

Chapter 1 : Chrome DevTools for troubleshooting HTML and CSS | Rich Finelli

Sometimes CSS can be frustrating. Learn about some tricky properties, the common issues they can cause and how to solve them. CSS is a mess. First introduced in ~, it was meant to style basic text documents. Not websites. Not applications. Text documents. It has come a long way since then.

Inheritance[edit] Inheritance is a key feature in CSS; it relies on the ancestor-descendant relationship to operate. Inheritance is the mechanism by which properties are applied not only to a specified element, but also to its descendants. Descendant elements may inherit CSS property values from any ancestor element enclosing them. In general, descendant elements inherit text-related properties, but box-related properties are not inherited. Properties that can be inherited are color, font, letter-spacing, line-height, list-style, text-align, text-indent, text-transform, visibility, white-space and word-spacing. Properties that cannot be inherited are background, border, display, float and clear, height, and width, margin, min- and max-height and -width, outline, overflow, padding, position, text-decoration, vertical-align and z-index. Inheritance can be used to avoid declaring certain properties over and over again in a style sheet, allowing for shorter CSS. Inheritance in CSS is not the same as inheritance in class-based programming languages , where it is possible to define class B as "like class A, but with modifications". However, it is not possible to define a CSS class B like that, which could then be used to style multiple elements without having to repeat the modifications. Given the following style sheet: The style sheet h1 has the color pink, hence, the em element is likewise pink. Whitespace between properties and selectors is ignored. In addition to formatting CSS for readability, you can use shorthand properties to write out the code faster which also gets processed more quickly when being rendered, [16] like so: Normal flow Inline items are laid out in the same way as the letters in words in text, one after the other across the available space until there is no more room, then starting a new line below. Block items stack vertically, like paragraphs and like the items in a bulleted list. Normal flow also includes relative positioning of block or inline items, and run-in boxes. Floats A floated item is taken out of the normal flow and shifted to the left or right as far as possible in the space available. Other content then flows alongside the floated item. Absolute positioning An absolutely positioned item has no place in, and no effect on, the normal flow of other items. It occupies its assigned position in its container independently of other items. If an item is positioned in any way other than static, then the further properties top, bottom, left, and right are used to specify offsets and positions. Static The default value places the item in the normal flow Relative The item is placed in the normal flow, and then shifted or offset from that position. Subsequent flow items are laid out as if the item had not been moved. Absolute Specifies absolute positioning. The element is positioned in relation to its nearest non-static ancestor. Fixed The item is absolutely positioned in a fixed position on the screen even as the rest of the document is scrolled [17] Float and clear[edit] The float property may have one of three values. Absolutely positioned or fixed items cannot be floated. Other elements normally flow around floated items, unless they are prevented from doing so by their clear property. This evolution gave the designer more control over site appearance, at the cost of more complex HTML. Variations in web browser implementations, such as ViolaWWW and WorldWideWeb , [24] made consistent site appearance difficult, and users had less control over how web content was displayed. The style sheets could therefore not be linked to documents on the web. It organized a workshop toward that end chaired by Steven Pemberton. Lie and Bos were the primary technical staff on this aspect of the project, with additional members, including Thomas Reardon of Microsoft , participating as well. Early in , the ERB was split into three working groups: It was published as a W3C Recommendation on May 12, CSS level 3, which was started in , is still under development as of IE 4 and Netscape 4. It was more than three years before any web browser achieved near-full implementation of the specification. Microsoft Internet Explorer 5. This is a set of rules that specify certain aspects of the sizing and layout of components of a web page. Such inconsistencies and variation in feature support made it difficult for designers to achieve a consistent appearance across browsers and platforms without the use of workarounds termed CSS hacks and filters. Variations[edit] CSS has various levels and profiles. Profiles are typically a subset of one or more levels of CSS built for a particular device or user interface. Currently there are profiles

for mobile devices, printers, and television sets. It returned to Candidate Recommendation on 19 July and then updated twice in However, because changes and clarifications were made, it again went back to Last Call Working Draft on 7 December Taxonomy and status of CSS3 modules. For other uses, see CSS3 disambiguation. At CR stage, implementations are advised to drop vendor prefixes.

Chapter 2 : CSS: Width problems - Stack Overflow

Objective: To be able to diagnose common HTML and CSS cross browser problems, and use appropriate tools and techniques to fix them. The trouble with HTML and CSS Some of the trouble with HTML and CSS lies with the fact that both languages are fairly simple, and often developers don't take them.

To help you sort through the options, the tutorial below focuses on some of the common challenges you may face when editing CSS in Dreamweaver. But when you look at the pages in your site, many headlines still display in a green color from your old design. To use this feature, first click to select an element, such as a headline, then click on the Current tab at the top of the CSS Styles panel. The Properties pane at the bottom of the panel displays how styles are applied to the selected element. In the example shown here, the green color in the headline is being caused by a class style called. In this case, if you remove the. Click to place your cursor in the element, then find the style in the Tag selector at the bottom of the workspace in this case, the. Right-click and choose Set Class to None to remove a class style. Using the drop-down lists in the Find and Replace dialog, you can specify if you want to change the style name in all of the files in a site, only in the current document, in open documents, or within selected text. As you see here, the attributes are automatically filled in by Dreamweaver when you use the rename option, making it easy to use the correct search string to change the style name of the style in the code. Right-click the Tag selector, and choose the Quick Tag Editor. That brings up a little dialog where you can edit the style name without having to view the code and thanks to shortcuts, you can select the new name from a list of available styles. If you opt to type, make sure you do it carefully. The name has to be exactly the same. Use Div tag to center background If you follow the box model to create centered CSS designs, you probably create a div with a style called something like container, and then insert all of your content, including other divs, inside that main container div. The predesigned CSS layouts in Dreamweaver follow this model. If you want your background image to be centered, the trick is to set the image as the background of the container div instead of setting it as the background of the entire page. When you do, Dreamweaver automatically creates styles named a: Create an Advanced or Compound style To create a second set of link styles for a navigation bar, create the links inside a div and then include the ID for the div in the name of the link styles. Note the space between the ID name and the tag name. These advanced styles instruct the browser to only apply the style definition when a link appears inside an element with the ID navbar. Not all browsers display CSS the same One of the greatest frustrations when it comes to working with CSS is that not all Web browsers display styles the same way. For example, early versions of Internet Explorer display padding differently than Firefox, which can throw off the alignment of elements. Unfortunately, the differences among browsers are far too complex to cover in a step or two in this article, so the tip here is to use the Browser Compatibility feature to identify possible problems and tips for how to resolve them. This handy addition to Firefox makes it possible to disable, view, and even edit the CSS on any webpage.

Chapter 3 : 4 simple techniques to quickly debug and fix your CSS code in almost any browser | Big Empl

If you would like to learn more about CSS troubleshooting, check out my new course called CSS Troubleshooting in 6 Easy Steps on Learnable. The course is designed for beginners and those who have.

A class may be produced by the software, e. Similarly an id may be produced by the software, e. In the case of conflicting style settings for a piece of content, the resulting setting depends primarily on the indication "!". Secondly, if both are important, the user wins, if neither is, the author wins. Tertiarily it depends on specificity. Only lastly it depends on order between and within style sheets: There were some bugs in CSS support in earlier versions. Some wikitext elements allow you to insert CSS styling directly into them. An example is the table syntax: Tables MediaWiki existing styles[edit] You may wish to use a style type that is already predefined by MediaWiki, or the site that you are visiting. You can also create a style that is unique to your page. Vector is the default style, you can view it at: Tips and tricks[edit] Non-display[edit] In an embedded page, one can hide comments in one version, and show them in another view. One extreme "style" for a text is not displaying it, with. Non-displayed links do not work as opposed to links in a very small font. It cannot be used to remove text in expressions for template names, parameter names, parameter values, page names in links, etc. It will make all hidden elements appear. Non-print[edit] One can exclude content from being printed if the browser supports CSS by declaring the content to be of the "noprnt" class: If you understand it, please provide an image. Identified blocks using that class: For one or more of the possible class names the style of that class can be defined. If the class is undefined it is ignored, so the standard style is used. In the simplest case we have e. If the parameter value is "def" it applies. If a page for general use only makes sense when styles are defined for certain classes, then these have to be specified in the page MediaWiki: If the value of parameter 3 is a display style other than "none", that style is applied. Wiki headings use the following default CSS:

Chapter 4 : Help With HTML & CSS

CSS (Cascading Style Sheets) are bits of code that influence the presentation or the look of your page's HTML code. In WordPress, the CSS styles are generally found in a file called `calendrierdelascience.com` in the specific Theme folder you are using.

Also, backup the files you are working on as you try different things so you have some places to go back along the way. You can do "live" CSS testing without editing your Wordpress files. If you have the means, it is much quicker and safer to do your CSS testing and troubleshooting "on the fly" using `e. The Web Developer` extension for Firefox can help, too. At the least, you will get a basic overview of what CSS is, the impact it has on the HTML or structure of your page, and learn some jargon to help you ask a more informed question on the forums. You will also need to know some basic terminology to help you express your problem to others. For help on modifying those, check out `Using Themes` for more information. CSS selectors names are generally grouped into three specific references: `And some more rambling here`. This example indents the quote on both sides and puts a blue line on the left side of the quote and makes the text italic. Certain themes have their CSS styles in more than once place, the Kubrick theme being one of them. This means looking under the hood. Under the pretty hood of your web page, the nice layout you see on your Internet browser, is a whole bunch of code with strange and foreign sounding references. You know that all of that junk makes the car go, but what the heck is all that mess under the hood? Another page will pop-up either inside another browser window or inside of a program that comes with your operating system called Notepad, or some variation thereof. This is your HTML page which structures your page. To view your CSS, either double click on the file name or type in the specific link to the file in your web browser such as: `This is often complex and confusing code.` To view the HTML, view a generated page, such as a sample post. The `Codex` page on using themes has more information on how to view your Theme Templates and find out which Template is associated with which section on your page. The problem-solving techniques herein describe how to change the CSS to influence the layout of the page. To make actual changes to the Themes, check out `Theme Development`. To start, you have to know what you are looking at and where to find the basic elements to help you find the culprit. The HTML page features all the structural code that sets up the "grid" into which your page sits. Think of it as a map with notes written all over it. The notes are actually pointers to the directions which are found within the CSS file. Below this will be the layout "body" of your web page. From here on, every bit of information is critical to the structure and design of your page. For example, you may see the following: It begins with the division called "page" which sets up the look of the entire page. If you look in the CSS file for page selector, you will see the presentation styles associated with it. It is followed by the header division which includes a heading H1 with the title of your site. This particular layout is based upon the default Theme for WordPress v1. After that comes a division with a CLASS reference called "description" which is the place you have a subtitle or description of your site. Again, look in your CSS to find. The "content" and "widecolumn" layout the overall look of the containers holding your "post". And, as you will see in the next section, this is one of the major places where CSS layout problems begin! The CSS Parent and Child Relationship One of the biggest problems in designing web pages is understanding where a problem occurs and whose influence might be affecting the problem. The parent page was telling the child header what to do and it wanted to do something else. If we changed the right and left margins of page, it eliminated the margin problems for the header. The parent page continues its influence and now all of the content is spread across the whole page width. A change needed to also happen with the left and right margins of the content so the wide margins are back in place. To make the whole family happy, the new margins looked like this: To isolate and identify your various CSS sections, divisions, and classes, here are some simple tricks. Before you begin, be sure and make a backup of all of your main files including your CSS to make sure you have something to recover from if this gets out of control. Once you have identified the culprit and fixed it, make sure you remove these testing features so your web page will look "normal" again. You should now see a different colored box around each of the different sections: This is a section of rambling text that goes on and

on. This is another section that has been highlighted in a red box. This is the rest of the text back to normal. If the problem you are having is in the blue box, then you know where to start solving your problems. There are browser extensions and add-ons that offer this technique, too. Highlight Sections Besides putting boxes around the different sections to isolate the problem CSS or HTML, you can dramatically change the colors of the content to make the problem "jump" right out at you. By changing the text color or background color of a section, you will spot it immediately when you view the screen. Be sure and make note of the original colors if you change them during testing so you can go back to them. And make frequent backups! To change the background color of a section, you can add background: This is some text that goes on babbling here and there. This is some text with the background color changed so you can see it. The entire header division will now feature a pink background. Validate Your Source Code Sometimes the smallest detail can send your page out of whack. A misspelled tag rhref instead of href , a forgotten closing tag, a missing attribute, or even the wrong attribute can send your page into a design tail spin. Most online validators let you either type in the URI link to your site to initiate the validation process, or may even allow you to paste in code or upload a file to have it inspected. WordPress, by default, validates its default coding, but if you are making modifications, the slightest slip can screw things up. Many validators will even recommend making some changes, but find your problem before you start creating new ones. There are two Slash and Burn techniques. If you are weak of heart, check out the second one. Make backups of all of your files. Save this source file as a text document called "junk. It will remain open during this entire process. Copy your CSS file to the same test folder. If you are having problems with the graphics, copy the graphics folder or the graphics to the test folder. In the test folder, double click on junk. You should see the general layout of the page with the graphics, if appropriate. If not, double check the link reference to the style sheet. Move to the section above a section which includes opening and closing tags such as: You should see the removed section missing. Check below to see if this fixed the problem or if it went away. If yes, this section is your problem. If not, go to the next step. If the problem is still there, move back to the junk. Move to another section above or below this point and repeat steps 8 through At some point in this process, you will see the problem either fix itself or disappear. Begin with large sections and when you find the large section problem area, break it up into smaller pieces. Eventually, you will isolate the area that is causing you grief. Gentle Slash and Burn To use the gentle version of slash and burn, instead of deleting the sections as shown above, cut and paste them into Notepad or another text editor so they are protected in case you get distracted from the cut and paste process. ALWAYS back everything up as you go along, just in case which happens a lot more than you might think! Common Errors We all make mistakes. Here are some of the most common problems that creep up with CSS. Missed Spellings Just so you know, leftt is not the same as left and this could be the reason something is on the right instead of the left side of your page. Missed spelling errors are common and easy to overlook. Luckily, CSS validators can often catch these tiny mistakes for us. Forgotten Details As creative as you can be with CSS, there are some ground rules you have to follow. Miss one of these little details and nothing will happen, or strange things might. CSS validators will usually catch these little forgotten details for you. Luckily, you can usually see these immediately upon viewing the page, so just cut and paste them in the right tag Of course, you can refer to your frequently backed-up file to get the lost code. Double check which modular section you are supposed to be working on all the time. And if you mess one up by accident, there is always that faithful backup file to make things new again. If there are two references to the same selector with conflicting information, it has to decide which one it will use.

Chapter 5 : CSS in HTML Email

Have you created custom CSS for your forms, but not been able to see those styles when viewing your site? In this tutorial, we'll walk through options to troubleshoot why your CSS is not working and offer possible solutions.

What is margin collapsing? Float clearing, an old battle I think this has to be the most common wat? Basically, when an element only contains floated elements, it collapses on itself. This is due to the fact that floated elements are kind of pulled out of the flow so the wrapper behaves as if he has no child at all. There are a number of ways to fix this. Back in the days, we used to add an empty div with clear: Then, we replaced the div by a hr tag. Then Nicolas Gallagher came with a new way to clear floats from the parent without touching the markup at all. After a lot of discussions and tests to bring it down to the minimum amount of characters required to make it work, here is the latest version: This is the simplest and cleanest way to deal with floats.

How to fight inline-block spacing? Nowadays, more and more front-end developers get rid of floats for inline-blocks when they have the option to. They can be sized, they can have margins but their default width depends on the content instead of being full parent width among other specifications. Problem is since they are half-inline, they are spaced from each other by the width of a blank character. With a default 16px baseline with a regular font, that is 4px. Anyway, this can be annoying when layouting elements. Thankfully, there are a couple of ways to get rid of these annoying spaces, each of them with their strengths and weaknesses. To be totally honest, there is no perfect solution yet. The first way to fix the problem is to simply remove the spaces. Perhaps we can reorganize our tags instead of putting them all on the same line so they remain readable: The code is readable and it works fine. Even if it seems odd at first, you would probably get used to something like this. I personally use this method when I have to remove the gap between inline-block elements. This leads us to CSS-side solutions. Plus with some fonts, you might have to go slightly lower than Adjust to your case. The main problem is that this fails in Internet Explorer 6 and Internet Explorer 7 which do not like negative margins. Plus, we have to remove the left margin of the first element to make our children perfectly fit into the container. So, definitely not the best solution. As I said earlier, I would probably go with the comment way. If you feel like this is all too complicated, you may get back to floating your elements, or better using flexbox.

Understanding absolute positioning Positioning is tricky and has always been. Most beginners struggle when positioning elements on a page. They often mis use the position property. This property defines how an element is able to be moved with offsets top, right, bottom and left. It accepts four values: Actually, this is the intended behavior not by you, but definitely by the browser. The keyword here is context. So what is the context? It is the first not static ancestor. It can be the direct parent. Or the parent of the parent. Or the parent of the parent of the parent. As long as it is the first which is not static. This is a tricky concept to comprehend especially for a beginner, but once you get this you can do pretty much whatever you want with absolute positioning without crying out loud because everything is a mess. Here is a quick demo to illustrate what we just saw. Two parents, each one with a child absolutely positioned with top: On the left side, the parent has position: On the right side the parent is static fail. When to use height: To answer this question, it is important to understand what height: So if you want your main container to have the height of the window, setting height: Because the parent of your container body has a default height of auto, which means it is sized according to its content. Then, you can try adding height: Because the parent of body html has a default height of auto, which means it is sized according to its content you get the idea. Now what if you try to add height: To put it simple, it is the browser window. So if you set height: As simple as that. To sum up our story with a tiny bit of code: What if the parent has min-height and no height? Roger Johansson recently discovered that there was an issue with height: First of all, a quick reminder: Now let me tell you a little secret. When adding padding and borders to your width: Because of padding and borders. And because width should have been called content-width. Please consider the following demo to see what I mean. The parent has a width of 25em. Eeerrr, Houston we have a problem. There are four ways to fix this. The first one and definitely the best one is to avoid setting width: The child element is a block level element which automatically expands to the width of its parent without the issue seen above. We could avoid using width: Needless to say this solution sucks since

we have to compute the width manually. We need a better way! The third one would be to use calc to do the calculation automatically: The fourth idea is to use box-sizing: Basically, it changes the box model so the width property is actually set to the total width of the element, borders and padding included. The best news is the browser support is very good everything except IE 7- and Opera And for unsupported browser, we can still use a polyfill. How not to screw up z-index? All boxes in a page are positioned in a 3 dimensional space: At first, this seems very simple: Unfortunately, things are more complicated than that. The z-index property has no effect on a static element. In order to be able to move an element on the Z-axis, you have to define it either relative, absolute or fixed. So the first thing to do is to make sure your element has a position assigned before even thinking of applying z-index. Now, the thing to know about z-index is that all elements in the DOM are not placed on the same layer. It means z-indexing an element to a very high value may not be enough to make it appear on the foreground. This is called stacking contexts. To put it very simple, a stacking context is a kind of group based on a single HTML element within which all child elements share the same stacking order thus the same Z axis. Changing the Z value of those elements may make them overlap each other in the way you want. In the same stacking context, here is how elements are displayed back-to-front from the CSS specifications: This would have been far too easy! The thing is, every stacking context has its own Z scale. Basically, an element A in a stacking context 1 and an element B in a stacking context 2 cannot interact through Z-indexes. That means if element A is part of a stacking context at the bottom of the stacking order, there is no way to get it to appear in front of element B in a different stacking context which is higher in the stacking order, even with a very high z-index value. I feel like we could make a keyword for z-index:

Chapter 6 : Troubleshooting Conflicting Styles and Other CSS Challenges in Dreamweaver -

Tour Start here for a quick overview of the site Help Center Detailed answers to any questions you might have.

Not knowing the tools available to figure out why its not working is more frustrating. And it usually tells me. This opens the developer tools with the Elements tab open, showing html on the left and css on the right. The Elements Panel This is the default tab that opens and is probably the most important for CSS troubleshooting and designing in the browser. It also does this awesome highlight feature where it highlights the element you selected showing the space it occupies plus the margin and padding. That in itself is very useful especially when you get in to the territory of floats and collapsed elements due to floats. The Styles panel on the right is even more helpful. The most specific selector is going to be on top, and in this case is. The second selector from the top h1, h2, h3,. For instance, if i want to change the color of the border-bottom I can simply click on the color swatch and a color picker appears that allows me to choose my desired color. Changes do not persist by default. So as soon as you refresh the page you go back to the original styles. Circle back to that later when you have a need for it. I think everyone has issues like this from time to time. A lot of time its a specificity issue. This is nice because it also shows you the exact line numbers to find the more specific selector that is overriding it. Typing Problems The other common problem is you just type in the property name incorrectly. When you hit tab it will allow you to enter a new property. It also is smart enough to offer value suggestions when you tab to the value field. This speed is often nice to preview styles quickly. You then can copy the style changes directly from this panel and put them in your css file if you determine you want to accept the changes. This is great for previewing changes to existing selectors in your style sheet. Or, it already has a generic rule set, but maybe you want to create a more specific rule set that only applies to the element in certain places. Accessing hover states Making cool hover states is kind of the funnest part of CSS in my opinion. Sometimes these interactions can become very complex and its worth creating them in the browser with the hover-state permanently on. This is something that can be very helpful. Click on the icon in the top right corner, and select the hover checkbox. This will show you the hover styles that are being applied to any given element and allow you to modify and add more. Accessing Computed Values DevTools shows declared as well as inherited styles. When this happens its time consuming to find the rule set from the list of selectors that is truly having an impact. Switch over to the Computed tab to see an alphabetical list of computed properties that are affecting the element. Just expand the property name using the little carrot to see the selector and line number in the stylesheet that is applying that property. This little red error X icon with a number next to it shows up when Chrome finds problems with your website. Clicking on the error icon or switching over to the Console tab will provide further info. So maybe I typed the file name incorrectly. Device Emulation Mode Chrome has so many awesome developer tools and the device emulation has to be one of the cooler new features. Its hard to test your site on lots of devices. Although not perfect, emulating your site as various devices in Chrome is useful. Just click on the little phone icon in the top left of the developer tools. The entire browser window will change and give you the opportunity to select from a number of devices. Conclusion The Chrome DevTools are awesome and I tend to prefer them over other browser development tools.

Chapter 7 : Troubleshooting CSS Issues - mojoPortal

The following links provide solutions to common problems you can solve with CSS.

Norik Davtian 7 Comments You have designed the perfect layout for your new and upcoming website. But now converting all that beautiful Photoshop PSD layered file into a pixel perfect code is the challenging part. This is even more frustrating when fixing a bug in one browser now becomes a new bug in another browser. Often time this last case is the one that makes you pull your hair and say numerous WTFs before you completely understand what is going on in their code, because you probably know your entire style sheet by heart and this new code is no where close to any coding convention. However, moment of getting down and dirty with debugging that CSS code is much simpler and less frustrating than before, thanks to these very simple tools and techniques. If you are developing websites, then you need the right set of web development tools. And fortunately Inspect Element is now integrated natively with all modern web browsers as part of their developer tools. This tool enables you to select the specific element in the page that you would like to fix and immediately see what CSS rules are applied to that element that cause your selected element to display that way. Now you have full visual access to all the rules that are computed for that element. If you have not selected the right element, you could simply click on the source and pin point the exact element you would like to diagnose. As you see in the screenshot above, now you could tweak your CSS rules. Using the checkbox on the left side of each CSS property you could toggle that property, or even click on the property and change its value, or even hit enter key and add new rules for your element. Even most amazing is how easily it is possible to debug a hover state of an element by simply opening the drop down menu next to the selected element and choose: Moreover, it is more comprehensible to understand why our element has a 10px top and bottom padding instead of 5px, because the rule at line of style. Having this wealth of information now makes our debugging job much more easier. Suppose your style sheet email form field is not similar to other input elements and you wonder how you could solve this. It is even possible to change your debugging mode to the screen resolution of a device that you would like to test your layout and find out how your media queries CSS is behaving under those conditions. These developer tools have very powerful capabilities even beyond CSS development such as DOM inspection, JavaScript profiling, JavaScript Console, resource monitoring tools, and whole other set of features that need their own tutorial section beyond this article. As you could see in the picture above, in Chrome developer tools, it is even possible to emulate different browser user agent, or even emulate touch events such as swiping. At the end of the day it all depends on what you are trying to achieve and if that tool set enables you to get to your final goal. I personally use all these tools to my advantage. Each have something that I like and find helpful when I want to debug my code. For example I find the little square color representation of the hex colors next to the CSS color properties somehow useful, even though I can process the RGB color in my head and get approximate color palate visualization, these small differences make me alternate between these developer tools. Firebug Firebug was one of the first tools that provided inspector feature. It is one of the most popular web development tools, with almost 3 million average daily users. Back in the days when none of these developer tools were available for debugging old IE browsers, this second technique usually was very handy. I love to know boundaries of the element I am styling, so I know if there is any overflows, or floating issues, or etcâ€¦ For that reason I add a bright and noticeable background color to the active element I am styling and take the background color out once the styling is done. Sometimes, I go back and forth adjusting these elements, so I just move the debugging code to the end of my style sheet and comment it out, and uncomment it once I need it again. This way I can quickly toggle debugging mode and see how each element is positioned on the screen. Which results to â€¦ For this example I just added debug code to the main structure block items in my page. Now I can tell where each post starts and how much gap there is between each post or widget item in my sidebar, and if I want to add a gap between the sidebar and the main content, I can visually see it in bright colors. The opposite of this is also possible, by defining a series of debug rules, and adding that class to whatever item we want to see. For example, above I have defined a new class called debug with the following code. This is a piece of jQuery

code that comes to our aid by applying the debug class to foo element, or even adding CSS directly to the bar element in this case. For example, more specific selectors will override more general ones, or ID selector overrides class selector, or! Getting to know the Cascade will help you resolve your Cascading Style Sheets code faster. Here is an example straight from W3C spec on how selector specificity works. Final notes Achieving pixel perfect results is hard due to different renderings by different browsers. Share Articles Post navigation.

Chapter 8 : Inspect Element: Troubleshooting CSS in the Browser

Dreamweaver CS3 provides a great collection of tools for creating page designs with CSS, but when it comes to altering and troubleshooting CSS, things can get confusing in a hurry. To help you sort through the options, this column focuses on some of the challenges you may face when working with CSS.

Chapter 9 : Help:Cascading Style Sheets - Wikipedia

Most display problems in HTML and CSS come from cross-browser issues. The site looks fine in one browser, but there's problems in another. This is almost always a problem on any web project, but you can minimize the number of cross-browser issues by preventing many of them from happening in the first place.