**In-class exercise: Chrome Extension (using jQuery)**

In order to explore the potential for parasitic web projects, and learn about Javascript, we will create a simple Chrome Extension using jQuery. Chrome Extensions are essentially HTML, CSS and Javascript that you can trigger in (almost) any web page when using the Google Chrome browser. These extensions allow us to change the behavior of the websites we, and others, interact with without having access to the websites themselves.

This exercise will introduce you to using Javascript and Chrome extensions.

Due In-class. **Create a simple webpage with a link to a \*.zip file of your Chrome Extension and upload to your class site under exercises. Try considering your extension as a component to your project.**

→

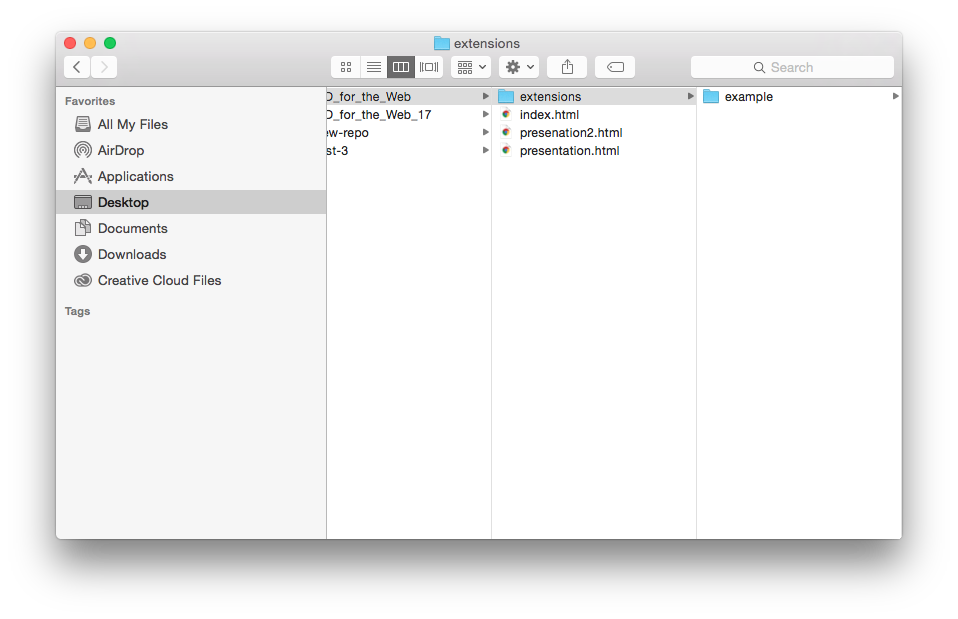
Choose a website that you’re interested in working with. This could be the platform you are investigating or another website that relates to it. It could also be a site you use regularly (Facebook), a media outlet (NY Times), an ecommerce site (Amazon)...

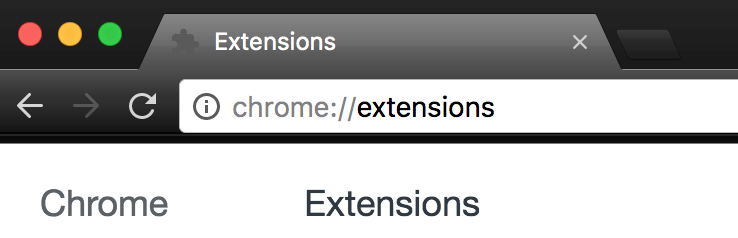
→

Using [this example extension](http://spring2017.wwweb.work/resources/example.zip), affect the page in a general way. Adjust the following: the appearance of the page directly using the “style” attribute (change its background color, font-size, rotate elements, etc.), adjust the appearance of the page using a stylesheet, and adjust the content on the page (swap out an image, or replace a string of text).

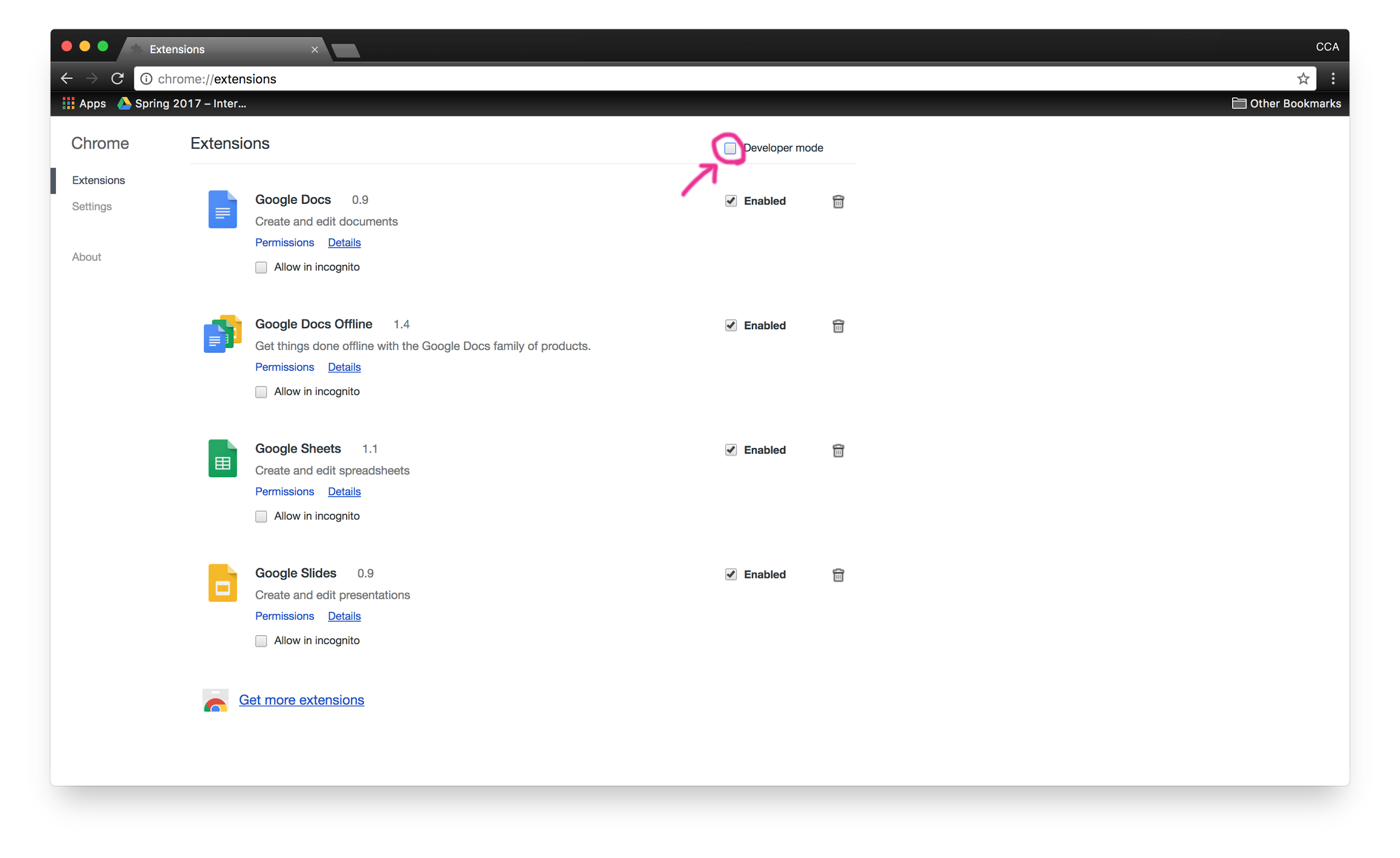
→

To install the sample extension, first download and unzip the file. Then move the unzipped folder (‘example’) into another folder in your class site (GD\_for\_the\_Web>extensions>example).

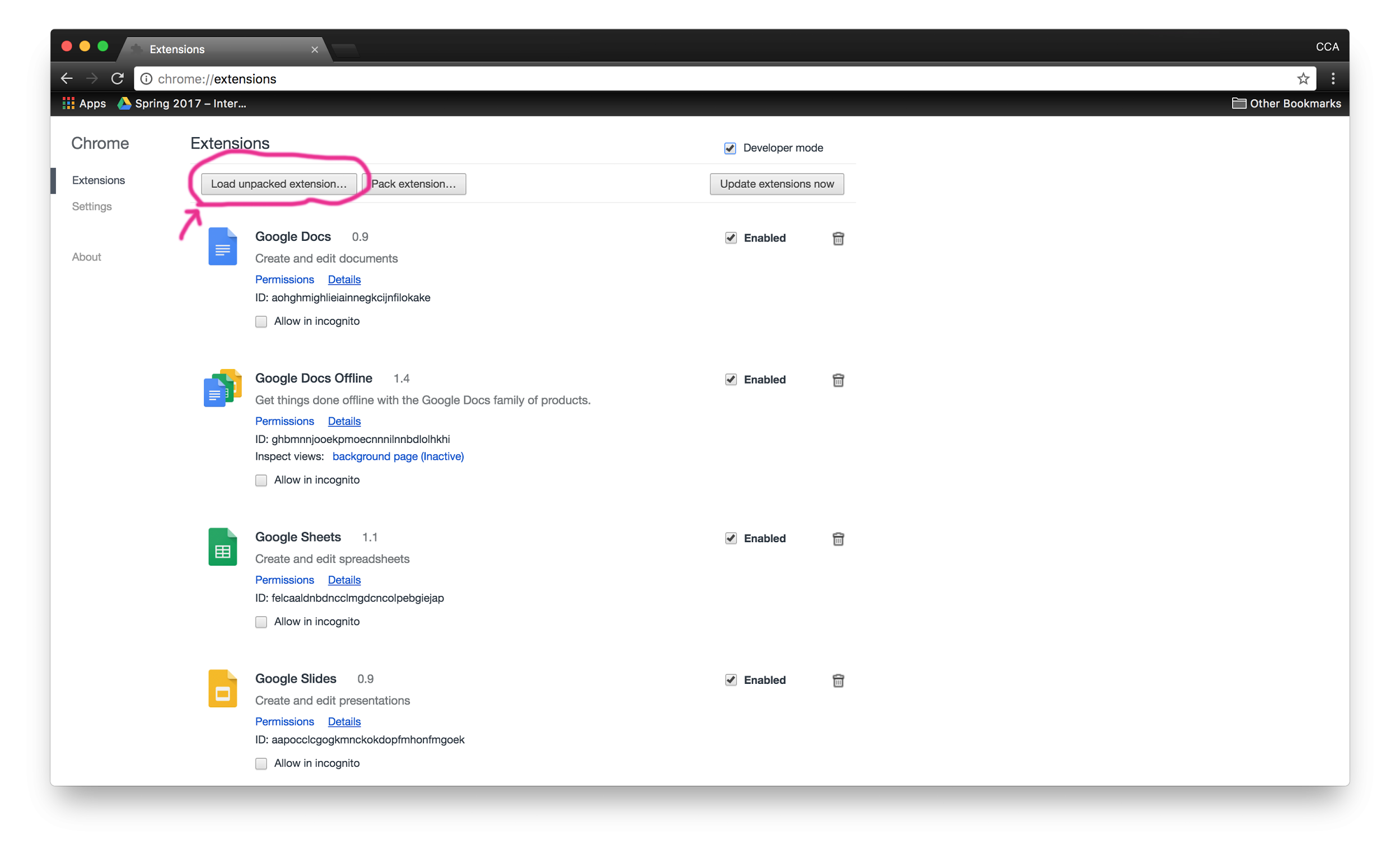




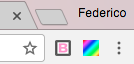
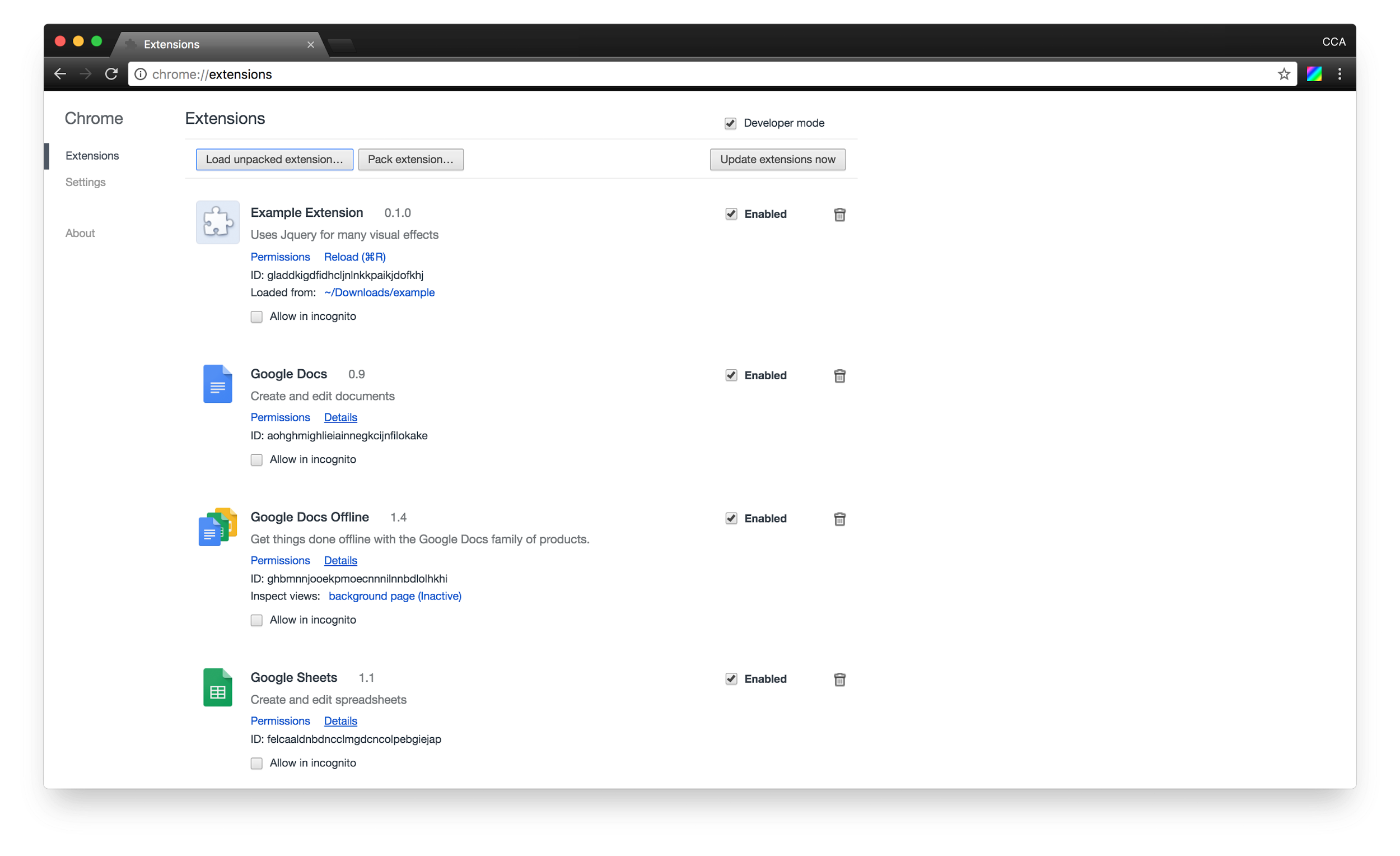
Next, type ‘chrome://extensions/’ in your Google Chrome web browser (*you must use Google Chrome for this to work*) and enable Developer Mode by clicking the checkbox at the top right of the window.



Then, load the example extension by clicking “Load unpacked extension…”. And selecting the ‘example’ folder.

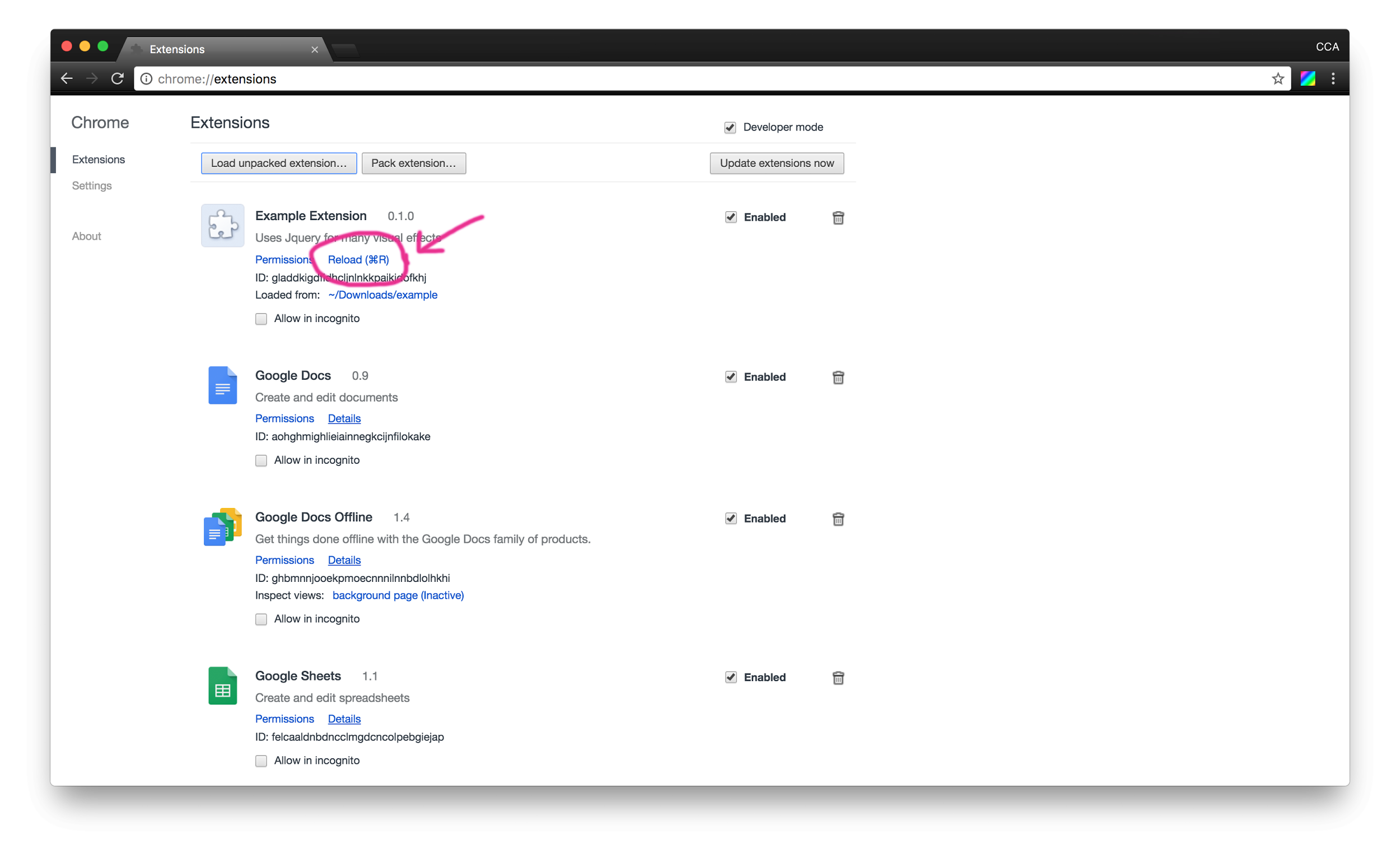


The example extension will appear in your list of Enabled extensions, in addition to the extension’s icon appearing at the top right of your window. This means it is active and in use.



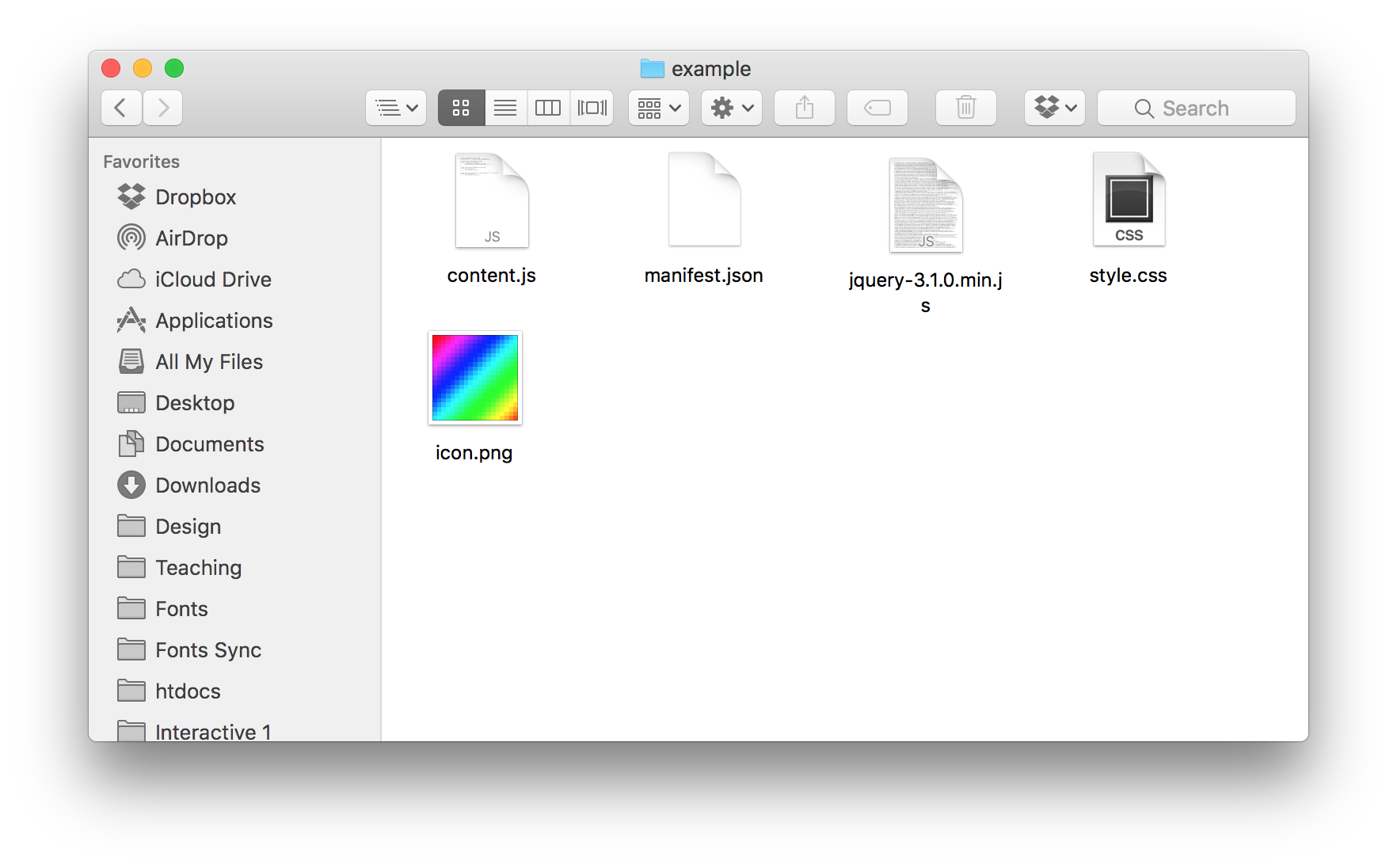
As you navigate the web, the Chrome extension will automatically function. Try testing it by visiting web pages and clicking on the pages ‘body’ and ‘img’s.

**Whenever you edit your extension, return to this page and ‘Reload’ it so it is up-to-date.**

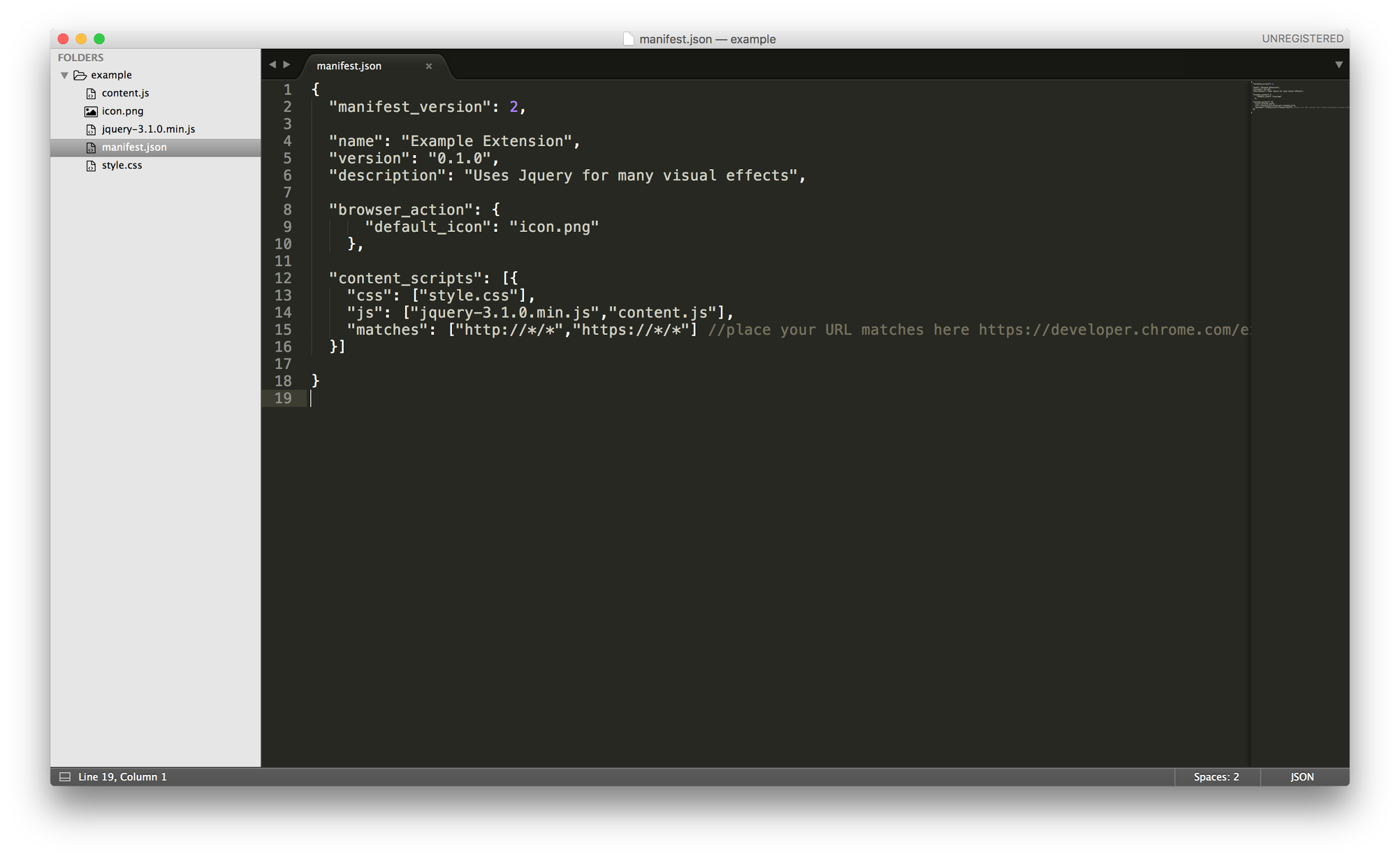


→

To edit your Chrome extension, it’s important to understand how it is put together. In the example extension there are 5 files. The icon which appears in your browser toolbar (‘icon.png’), the JQuery Javascript library (jquery-3.1.0.min.js), a CSS stylesheet you can reference from the webpage you are visiting (style.css), a description of the extension (manifest.json), and the code the extension runs (content.js).



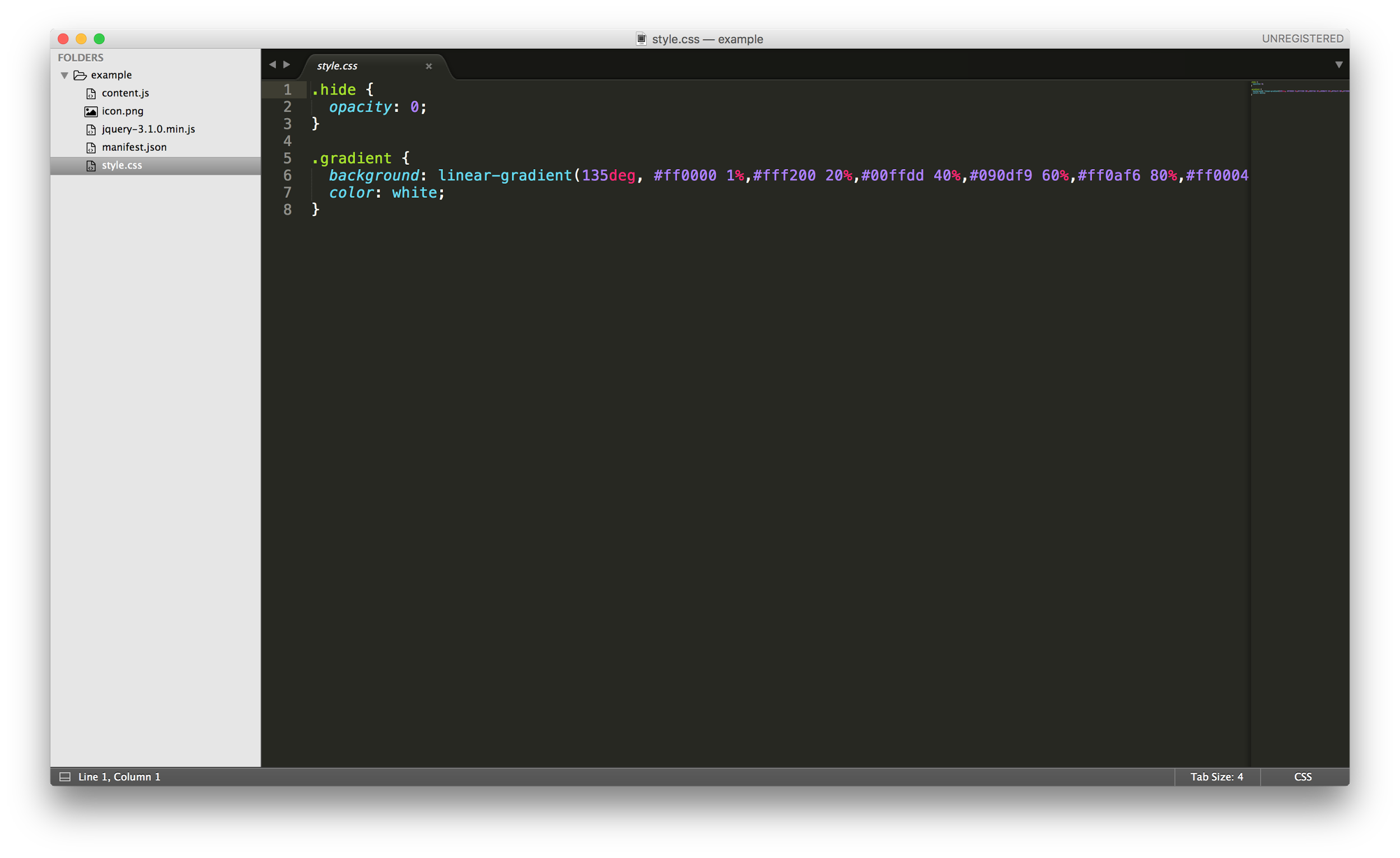
The ‘manifest.json’ file holds the metadata of the Extension, and tells the browser which icons to use, and the files to load in order to trigger the extension. Make sure to update the metadata here (name, and description) as you modify your extension.



‘Content,js’ is the actual code that is triggered as you browse the web. This file is written in Javascript and built upon the JQuery library. At the moment, the file by default replaces all the images in the page and the '@' signs with '$' signs when the page loads. When you click on an image or the body tag it applies CSS to the page that is found in ‘style.css’.

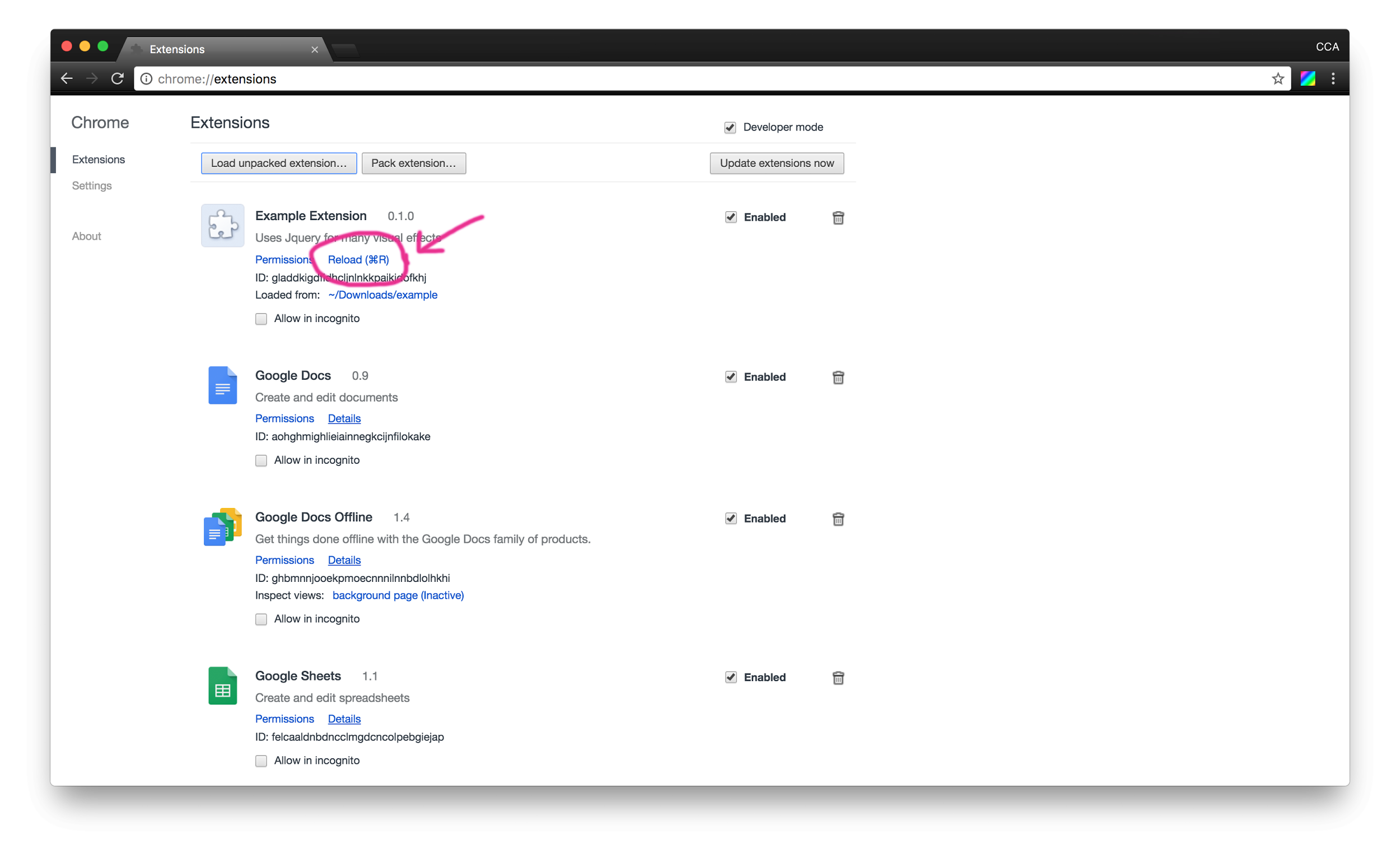


Try adjusting the Jquery selectors and its effects to test the code. For instance, try swapping out the ‘src’ value for the ‘img’ ‘attr’. Change the text which gets replaced. And call a new CSS class on click.



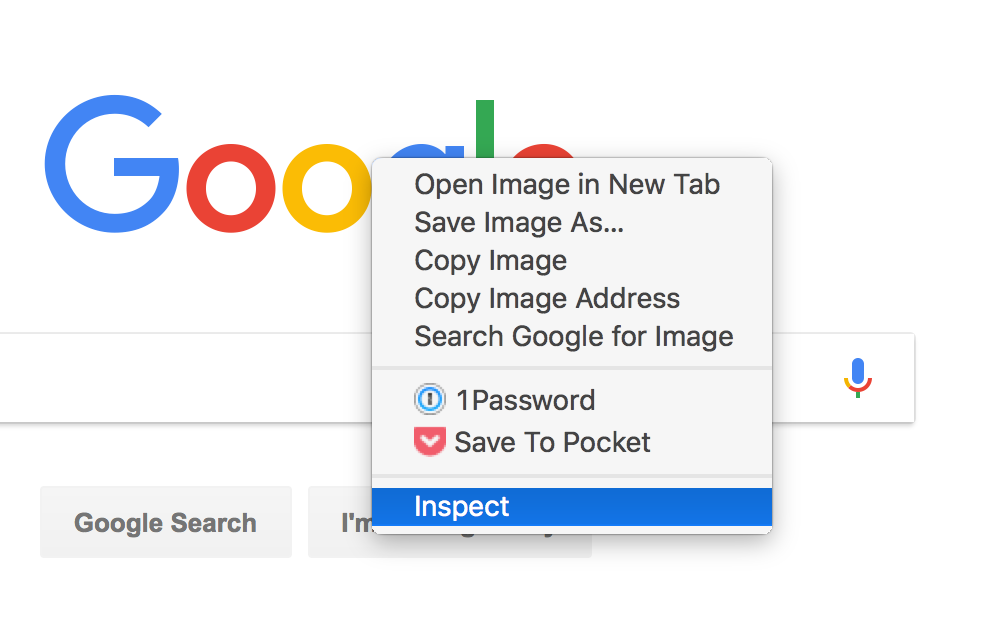
“style.css” is a stylesheet for your extension. This stylesheets works just like it does on your own websites, but you are applying it to websites you are visiting. Sometimes, your CSS will conflict with that of the page you are visiting. In general, these Chrome Extensions require a lot of testing! Now, when you click on any image on a webpage, it adds the ‘.hide’ class to that image found in this CSS file. ‘.gradient’ is applied to the body when you click on the page. Try creating CSS for a new ‘class’ and applying it to the web page via the ‘content.js’ file.

**Remember: Whenever you edit your extension, return to the ‘chrome://extensions/’ page and ‘Reload’ the extension so it is up-to-date.**

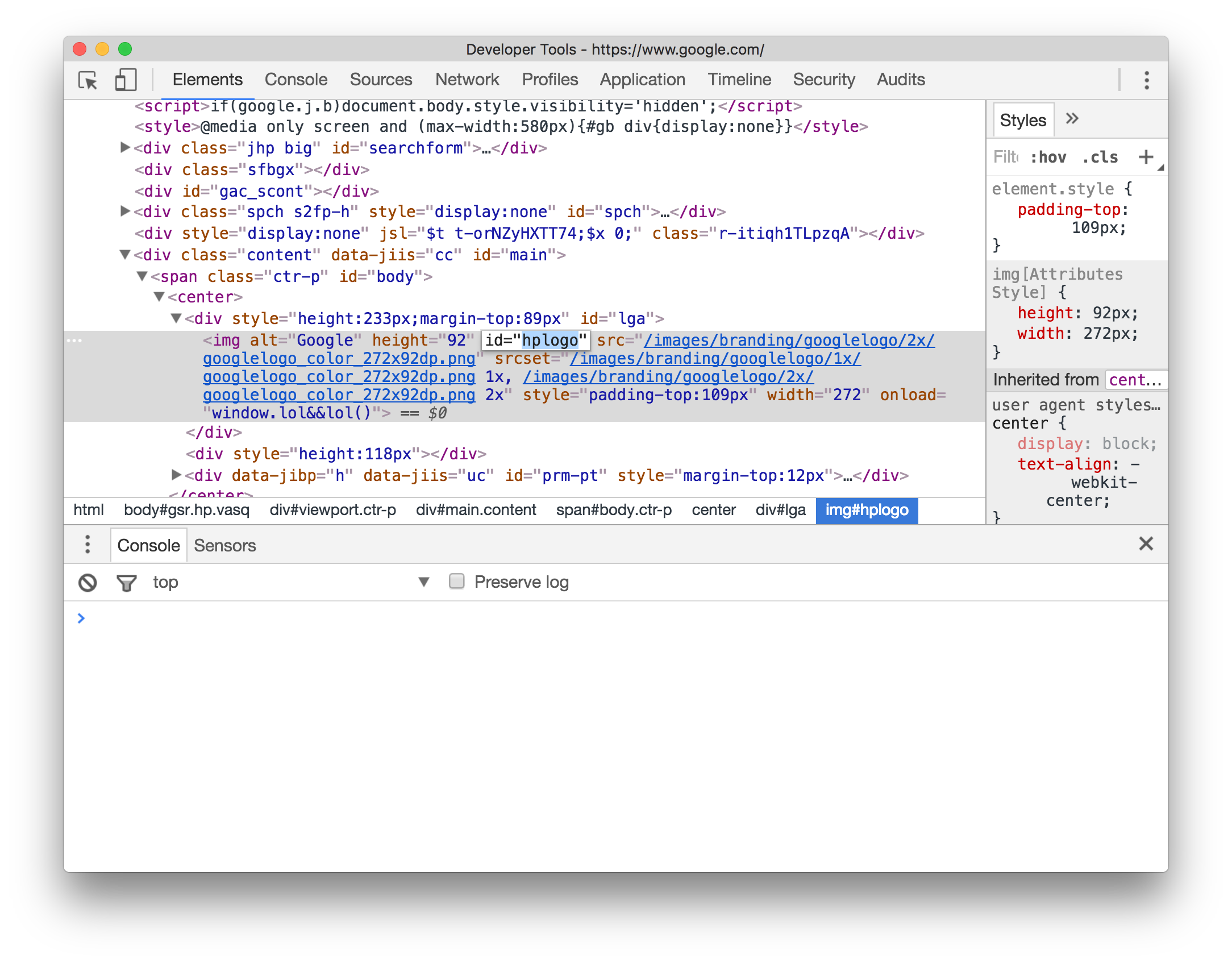


→

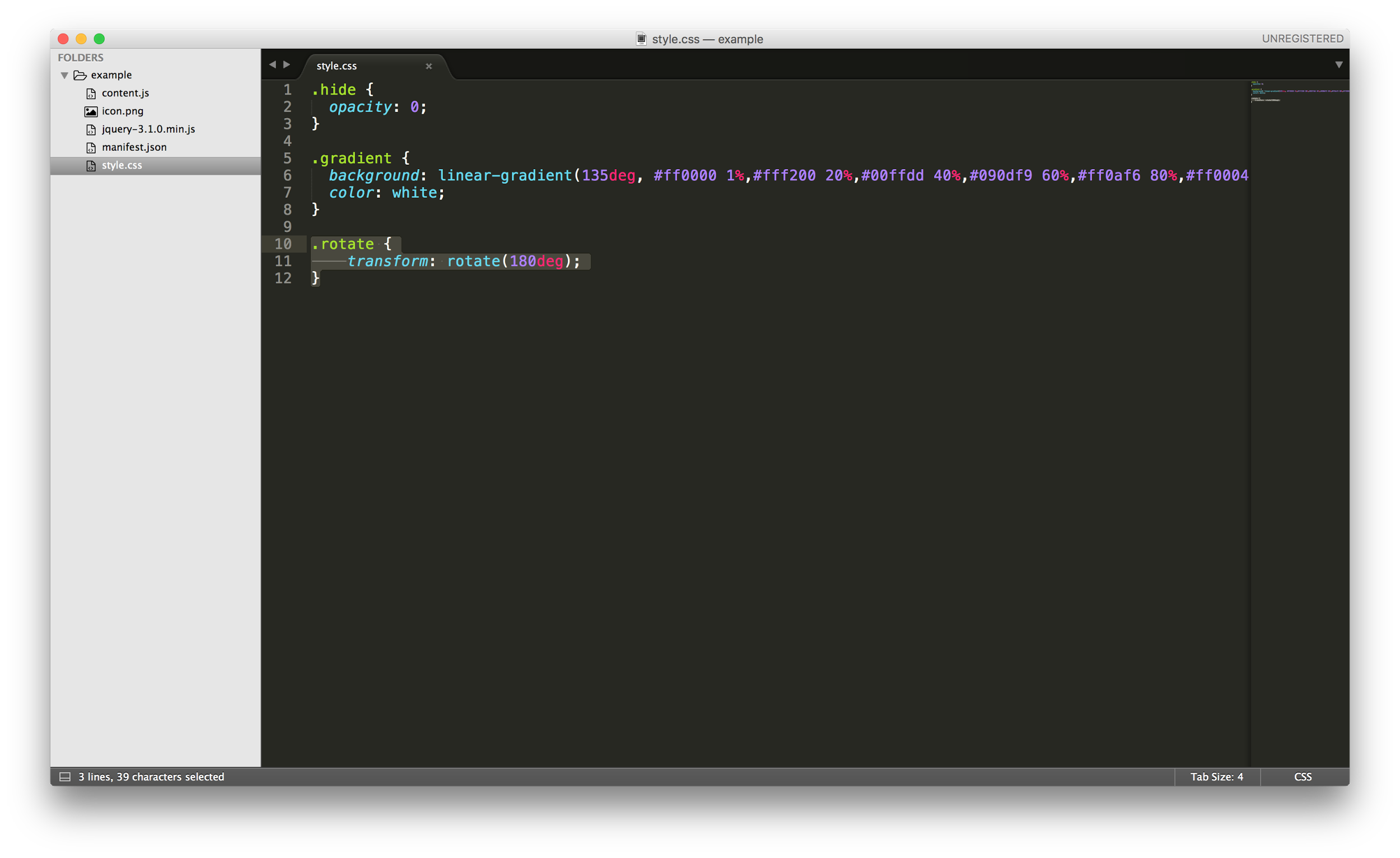
After you’ve tried these more general interventions, be more specific in what you’re targeting. Using the Chrome inspector (right click on the page and select “Inspect”). This will bring up the Chrome Web Inspector.



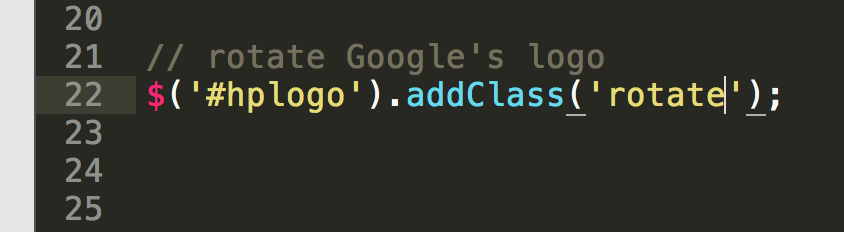
Using the inspector, you can analyze the structure of a webpage – testing for errors using the ‘console’, and identifying classes and IDs that would be helpful when making your Chrome extension. For instance, on google.com, it’s logo has the ID of “hplogo”.



As a test, in your CSS file, try adding the class ‘.rotate’ which rotates an element 180°.



Then applying that class to ‘#hplogo’ using the ‘content.js’ file.



Now visit google.com to test your extension.

Find something specific to the page you are analyzing, and write Javascript to target that element in particular.

→

Once you have made your extension target something specific, update your Chrome extensions metadata and icon. Then Zip your Chrome extension, and create a simple webpage that links to that file with a short description of the Extension. Then upload this page and the zipped extension to your class site and link and if it makes sense link it to your project for next class.

References:

Technical

* <https://developer.chrome.com/extensions/getstarted>
* <https://robots.thoughtbot.com/how-to-make-a-chrome-extension>
* <https://www.sitepoint.com/create-chrome-extension-10-minutes-flat/>

Examples

* <https://supermute.feeltrain.com>
* <https://panopticlick.eff.org/>
* <https://www.eff.org/https-everywhere>
* <https://www.eff.org/privacybadger>
* <https://chrome.google.com/webstore/detail/wayback-machine/fpnmgdkabkmnadcjpehmlllkndpkmiak?hl=en-US>
* <http://projectseen.com/>
* <http://citizen-ex.com/>

JQuery Reference

* <https://api.jquery.com/category/events/>
* [Click a button and toggle](http://jsfiddle.net/w0a3Lgx6/)
* [Mouseenter, mouseleave](http://jsfiddle.net/cvyxp15z/) and [Hover](http://jsfiddle.net/o1udqwbm/)
* [Scroll and alert](http://jsfiddle.net/p0uxmv0n/)
* [Accordion slideToggle](http://jsfiddle.net/70Lnbvor/)
* [Turn on and off lights with button](https://jsfiddle.net/mf1vdqb0/9/)
* [jQuery Examples](http://jsfiddle.net/845rzqy2/13/)
* [Change background color](http://jsfiddle.net/r0g0y7f1/8/)
* [Scroll to anchor smoothly](http://jsfiddle.net/9SDLw/)
* [Draggable div](http://jsfiddle.net/vxepqwd4/)