

Re-re-re-recap

What did we learn again?

Last we got started on our path to the wonderful land of Front-End Development.

Now, let's go over some of the both the fun and the sticky bits.

General:

HTML: defines the structure of our content (frame of the house)

CSS: defines the appearance (paint and decor)

HTML & CSS

`<!doctype html>` : Goes before all tags in HTML document
Says to the browser, hey, I'm using HTML version 5 (and not 4, etc.)

Basic HTML page structure:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

HTML & CSS

HTML5 Tags

<header></header>

<nav></nav>

<main></main>

<section></section>

<article></article>

<aside></aside>

<footer></footer>

HTML & CSS

CSS : A language to style HTML

Separates style from content

Can be used to set the style of an entire website

CSS selectors

#some-id { } – <tag id="some-id">

.some-class { } – <tag class="some-class">

tag { } – <tag>

Multiple classes

<p class="class-one class-two"></p>

Best Practices

Commenting your code!

HTML Comment:

```
<!-- Hello you knowledgeable person -->
```

CSS Comment:

```
/* Hello you fashionable person */
```

Styling

Use classes to identify tags in your HTML for styling.

Using an ID isn't wrong, but remember that IDs can only be used once on each page.

Sticky Bits

The D.O.M (Document Object Model)

CSS Specificity:

Most to least specific

1. `<style>` attribute
2. `#id`
3. `.class`
4. `p (tags)`

BEST PRACTICES PART 2

Let's add in a couple of new best practices in our workflow.

Filename: When you are naming your files, try to give them names that are memorable and relevant to the code they hold.

For Example:

main.css

styles.css

manic_pixie_dreams.css

whatcha_ma_call_it.css

Remember all lowercase filenames and no spaces.

BEST PRACTICES PART 2

Indenting Code: When coding you want to make sure to indent your code. This makes your code readable in general whitespace (spaces, new lines, tabs) have no effect on your code output.

GOOD!

```
HTML
1 <div class="bs-docs-section">
2   <h1 id="overview">Overview</h1>
3
4   <p class="lead">
5     Get the lowdown on the key pieces of
6     Bootstrap's infrastructure, including
7     our approach to better, faster,
8     stronger web development.
9   </p>
10
11 </div>
```

DEAR LORD WHY!!!

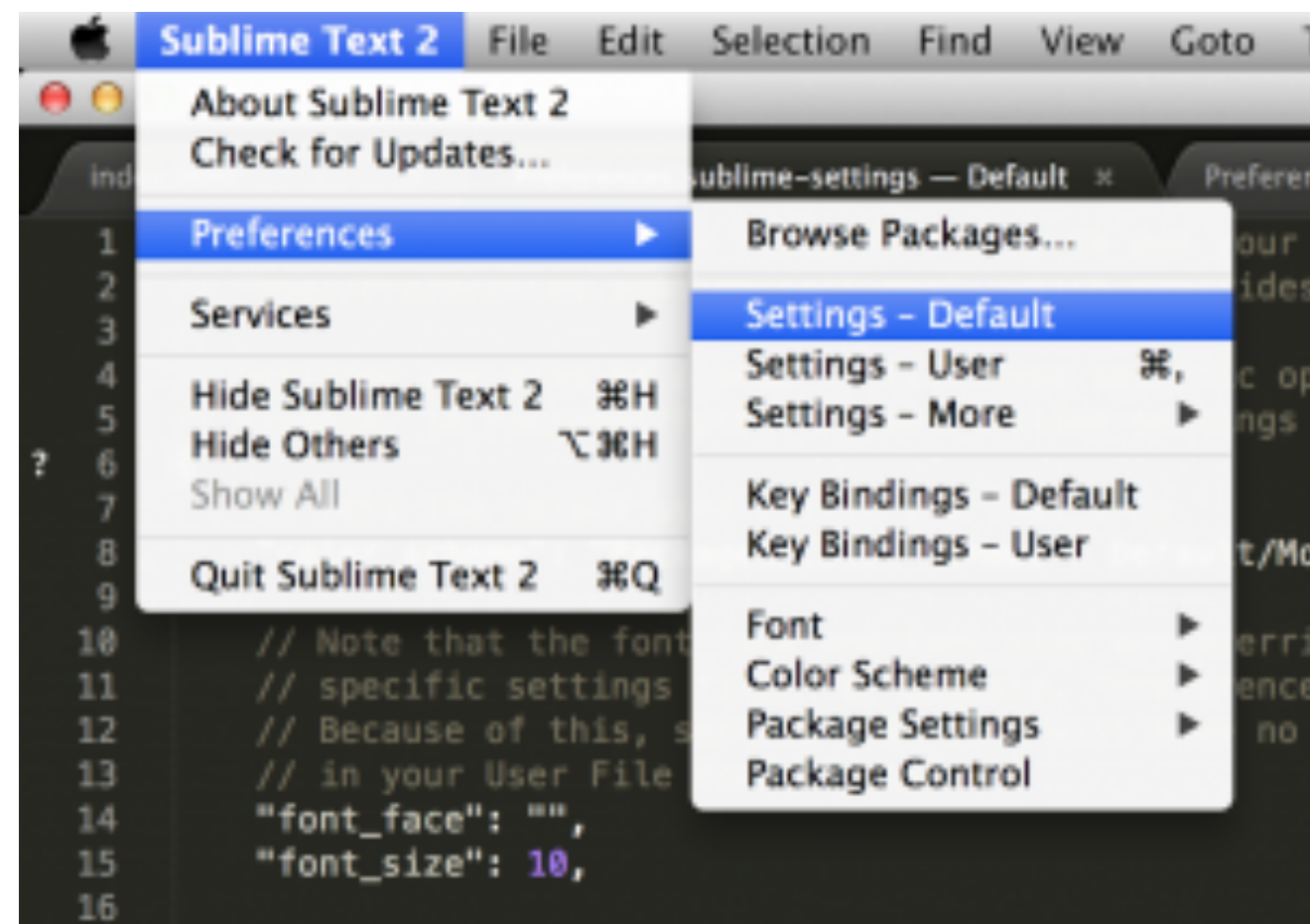
```
HTML
1 <div class="bs-docs-section"><h1 id="overview">Overview</h1>
2 <p class="lead">Get the lowdown on the key pieces of
3 Bootstrap's infrastructure, including
4 our approach to better, faster,
5 stronger web development.</p></div>
```

BEST PRACTICES PART 2

SUBLIME CAN DO WHAT NOW!

Setup up your indentation:
Open up the default settings in
Sublime

Find these two lines:
// The number of spaces a tab is
considered equal to "tab_size": 4,
(change to 2)
// Set to true to insert spaces when
tab is pressed
"translate_tabs_to_spaces": false,
(change to true)



BEST PRACTICES PART 2

AUTO-BUILD IN CHROME FOR SUBLIME

Auto-build will allow you to type command/control B in your html file, and have it pop up in Chrome (or the browser of your choice).

To set this up, complete the following: Sublime Build Create a new build system: Write the follow code the following into the file that pops up: (Do not copy and paste and it won't work.)

```
{ "cmd": ["open", "-a", "Google Chrome", "$file"] }
```

BEST PRACTICES PART 2

AUTO-BUILD IN CHROME FOR SUBLIME

Save the file, and then completely quit out of Sublime. Restart Sublime, go back to the tools menu to select your new build (Chrome should now be an option in the build system menu).

Now that you have selected Chrome as the build system, whenever you type command/control B, the file you are working on will open in Chrome.

