

Website Walk-through

Using Brackets

Brackets - brackets.io/ - FREE

Brackets is a free and open source editor by a big name. Adobe, to be exact. Brackets comes with some unique and exciting features. One of those being Extract, which allows you to extract information directly from PSDs such as colors, fonts, gradients, and measurements as clean CSS with out contextual code hints. Plugin friendly.

Mac, Win, Linux

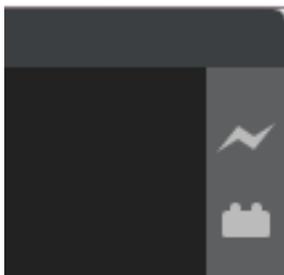
Code Editor Features

Being productive and getting things done quickly is a must. These days most good editors have plugins or extension that extend the features and capabilities of the software.

Here are just a few plugins to look for on your coding path:

- Emmet - faster coding
- LiveReload - changes happen in the browser instantly
- Color coding for errors
- Store and use code snippets

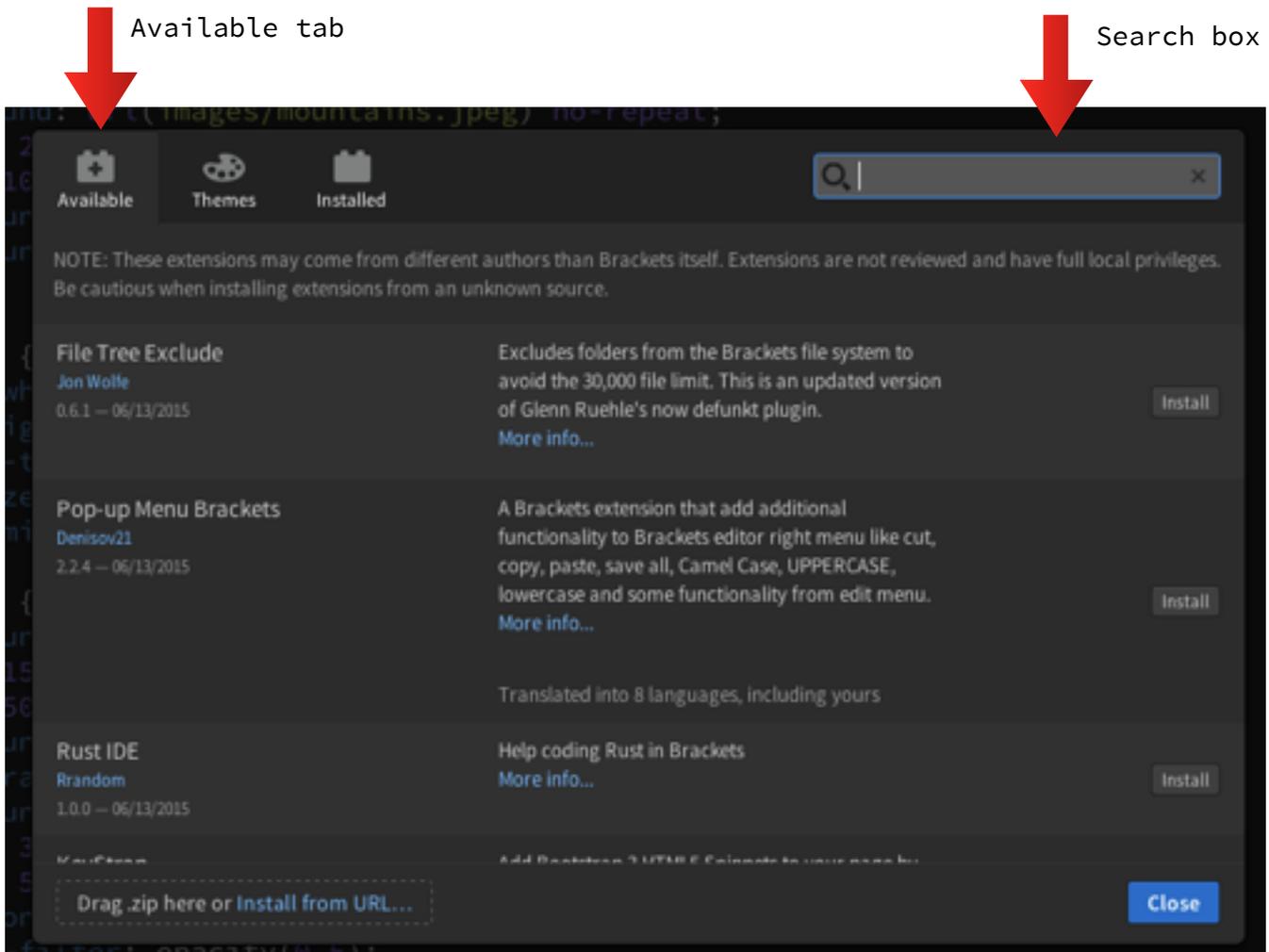
Brackets Extension Manager



In the top right corner of Brackets, there is a “lego brick” looking icon that is the Extension Manager.

{WebCodeChicks}

Use the search box to find the extensions you are looking for with the available tab selected.



Search for these time saving extensions:

- Autosave Files on Window Blur
- *Brackets Snippets by edc
- ColorHints
- Duplicate Files and Folders
- *Emmet
- Static Preview - alternative for Brackets Live Preview that works with any browser and Chrome Dev tools.

* - must have

Helpful Keyboard Shortcuts

Keyboard Shortcut	Mac	Windows
Increase/Decrease Font Size	Command-+	Ctrl-+/-
Move Line	Control-Command-Up/Down	Ctrl-Shift-Up/Down
Indent	Command-[Ctrl-[
UnIndent	Command-]	Ctrl-]

Lorem Ipsum is a filler text commonly used to demonstrate the graphic elements of a document or visual presentation. Replacing meaningful content with placeholder text allows viewers to focus on graphic aspects such as font, typography, and page layout without being distracted by the content.

With the **Emmet extension installed**, by typing 'lorem25' and then pressing tab, you will get 25 words of filler text. Replace the 25 with any number to get more or less words. Awesome! FYI it is nonsensical Latin.

Let's get this party project started!

Go get the files for this project at the WebCodeChick Bitbucket Repository:

<https://bitbucket.org/wcc-sc/>

wccwalkthrough2015

You will get two folders:

practice

final

The **practice** folder has blank html pages (except for the navigation) and css file and all assets

The **final** folder is the completed project with all assets

Free Theme for Practice:

<http://www.blacktie.co/2013/10/spot-freelance-agency-theme/>

{WebCodeChicks}

I made some changes, took some stuff out to make it a basic theme. Go to the link above to get the full version.

(For time constraints of this class in the classroom, I have decided to put the navigation in all pages. I describe the details of what is happening in the Bootstrap navigation, but to go through the typing of the menu would take away from time of building the site.)

Google Fonts - <https://www.google.com/fonts>

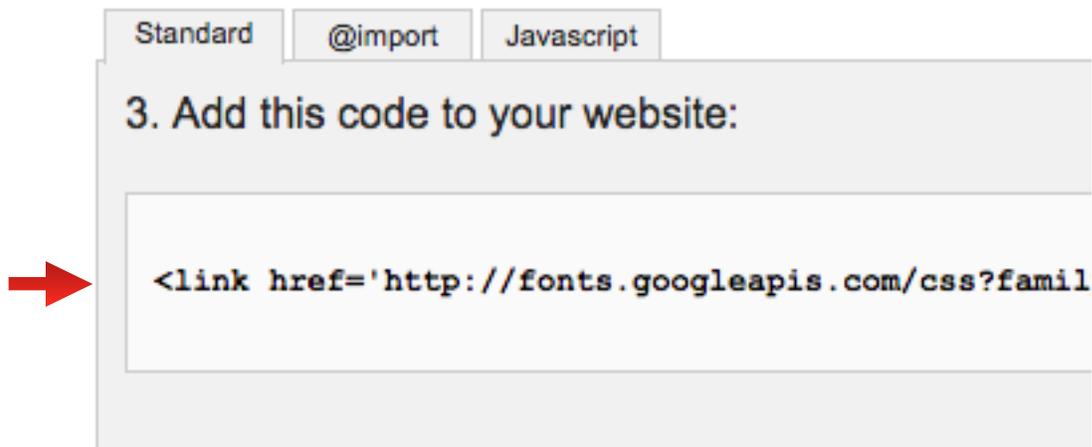
Raleway and Lato

Both of these fonts have multiple font weights: 300, 400, 700, 900.



Quick Use Button

1. Select the Font Sizes
2. Skip this section unless you know you need Character Sets.
3. Click and drag over the code to use in your HTML pages.



Which way should I add the fonts to the web page: Standard, @import or Javascript?

One would argue about repeating the code as a <link> every HTML page. The alternative is to call it once in CSS as @import is the “better” way to call the fonts.

As a rule of thumb, you want to avoid @import rules because it defers the loading of the included resource (it is loaded later than the page itself) until the css file is fetched, then the call to Google Fonts. By placing the link to Google Fonts in the the top of the HTML, it is loaded early with the rest of the page. (The default/standard choice that Google presents.)

This goes for the other files that are needed for the site as well: Bootstrap, Font-Awesome and jQuery.

Bootstrap - <http://getbootstrap.com>

This is a whole other class to get to know. I will be going over the basics of the grid, and the navigation.

CSS, Specificity and Inheritance

The concepts of inheritance and overriding properties.

This is the magic understanding of CSS - Specificity!

Here are some links:

MDN Mozilla Developer Network (great documentation)

<https://developer.mozilla.org/en-US/docs/Web/CSS/Specificity>

Jonathn Snook

http://snook.ca/archives/html_and_css/understanding_c/

Specificity for Poker Players - <http://carl.camera/?id=95>

JavaScript

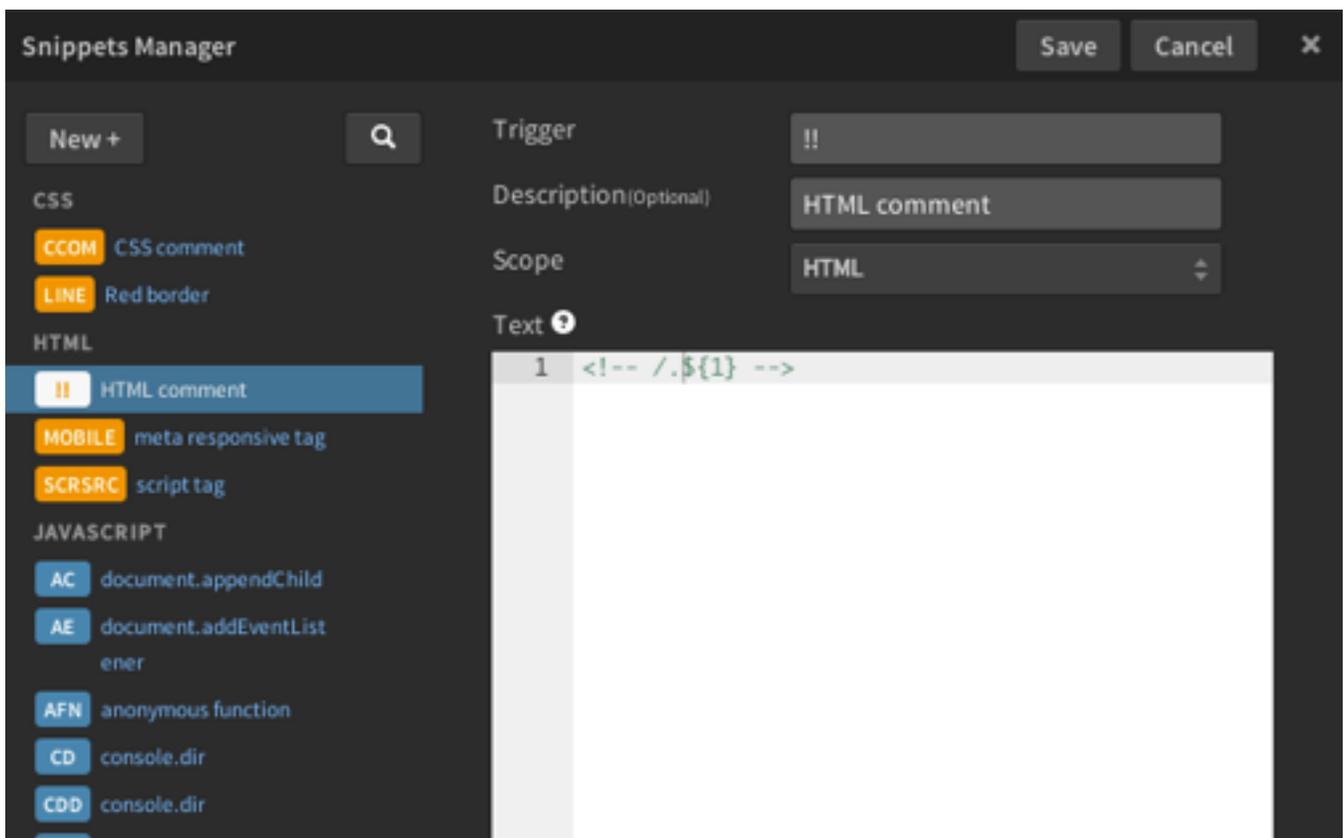
JavaScript files are usually placed at the bottom of the HTML page inside the `</body>`. This also lets the page load and then the JS after the page is seen.

Brackets Snippets

After installing the Brackets Snippets by edc (more than one, this is the best one!), The lightbulb icon shows up on the right bar:



By clicking on it, a Snippets Manager pane shows up at the bottom:



{WebCodeChicks}

Use the New+ button to create a new snippet. Your new snippets are in orange. In the image above I have given you an example of the HTML comment and it's code.

Trigger field is for what keystrokes you want to “trigger” the code to be entered after hitting the tab key.

Description helps you remember what the code is or does.

Scope is the language in which the code will only be triggered.

Text is the code you want to see after typing the trigger.

Cursor place holders are created by using “\${1}” and can contain more than one. Just number them 2,3,4 etc.

In the example above, I could have entered a second cursor point at the end of the code like so:

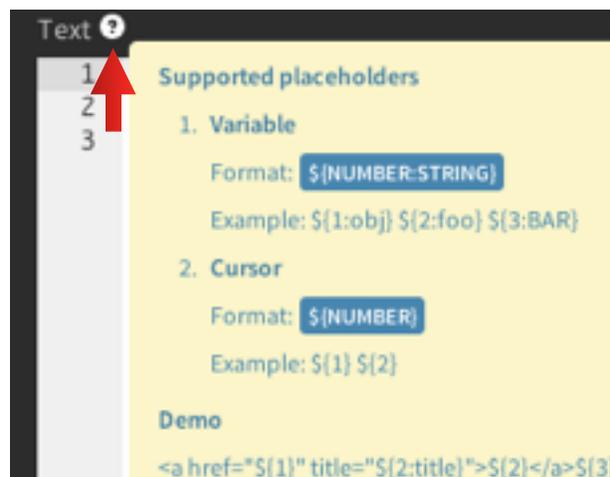
```
<!-- /.${1} ->${2}
```

to jump to the end of the code and keep on going.

*** FYI - Unfortunately there is a conflict with Emmet and this Snippets extension which breaks the use of more than one cursor point. If you uninstall

Emmet :(then this extension will work as intended. This is in the works by the author to be fixed.

For more info on this extension, hover over the ?



Font Awesome - <http://fontawesome.github.io/Font-Awesome/>

Listing of the font names:

<http://fontawesome.github.io/Font-Awesome/icons/>

You can place Font Awesome icons just about anywhere using the CSS Prefix fa and the icon's name. Font Awesome is designed to be used with inline elements (we like the `<i>` tag for brevity, but using a `` is more semantically correct).

```
<i class="fa fa-camera-retro"></i>  
<span class="fa fa-envelope"></span>
```

Examples Page:

<http://fontawesome.github.io/Font-Awesome/examples/>

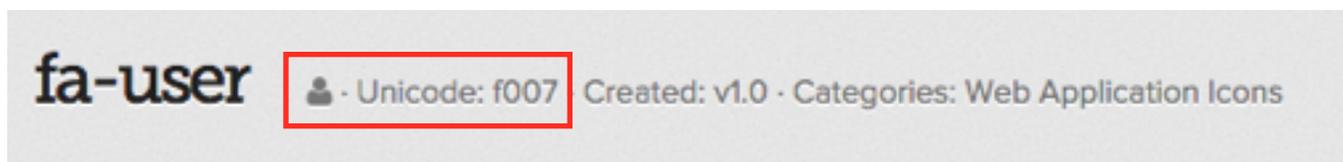
Using the fonts in pseudo classes of before or after:

```
.icon:before {  
  content: '\f007'; !IMPORTANT Put \ before code #  
  font-family: 'FontAwesome'; !IMPORTANT calls the font  
  font-size: 28px;  
  margin-right: 10px;  
  color: #00ADC8;  
}
```

The content is the Unicode value of the icon font. By clicking on each icon in this page:

<http://fontawesome.github.io/Font-Awesome/icons/>

you get the detail about each icon.



Bootstrap Glyphicons

Bootstrap does come with some icons too. In the fonts folder you will see them listed by the name 'glyphicon'.

See this page for more information:

<http://getbootstrap.com/components/>

They are incorporated very similarly, but **do not** have Unicode equivalents.

Glyphicon Specifics:

Don't mix with other components

Icon classes cannot be directly combined with other components. They should not be used along with other classes on the same element. Instead, add a nested `` and apply the icon classes to the ``.

Only for use on empty elements

Icon classes should only be used on elements that contain no text content and have no child elements.