

Cross manual

COMNAVSURFPAC/COMNAVSURFLANTINST 4790.9, IMPLEMENTATION AND UTILIZATION OF THE COMBAT SYSTEM OPERATIONAL SEQUENCING SYSTEM (CSOSS), COMNAVSURFPAC/COMNAVSURFLANTINST 4790.9, COMNAVSURFPAC/COMNAVSURFLANTINST 4790.9, PAGE 2

Reference (b) assigned Commander, Naval Sea Systems Command, as the agent for CSOSS development, implementation, and life cycle maintenance. Naval Surface Warfare Center Port Hueneme (NSWC PHD) Division Combat System Technical Documentation Branch (Code A34) is the technical authority for CSOSS development for new construction and in-service ships. Ships with CSOSS formally implemented will follow its guidelines for all conditions of readiness. Use of CSOSS as a readiness-management tool will complement Planned Maintenance System (PMS), Combat System Technical Operational Manuals (CSTOM), and other technical documentation. All CSOSS operations will be conducted by personnel qualified in accordance with reference (c).

File Name: Ccross manual.pdf

Size: 1813 KB

Type: PDF, ePub, eBook

Category: Book

Uploaded: 4 May 2019, 19:46 PM

Rating: 4.6/5 from 638 votes.

Download Now!

Please check the box below to proceed.



I'm not a robot



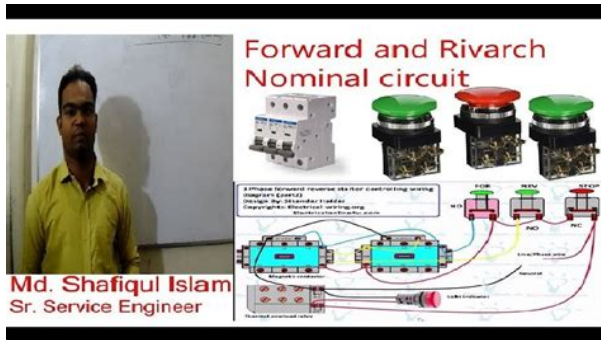
reCAPTCHA
Privacy - Terms

Book Descriptions:

Csoss manual

With our editor, you can edit the CSS, and click on aExamples might be simplified to improve reading and basic understanding. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using this site, you agree to have read and accepted our terms of use, All Rights Reserved. Powered by W3.CSS. This tutorial covers both the versions CSS1, CSS2 and CSS3, and gives a complete understanding of CSS, starting from its basics to advanced concepts. I will list down some of the key advantages of learning CSS Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects. Most commonly, CSS is combined with the markup languages HTML or XHTML. Im going to list few of them here You can define a style for each HTML element and apply it to as many Web pages as you want. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers. This complete guide explains everything about flexbox, focusing on all the different possible properties for the parent element the flex container and the child elements the flex items. It also includes history, demos, patterns, and a browser support chart. While those work well for pages, they lack flexibility no pun intended to support large or complex applications especially when it comes to orientation changing, resizing, stretching, shrinking, etc..<http://cstransit.ru/userfiles/hotpoint-tvm562g-manual.xml>

- **csoss manual, csoss user s manual, 1.0, csoss manual, csoss user s manual.**



Some of them are meant to be set on the container parent element, known as “flex container” whereas the others are meant to be set on the children said “flex items”. Please have a look at this figure from the specification, explaining the main idea behind the flex layout. Beware, it is not necessarily horizontal; it depends on the flexdirection property see below. Its direction depends on the main axis direction. Pin a copy up on the office wall. It enables a flex context for all its direct children. Flexbox is aside from optional wrapping a singledirection layout concept. Think of flex items as primarily laying out either in horizontal rows or vertical columns. You can change that and allow the items to wrap as needed with this property. The default value is row nowrap. It helps distribute extra free space leftover when either all the flex items on a line are inflexible, or are flexible but have reached their maximum size. It also exerts some control over the alignment of items when they overflow the line. Note that visually the spaces aren’t equal, since all the items have equal space on both sides. The first item will have one unit of space against the container edge, but two units of space between the next item because that next item has its own spacing that applies. MDN has detailed charts. The safest values are flexstart, flexend, and center. Using safe ensures that however you do this type of positioning, you can’t push an element such that it renders offscreen e.g. off the top in such a way the content can’t be scrolled too called “data loss”. Think of it as the justifycontent version for the crossaxis perpendicular to the mainaxis. The difference between these is subtle, and is about respecting the flexdirection rules or the writingmode rules. The difference again is subtle and is about respecting flexdirection rules vs. The more supported flexstart honors the flexdirection while start honors the writingmode direction. <http://cokhihoangvinh.com/uploads/userfiles/hotpoint-ue89x-manual.xml>

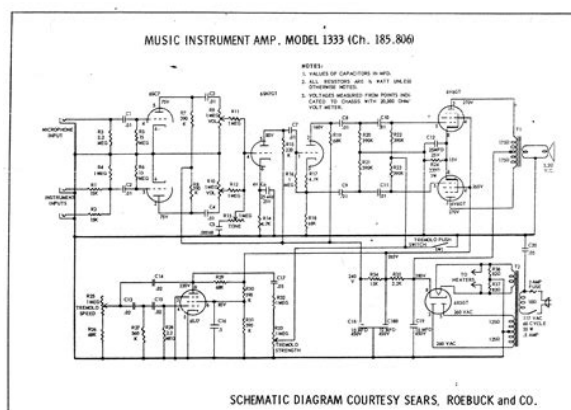
COMNAVSURFPAC/COMNAVSURFLANTINST 4790.9, IMPLEMENTATION AND UTILIZATION OF THE COMBAT SYSTEM OPERATIONAL SEQUENCING SYSTEM (CSOSS), COMNAVSURFPAC/COMNAVSURFLANTINST 4790.9, COMNAVSURFPAC/COMNAVSURFLANTINST 4790.9, PAGE 2

Reference (b) assigned Commander, Naval Sea Systems Command, as the agent for CSOSS development, implementation, and life cycle maintenance. Naval Surface Warfare Center Port Hueneme (NSWC PHD) Division Combat System Technical Documentation Branch (Code A34) is the technical authority for CSOSS development for new construction and in-service ships. Ships with CSOSS formally implemented will follow its guidelines for all conditions of readiness. Use of CSOSS as a readiness-management tool will complement Planned Maintenance System (PMS), Combat System Technical Operational Manuals (CSTOM), and other technical documentation. All CSOSS operations will be conducted by personnel qualified in accordance with reference (c).

The more support flexend honors the flexdirection while end honors the writingmode direction. However, the order property controls the order in which they appear in the flex container. It accepts a unitless value that serves as a proportion. It dictates what amount of the available space inside the flex container the item should take up. If one of the children has a value of 2, the remaining space would take up twice as much space as the others or it will try to, at least. It can be a length e.g. 20%, 5rem, etc. or a keyword. The auto keyword means “look at my width or height property” which was temporarily done by the mainsize keyword until deprecated. If set to auto, the extra space is distributed based on its flexgrow value. See this graphic. The second and third parameters flexshrink and flexbasis are optional. The default is 0 1 auto, but if you set it with a

single number value, it's like 1 0. The shorthand sets the other values intelligently. It couldn't be any simpler if you use flexbox. So setting a vertical margin of auto will make the item perfectly centered in both axes. Consider a list of 6 items, all with fixed dimensions, but can be autosized. We want them to be evenly distributed on the horizontal axis so that when we resize the browser, everything scales nicely, and without media queries. Below is a pen featuring this example. Be sure to go to CodePen and try resizing your windows to see what happens. Imagine we have a rightaligned navigation element on the very top of our website, but we want it to be centered on mediumsized screens and singlecolumned on small devices. Easy enough. What about a mobilefirst 3columns layout with fullwidth header and footer. And independent from source order. It doesn't just include prepending properties with the vendor prefix, but there are actually entirely different property and value names. This is because the Flexbox spec has changed over time, creating an "old", "tweener", and "new" versions.

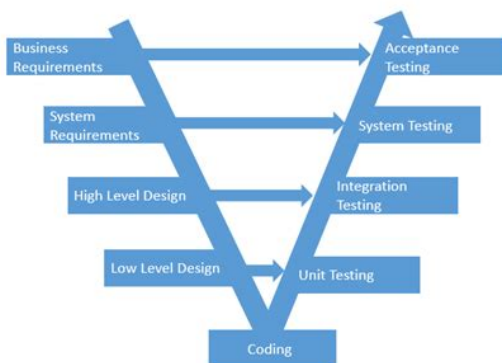
The best collection of them I've seen is Philip Walton and Greg Whitworth's Flexbugs. It's an opensource place to track all of them, so I think it's best to just link to that. They have courses on all the most important frontend technologies, from React to CSS, from Vue to D3, and beyond with Node.js and Full Stack. What's not to like Great info, as always! Otherwise could you build this layout using flexbox thanks How about managing 3rds, 5ths, 6ths, 12fths, etc., and when columns change to use different widths across viewports Perhaps not ideal, but they solve a complex problem elegantly. Then using media queries to make appropriate adjustments based on the users screen. Any solution If you put some text in Aside1 the 3 column Layout is gone. The truth is, is optional as long as is present and obviously when the value is none . It's only required when is present. Everything still looks great in Chrome. The prefixes still should be available if needed, but it shouldn't be necessary. They arguably much more importantly separate out implementation differences. What would happen if we just had one unprefixed word for a feature, and the syntax of its attributes was consistent across browsers, but the rendering behavior was different. Then you'd have to do ugly browser sniffing and serve different files to the client conditionally, like we did back in the dark ages of IE6. Otherwise, the most popular browser's implementation of the feature becomes the de facto standard even if it's the most broken again, IE6 Let's say there's only room for 4 of the items on the first row, the remaining 2 will be evenly spaced on the second row. ughh This is indeed a thing that could be added.It doesn't allow the paragraphs to break. It's like it's treating the displayflex as displayinlineflex. The only way I've been able to get around this is to change the aboutconfig of Firefox to multiline, but visitors won't have that set by default.



Currently I'm using flexbox for webkit and equalize.js for other browsers. That's why I'm running the v22 beta at the moment. You can always use the displaybox until ff22 is released. I have a really

awkward situation and I can't use display table. It messes up with the fluidity of the images. I played around with a few values and found that explicitly adding some height to the ul.navigation made the li's stack vertically. Is there a better way around this without requiring a hardcoded height It only contains one for webkit. Once I added that in, it does it nicely in my FF. I found it highly insightful. I tried recreating it on CodePen and noticed it wasn't working, even when I copied and pasted. Then I tried recreating it locally, copied and pasted, and again it didn't work. So then I took to the Chrome DevTools to take a look at what was going on and it looks like even though the pen uses the rule justifycontent spacearound;, what is actually rendered on the page is webkitjustifycontent spacearound. Turns out prefixfree was turned on in the CodePen config for the Scss panel. Too bad we don't use SASS, we rely almost solely on LESS. We would love to use Flexbox for clients, but it doesn't seem to play nicely cross browser. I checked this page in FF22 and IE10 and it was a mess. When you define mainaxis you say that its direction depends on the justifycontent property, but isn't the flexdirection property that defines if flex items are layed out as a row or as a column. Am I misunderstanding something I just found this Sass helper, which is working really well so far. The steps which you have mentioned are really perfect. Both tomato blocks and very last demoes do not work. Is there some workaround already As a workaround, you can use nested flexboxes in combination with media queries, as in my comment above it's not so flexible as true multiline flexboxes, but still better than nothing or use graceful degradation to old techniques like inlineblocks.

<https://jackson-pr.com/images/compressor-manuals-download.pdf>



Anyone else observed this, or have an idea as to why You can check out the codepen at The site scales ok but the Vimeo iframe videos do not. I was trying to use the FitVids.js script to make this work but I am not sure how to make that work with my Weebly template. YES I am not a website professional, I know nothing about CSS or HTML But I have been tasked with this job and I need to make it work properly. Any help would be appreciated. It got me started with my project. This time it's with IE11. If you look at your example of the menu, you will see that on the smallest width the menus are not shown in columns and stays as rows. So everything collapses for no reason. Of course it's fine in Chrome and Firefox 25 It doesn't inhibit understanding the content, but it would be nice if you fix it. Also it doesn't work for me in IE10. Autoprefixer does a great job of writing in the latest syntax and handling the fallbacks. The company I contract for right now uses IE8 so I have to wait until they move to newer version of IE. I have always wondered why a good layout system has been missing from CSS. Better late than never I guess. I look forward to using this on touch devices with webkit. I'm interested but a bit confused at the same time. If I code it literally copy it from what you have here to CodePen it runs as yours did. If, however, I try that on JSFiddle where I normally mess around the colors come out in a straight line only. I've even tried injecting the CSS into the Header before building the page via jQuery with much the same result. Seems that this only works without jQuery and for whatever reason only on CodePen. Any help would be appreciated, thanks! On

CodePen we use jQuery UI draggable, but there are others out there. Flexbox does make the situation easier though. One time I redid the whole CodePen editor layout in Flexbox for fun and it was way easier, but of course I can't find it now.

<http://hillstromplasticsurgery.com/images/compressor-mycom-manual.pdf>

Basically if the flex items have flex 1; they will fill the area, so you just resize one of them to a specific height or width and the other will fill the remaining space. So no math. I've been messing with it for a couple of days now and can't seem to figure it out. Using this I'm a frontend developer and still couldn't understand a single term that was used to explain what I was looking at. What I want is 3 breakpoints like this 1 3 rows containers vertical, small screen 2 2 columns, 2 rows medium screen 3 3 columns large screen 1 en 3 are easy, I just change the flexdirection from column to row. But how about 2 So basically it must look like The reason I could not get it to work is because IE11 does not like a maxwidth to be set on any flexitem. If you do, it wrongly calculates the space around or between the items. I was beating my head against it for a good hour until I discovered that IE11 doesn't like maxwidth on flex items. I have implemented a basic Holy Grail template. Flexbox is a thing of beauty! But then the layout becomes "infinite" you can make the screen wider and wider and the boxes and spaces will happily distribute themselves across that space possibly breaking any design restrictions. So in order to prevent that we could set maxwidth on the flex container, but that cancels out the centering for some reason and the page flushes left. In short flexbox will only work practically when using the full screen width and not limiting any flexible item with a maxwidth. As soon as you want to set a limit to any item, it falls apart. Also, very important. Make sure the child elements of the parent flex container don't have display inline; applied to them. It breaks it for some reason. I hope this helps someone! I feel like it's best at this point at least in terms of this guide to focus on the current properties. Also best in practice to let a tool like Autoprefixer deal with inserting those older properties for you when needed.

I'd love to see the pens using the flex wrap updated with "flexflow row wrap;" added unprefixed so they work in Firefox 29. But still a very good and informative article. The only way I've currently found forces me to add a padding to the container which isn't ideal. Give the container the same, but negative margin to still use the full width. And is that something one would even want to do Really like the rehaul, makes it even more useful. Cheers to you, Chris. The guide was crazy informative before but now it's also a great cheat sheet when needed. Thanks! Has always been very useful. How

does the flexbox fall on browsers that don't support the CSS3 You deserve at least a sixpack of Rolling Rock for this one, Chris — if that's still your brew of choice that is Is there an easy way to center everything in a container box when arranging elements as columns. Hope this makes sense. If you flow the elements by column vertically, the justify-content center will really display the elements in the center bit of the flex box vertically, i.e, some space at the top, then your elements, then some space at the bottom. What you wanted is for each element to center align horizontally, which you can probably achieve by using text-align property. Ah yes, I guess because I didn't set a height on the flexbox, I didn't see how the elements were centering vertically. Crisp, crucial and highly valuable. I enjoyed it. I see that they have what seems to be the old version of flex box in the documentation. But then on codepen.io, when you include compass you are able to use the other directives. I'm unable to get this working locally however. Ideas Should read Where is flexbox standing now for support I'm a little hesitant because of browser version support. I've set a container width to 100% and put six div items with width of 20% in it. I was expecting to see five divs evenly space and the sixth div directly underneath the others, one line down I'm using rowwrap.

<http://conservationenergy.com/wp-content/plugins/formcraft/file-upload/server/content/files/16288106b240d7---cambridge-audio-cd6-service-manual.pdf>

This kinda works, but there is a big gap between the five divs across the top of the page and the sixth div below them. I need to know how to get rid of the gap. Here is the Codepen I'm not too sure if it will help for your purpose, but with your demo it works. Also, I would rather set flex 1 1 20%; on each sub item instead of specifying the width again, it depends on what you want to do. In all honesty, I'm not really sure how the code is being interpreted. I understand that giving everything a flex size of 1 gives everything an equal amount of space, but is the 20% overriding everything the first 1 I've played around with the second 1 in the code you provided, but it doesn't seem to do anything. Oh, and the purple box now fills the entire width of the screen, which looks good, but is it the first 1 doing that since it is clearly taking up more than 20% of the container now. Anyhow, don't mean to be lazy; I can look this stuff up tomorrow. Time for bed in the UK though. It doesn't look good in safari, even doesn't look anyhow. 8 how to make it work in safari Not supporting it on my sites. You could just revert to floats for it, but when I discussed it with my employer he said "no one uses it anyways". Inside this container I have 3 divs. I want last one footer to be always at the bottom of this page. Is this possible to do. I know it is of course; but I want to use only flexbox model. Thanks anyway! This is best place to learn CSS Tricks. This is the best resource I've found so far. If any possible on that particular IE11 alignment modification stylesheet. Only downside is all the prefixes that you need. Wanted to spread the word, since there seems to be some confusion around this property flying around in the wake of Firefox previously not having supported it. It's a reference. Could anyone help me with this. I was written some code reading article. Want to have this I want to get it on the right side of the block number two, but below of the block number three.

<BACSIHA.COM/public/ckfinder/userfiles/files/casio-calculator-df-320tm-instruction-manual.pdf>

i must remove because message was rendering in wrong way I will implement it in a new project.D Sad to think we're still another few years out from implementing this without fallback support. Using "space-between" on the container sticks the footer to the bottom of the browser window and sticks the content area to the top of the browser window. I have tried and it is failing to keep aspect ration and the usual padding trick doesn't seem to work. Alternatively is there an easy solution you could give me here. Thank you for your work. If it is valid is there are work around to still using flexbox for page layout without the performace hit Both Flexbox and Grid layout have their pro's and cons. The flexbox is more suitable for dynamic content think about displaying a random amount of images of a random size, where the grid layout is preferable for known content areas. Both can adjust for the screensize, but are optimized for different applications. I've noticed that any example, where flexbox is used for the entire layout, leaves out content inside these boxes. Doesn't seem like flexbox is

useful for layouts without a lot of hacking. We're actively looking for feedback on that issue at the moment, so please let us know if any! Just a couple things I noticed from a skim. The shorthand resets things in appropriate ways, and will therefore result in fewer cascading errors. Please use the shorthand! Nesting a few flex'd containers causes Firefox to become unresponsive. I've tried many things and I've check many websites. I would appreciate any help. Thanks in advance. Kudos for taking the time to make this super intuitive. Unfortunately it still seems to be in it's revolutionary infancy and I don't think my employer would be happy if I tried to implement this on our sites. The video lesson link and a five lesson tut Only specifying widths every time is not very effective. No one should have to add a width 1px; to every element within if they want it to behave properly.

I just had to redo an entire page which used to use an HTML table to present a matrix. After requirements changed, I realized I could no longer use a table since each "column" needed to have an arbitrary number of "rows". In other words, I had to go from rowmajor format to columnmajor format. So I used Flexbox to lay out the columns in lefttoright row direction, and then lay out each child in each row in toptobottom column direction. But then I needed to reorder each row in reverse order, which Flexbox also made easy use either the "order" property or set the direction to "columnreverse". Done. Voila. The JS that I wrote to make it happen is now half the size, and the CSS is turning out to be smaller, too. Woohoo!!!! Let me know when you can shiv it back to ie9. IE doesn't seem to like webkitflexflow. Adding flexwrapwrap; flexdirection row; or just flexflow row wrap; works though. I write css for the screens 1440 resolution. Got a container and 3 columns in it. Used this tutorial and it worked great in FF and Chrome, but in Opera it does not. Col 1 and 2 are fully apart and the 3rd column is under the 1st. Just to mention I am new here i mean webdesign. That is the whole point of Flexboxes. So you don't have to use floats. It's not ridiculous to see something like Especially when you get like 8 levels deep. I find a difference between resizing my laptop monitor and actually viewing it on other devices. It really fasttracked my understanding of using the flexbox model. Not so much the concept of what they were, but how the actual values played out. This was not the case. It was always greater than 2 times. The grow and shrink values have nothing to do with each other. But the piece that was eluding me, and causing the actual width values to not follow this ratio, is that the ratio is based on the amount that the containers have grown past the basis width or under the base width for flexshrink.

That's the whole point of flexbox, right To align items along main axis you'll need to change alignitems instead. That's how I think about it anyway. Flexdirection Imagine we have a rightaligned navigation on the very top of our website, but we want it to be centered on mediumsized screens and singlecolumned on small devices. Easy enough." The navigation don't works in Chome 41.0.2272.101 m Just fixed it by adding TWO flexbox items into CSS. I do not understand. They work offtheshelf. Webpages development are becoming breezy now, given most of the common burden is taken by the framework. Love it, thanks! Also, you the container article is missing a height, which ends up in confusing the result of applying alignitems and justifycontent as the same in that special case. Using this page as a guide and reference, I created a webapp based log in template that looks like a phoneapp. It's mostly just an exercise in column layout for flex; it helped me gain a much greater understanding of flex properties and I thought someone else might care to poke at it to help learn. Is there a particular attribution you would like Do you have any suggestions for a graceful fallback or is it better to just style it "traditionally" for.noflexbox using Modernizr Is it possible to have, in a list comprised of multiple rows, the first row "spacearound," and the other rows after left align. In my list of items I'm not really a fan of if one or two items are wrapped to the next row, they "spacearound" and end up in the middle, it kind of makes you lose track if you are going down the list make sense. It's no biggie, just was wondering if there was a way to specify the last row or something. Great tutorial btw! Thanks in advance. Here's a very basic flexbox example; see if it helps. Feel free to fork, repost and question. In order for Safari to wrap via flexbox webkitflexbasis must be auto which is Safaris default value.

Really like how you formatted it, the other articles on the flex box suck compared to yours. Thank you sir. The alignment you see is on the last but one line. See the solution on I often use flexbox with margins and calc, so I might use something like I guess it's a rounding error, and it won't affect all resolutions, but a combination of screen width and element width might sometimes mean you only get two columns on a line instead of three. To get around this, I use It's doesn't look quite as clean as in the other browsers, but it does solve the problem and it isn't too convoluted. I'm grateful you posted it. Thank you very much. People, now I need help with this Share please! I followed it whilst updating something I did for a friend's project before, but have come into difficulties. The three elements the twitter widget's container, the cbox's container and the ccentre's content I was trying to update to use flex like in the tutorial, but it's not worked. It looks like the ccentre might be the cause. Any ideas Here it is on Codepen I don't know the answer and I wonder if there is any solution to this. Can tell the reader of this in advance. I don't completely understand the Note about the best use for Flexbox vs. Grid. I learned a ton. One question though, the note that you included in the background section "Note Flexbox layout is most appropriate to the components of an application, and smallscale layouts, while the Grid layout is intended for larger scale layouts. " links to an article that is over a year old and has a note on it saying that it is influx. Are there any updates to that article coming down the pipeline. I would love to read the two in tandem and better be able to grasp in which situations each would be most appropriate. Less code and it works even with old browsers. See Using inlineblock keeps you dependent on the browser default use of extra space left and right of inline li elements.

This rendering can be fixed by floating the li elements, but flexbox is a nicer modern way of achieving that effect. In this case, that's vertical space. I don't think there's a way to do what I'm trying to do with flexbox. Most of it are in fact additions to CSS and HTML, rather than changes. The reason was that certain page layouts that you see nowadays, were very difficult to implement with the old specification. Before it was very hard to make a dynamically scaling website. Using just percentages to scale the sections just didn't cut it. One improvement was the introduction of the calc function that could use percentages and static units together, but even with that it was still hard to read code. Flexbox was a great addition that is very easy to use once you read this article. Nobody is stopping you, but you deny yourself some awesome tools if you do There's always something more to learn. Without new features and new capabilities, we atrophy and fail to realize our full potential. It's an organization, and a democracy, guided by the people and companies that invented the web and continue to use to everyone's benefit. Everything's optional. After all, she already has TV, YouTube, and all the toys she needs at home Just fyi, no reply needed. Seeing the parent and child examples side by side meant it was easier to compare behaviours and to pick the right approach. Can somebody can give me an exemple about how to do. Better yet, post your question and all related code to a site like Stack Overflow that's designed for questions and answers So, is the above table wrong I use it almost once a day !! Many thanks to you Chris ! Suppose I have made two div of equal height using flex and now I want to make the all the elements inside the div to align to each each other. Is that possible I cannot wait to test it out more and see how it all works in different scenarios. My problem with flexbox is, that I can not get a second child item to align vertically.

You can see this in action here No matter what I try, I will either lose vertical centering of the heading or the second child won't align. What am I doing wrong here Thanks! She said "shoelace" or something is better can u confirm The truth is, is optional as long as is present and obviously when the value is none. It's only required when is present. It has helped me several times. I am having one issue that I cannot figure out. I can't get a single line of text to vertically center within an element. It seems so simple, and yet I've wasted hours without any luck. Would anyone be willing to comment on this Codepen. Resize the width below 900px and you'll see what happens. Thanks, Chris The height 100% is needed to stretch your This is just brilliant. Thanks so much. This is my goto site whenever using flexbox. I have encountered a bug on firefox that does not allow elements to be flex

containers. It took me AGES to find that out, so I wanna share this with other folks that might be going through the pain I have just experienced! If the answer is "depends on what browser support you need", I really wouldn't know or couldn't predict exactly who might visit my commercial site. With the statistics it gives you, you can see the browser breakdown of the people who come to your site. I think that would let you know how much of a need there really is for support for given browser versions. As I've been getting up to speed with css over the past year or so, I have referenced this page a thousand times. It has been just so helpful. I had to write and give you a welllearned "thank you". Great work. Much appreciated. Thank you. If somebody can explain. Thank you, Igor The CodePen examples took a little adjusting to work for me on Firefox 48. I had to remove the webkit prefix from webkitflexflow on examples 1 and 2. I uploaded everything to " ". But why do the two col1 at the top not have the same width as the col2 They should add up and make 50% width. Very helpful.