the **Sort Lines** command:

1. Select a block of lines.
2. Click **Edit | Sort Lines**.
   Or press **F5** on Mac OS X or **F9** on Windows or Linux.

To use the **Sort Lines (Case Sensitive)** command:

1. Select a block of lines.
2. Click **Edit | Sort Lines (Case Sensitive)**.
   Or press **Ctrl+F5** on Mac OS X or **Ctrl+F9** on Windows or Linux.

To use the **Reverse** command:

1. Select a block of lines.
2. Click **Edit | Permute Lines | Reverse**.

To use the **Shuffle** command:

1. Select a block of lines.
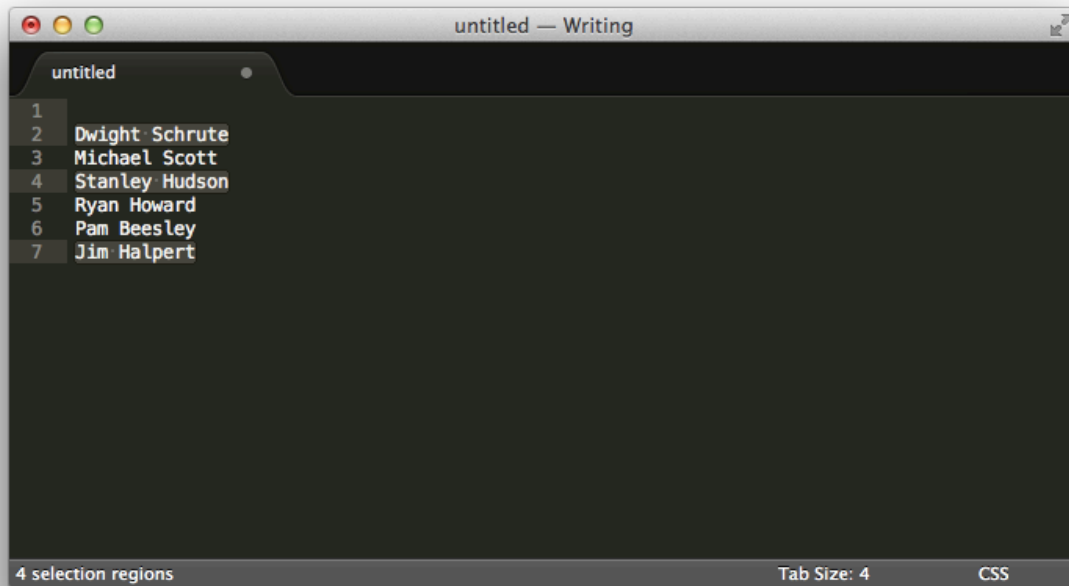2. Click **Edit | Permute Lines | Shuffle**.

To use the **Unique** command:

1. Select a block of lines.
2. Click **Edit | Permute Lines | Unique**.

Sorting and Reordering Selected Items

In addition to sorting a solid block of lines, Sublime can perform the same sorting functions on non-sequential lines using the items in the **Permute Selections** menu.
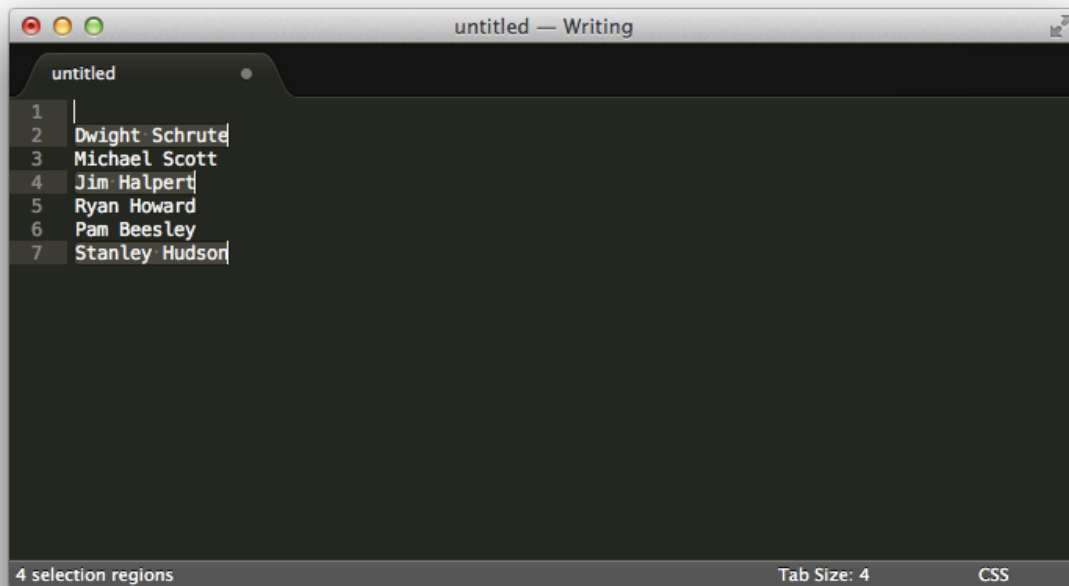
For example, this **Sort** variation turns this list:

Select several lines …

Into this:

And the selections are sorted in place!

To use the **Sort Lines** command:

1. Select several non-sequential lines.
2. Click **Edit | Permute Selection | Sort**.

To use the **Sort Lines (Case Sensitive)** command:

1. Select several non-sequential lines.
2. Click **Edit | Permute Selection | Sort Lines (Case Sensitive)**.

To use the **Reverse** command:

1. Select several non-sequential lines.
2. Click **Edit | Permute Selection | Reverse**.

To use the **Shuffle** command:

1. Select several non-sequential lines.
2. Click **Edit | Permute Selection | Shuffle**.

To use the **Unique** command:

1. Select several non-sequential lines.
2. Click **Edit | Permute Selection | Unique**.