How to organize your HTML, CSS, and Javascript files

A good friend of mine told me that the best system is the one that works better for you. This is only true if you never plan on sharing your work with someone else, or if you do not expect to use other people's code.

In this article I will outline a recommended structure to organize files in your HTML5 application. This is not an attempt to create any kind of standard. Instead, I will make suggestions on how to group and name files in a logical convenient way.

Your Project

Let's assume you are building an HTML5 application. In some cases you may use the root of your server as the main container but for the purpose of this article I will assume you HTML5 application is contained in a folder. Inside this folder you must create your application index file or main entry point.

- **appcropolis-project**
  - my-index.html

Generally, your application will be comprised of HTML, CSS, Images, and Javascript files. Some of those files will be specific to your application and some others can be used across multiple applications. This is a very important distention. To do a effective grouping of your files you must start by separating general-purpose files from application-specific resources.

- **appcropolis-project**
  - **resources**
  - **vendors**
  - my-index.html

This simple separation makes navigating through your files a lot easier. Once you place libraries and general-purpose files inside the **vendors** folder it is clear that the files you will be editing will be located in the **resources** folder.

Aside from your HTML code the rest of the files in your application are mostly CSS, Javascript, and images. Chances are that you already group your application files inside folders that correspond to these kind of assets.

- **appcropolis-project**
  - **vendors**
  - my-index.html

The **js** folder will hold your Javascript code. Similarly, the **images** folder is the place where you should add images that are used directly from the index.html or any other page in your application. This **images** folder should not be used to host stylesheet-related files. Your CSS code and related images should be located inside the **css** folder. By doing this, you can build pages that can easily use different themes and you allow your application to be more portable.

- **appcropolis-project**
  - my-index.html
  - my-contact-info.html
  - my-products.html

The previous example shows the content of the **css** folder. Notice that there is a file named default.css which should be used as your main CSS file. Images used by the default stylesheet should be place inside the **images** folder. If you want to create alternative stylesheets or if you wish to override rules defined in your default stylesheet, you can create additional CSS files and the correspondent folders. For example, you can create a blue-theme.css stylesheet and place all related images inside a **blue-theme** folder. If you have CSS or Javascript code that is only used by one page (in this case my-index.html), you can group page specific code inside .css and .js files with the same name of the page (e.i. my-index.css and my-index.js). Your CSS and Javascript code should be a generic as possible but you can keep track of exceptions by placing them in separate files.

Final Recommendations

Some final recommendations have to be made around folder and file names. As a general rule make sure that you use lower case letters in all folder and file names. When using multiple words to name a file or a folder separate them with a hyphen (i.e. my-company-logo-small.png). If you follow the advice in this article you should be able to combine multiple pages while keeping common resources together and custom code nicely separated.

Finally, even if you do not choose to use the structure recommended in this article, it is important to stick to a convention. It will increase your productivity and more important it will make the work that you do easy to understand by others.

By: Raul Sanchez
Categories: Blog, Web Technology
Posted on: Friday, June 1, 2012