

Background

Basics & Terminology



Properties for the Parent (flex container)

display

This defines a flex container; inline or block depending on the given value. It enables a flex context for all its direct children.

```
css
.container {
  display: flex; /* or inline-flex */
}
```

Note that CSS columns have no effect on a flex container.

flex-direction



This establishes the main axis, thus defining the direction flex items are placed in the flex container. Flexbox is (aside from optional wrapping) a single-direction layout concept. Think of flex items as primarily laying out either in horizontal rows or vertical columns.

```
css
.container {
  flex-direction: row | row-reverse | column | column-reverse;
}
```

- `row` (default): left to right in LTR; right to left in RTL
- `row-reverse`: right to left in LTR; left to right in RTL
- `column`: same as `row` but top to bottom
- `column-reverse`: same as `row-reverse` but bottom to top

flex-wrap



By default, flex items will all try to fit onto one line. You can change that and allow the items to wrap as needed with this property. Direction also plays a role here, determining the direction new lines are stacked in.

```
css
.container {
  flex-wrap: nowrap | wrap | wrap-reverse;
}
```

- `nowrap` (default): single line / left to right in LTR; right to left in RTL
- `wrap`: multi line / left to right in LTR; right to left in RTL
- `wrap-reverse`: multi line / right to left in LTR; left to right in RTL

flex-flow (Applies to: parent flex container element)

This is a shorthand `flex-direction` and `flex-wrap` properties, which together define the flex container's main and cross axes. Default is `row nowrap`.

```
css
flex-flow: <flex-direction> | <flex-wrap>
```

justify-content



This defines the alignment along the main axis. It helps distribute extra free space left over 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.

```
css
.container {
  justify-content: flex-start | flex-end | center | space-between | space-around;
}
```

- `flex-start` (default): items are packed toward the start line
- `flex-end`: items are packed toward to end line

- **center** : Items are centered along the line
- **space-between** : Items are evenly distributed in the line; first item is on the start line, last item on the end line
- **space-around** : Items are evenly distributed in the line with equal space around them. 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.

align-items



This defines the default behaviour for how flex items are laid out along the cross axis on the current line. Think of it as the **JUSTIFY:NBHFFF** version for the cross axis (perpendicular to the main axis).

```
css
.container {
  align-items: flex-start | flex-end | center | baseline | stretch;
}
```

- **flex-start** : cross start margin edge of the items is placed on the cross start line
- **flex-end** : cross end margin edge of the items is placed on the cross end line
- **center** : Items are centered in the cross axis
- **baseline** : Items are aligned such as their baselines align
- **stretch** (default) : stretch to fill the container (still respect min-width/max-width)

align-content



This aligns a flex container's lines within when there is extra space in the cross axis, similar to how **justify-content** aligns individual items within the main axis.

Note: this property has no effect when there is only one line of flex items.

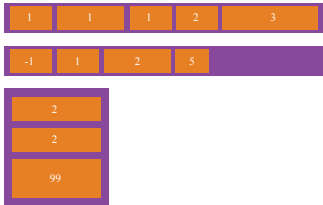
```
css
.container {
  align-content: flex-start | flex-end | center | space-between | space-around | stretch;
}
```

- **flex-start** : lines packed to the start of the container
- **flex-end** : lines packed to the end of the container
- **center** : lines packed to the center of the container
- **space-between** : lines evenly distributed, the first line is at the start of the container while the last one is at the end
- **space-around** : lines evenly distributed with equal space around each line
- **stretch** (default) : lines stretch to take up the remaining space



Properties for the Children (flex items)

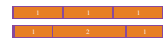
order



By default, flex items are laid out in the source order. However, the **order** property controls the order in which they appear in the flex container.

```
css
.item {
  order: <integer>;
}
```

flex-grow



This defines the ability for a flex item to grow if necessary. 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 all items have **flex-grow** set to 1, the remaining space in the container will be distributed equally to all children. If one of the children a value of 2, the remaining space would take up twice as much space as the others (or it will try to, at least).

```
css
.item {
  flex-grow: <number>; /* default: 0 */
}
```

Negative numbers are invalid.

flex-shrink

This defines the ability for a flex item to shrink if necessary.

```
css
.item {
  flex-shrink: number; /* default 1 */
}
```

Negative numbers are invalid.

flex-basis

This defines the default size of an element before the remaining space is distributed. It can be a length (e.g. 20%, 5em, etc.) or a keyword. The `auto` keyword means "look at my width or height property" (which was temporarily done by the `flex-size` keyword until deprecated). The `content` keyword means "size it based on the item's content" - this keyword isn't well supported yet, so it's hard to test and harder to know what its brethren `max-content`, `min-content`, and `fit-content` do.

```
css
.item {
  flex-basis: <length> | auto; /* default auto */
}
```

If set to `0`, the extra space around content isn't factored in. If set to `auto`, the extra space is distributed based on its `flex-grow` value. [See this graphic.](#)

flex

This is the shorthand for `flex-grow`, `flex-shrink` and `flex-basis` combined. The second and third parameters (`flex-shrink` and `flex-basis`) are optional. Default is `0 1 auto`.

```
css
.item {
  flex: none | [ <flex-grow> <flex-shrink>? ]? <flex-basis>;
}
```

It is recommended that you use this shorthand property rather than set the individual properties. The short hand sets the other values intelligently.

align-self



This allows the default alignment (or the one specified by `align-items`) to be overridden for individual flex items.

Please see the `align-items` explanation to understand the available values.

```
css
.item {
  align-self: auto | flex-start | flex-end | center | baseline | stretch;
}
```

Note that `flex`, `clear` and `vertical-align` have no effect on a flex item.

Examples

Let's start with a very very simple example, solving an almost daily problem: perfect centering. It couldn't be any simpler if you use flexbox.

```
css
.parent {
  display: flex;
  height: 100px; /* Or whatever */
}

.child {
  width: 100px; /* Or whatever */
  height: 100px; /* Or whatever */
  margin: auto; /* Magic! */
}
```

This relies on the fact a margin set to `auto` in a flex container absorbs extra space. So setting a vertical margin of `auto` will make the item perfectly centered in both axes.

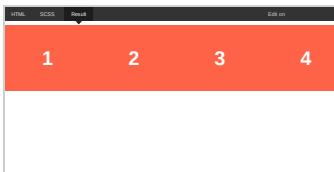
Now let's use some more properties. Consider a list of 6 items, all with a fixed dimensions in a matter of aesthetics but they could be auto-sized. We want them to be evenly and nicely distributed on the horizontal axis so that when we resize the browser, everything is fine (without media queries).

```
css
.flex-container {
  /* We first create a flex layout context */
  display: flex;

  /* Then we define the flow direction and (if we allow the items to wrap
   * Remember this is the same as:
   * flex-direction: row;
   * flex-wrap: wrap;
   */
  flex-flow: row wrap;

  /* Then we define how is distributed the remaining space */
  justify-content: space-around;
}
```

Done. Everything else is just some styling concern. Below is a pen featuring this example. Be sure to go to CodePen and try resizing your windows to see what happens.

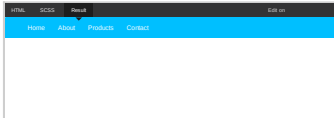


Let's try something else. Imagine we have a right-aligned navigation on the very top of our website, but we want it to be centered on medium-sized screens and single-columned on small devices. Easy enough.

```
css
/* Large */
.navigation {
  display: flex;
  flex-flow: row wrap;
  /* This aligns items to the end (line on main-axis) */
  justify-content: flex-end;
}

/* Medium screens */
@media all and (max-width: 800px) {
  .navigation {
    /* When on medium sized screens, we center it by evenly distributing empty space around items */
    justify-content: space-around;
  }
}

/* Small screens */
@media all and (max-width: 500px) {
  .navigation {
    /* On small screens, we are no longer using row direction but column */
    flex-direction: column;
  }
}
```



Let's try something even better by playing with flex items flexibility! What about a mobile-first 3 columns layout with full-width header and footer. And independent from source order.

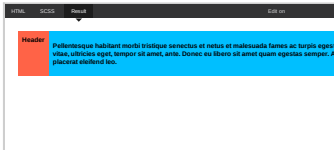
```
css
.wrapper {
  display: flex;
  flex-flow: row wrap;
}

/* We tell all items to be 100% width */
.header, .main, .nav, .aside, .footer {
  flex: 1 100%;
}

/* We rely on source order for mobile-first approach
 * in this case:
 * 1. header
 * 2. nav
 * 3. main
 * 4. aside
 * 5. footer
 */

/* Medium screens */
@media all and (min-width: 600px) {
  /* We tell both sidebars to share a row */
  .aside { flex: 1 auto; }
}

/* Large screens */
@media all and (min-width: 800px) {
  /* We invert order of first sidebar and main
   * And tell the main element to take twice as much width as the other two sidebars
   */
  .main { flex: 2 0px; }
  .aside-1 { order: 1; }
  .main { order: 2; }
  .aside-2 { order: 3; }
  .footer { order: 4; }
}
```



Prefixing Flexbox

Related Properties

Other Resources

Bugs

Browser Support

Broken up by "version" of flexbox:

- (new) means the recent syntax from the specification (e.g. `display: flex;`)
- (newer) means an odd unofficial syntax from 2011 (e.g. `display: flexbox;`)
- (old) means the old syntax from 2009 (e.g. `display: box;`)

Chrome	Safari	Firefox	Opera	IE	Android	iOS
20 (old) 21 (new)	3.1 (old) 6.1 (new)	2.20 (old) 22+ (new)	12.0 (new)	10 (newer) 11+ (new)	2.1 (old) 4.0+ (new)	3.2 (old) 7.1+ (new)

Blackberry browser 10+ supports the new syntax.

For more informations about how to mix syntaxes in order to get the best browser support, please refer to [this article \(CSS-Tricks\)](#) or [this article \(DevOpera\)](#).



PSD2HTML.com
Looking for Front-End Teams? We've got you covered. Start at Trial Price!



100% responsive images
File system as a service for web sites and mobile applications. Code files.

Comments



Bill Webb
#css #html #javascript
Yay. Less javascript and more CSS. What's not to like? Great info, as always!

Reply



Alex
#css #html #javascript
Pardon its size, but it's still not valid for a simple perfect "product grid" with no margins at first and last elements in row, and left aligned. Otherwise: could you build this layout using Flexbox? <http://i.imgur.com/9H1dZjg> Thanks



Lawrence Botha
#css #html #javascript
@Alex: Yes, you can. In the same manner that you do so with non-flex grids, apply a negative margin-left to the grid wrapper, and apply that same value as padding-left to all grid columns.

```
.grid { margin-left: -20px;
.grid_col { padding-left: 20px; }
```



mystdat
@Alex @Lawrence That has little to do with flexbox itself. ...el.not(:last-of-type) and similar exclusion selectors, negative margins are rubbish.



Lawrence Botha
@mystdat You're correct, it has nothing to do with flexbox. Using not selectors, however, will be unscalable, and you will lose IE8 support (bes of an issue now).
If I have a grid with 4 items, each occupying 25% of the width, that technique fails, since the 4th item will not sit flush with the container edges.
If I have a grid with 4 items, 25% width on desktop, and then 50% width on mobile, that technique fails again, for the above reason. How about managing 3rd, 5th, 6th, 12th, etc., and when columns change to use different widths across viewports?
I wouldn't call negative margins rubbish. Perhaps not ideal, but they solve a complex problem elegantly. <http://tympanus.net/codrops/2013/02/04/creating-modular-dynamic-grids/>



Yazlin
@Alex... actually, it's alot simpler. Just use

`justify-content: space-between;`

More [here](#)



Matt
The CSS Working Group has a document online of "... Mistakes in the Design of CSS", one of them is this:
"Flexbox should have been less crazy about flex-basis vs width/height. Perhaps, if width/height is auto, use flex-basis; otherwise, stick with width/height as an inflexible size. (This also makes min/max width/height behavior fall out of the generic definition.)"
Can you talk about what they mean by this?



macgal
Abuj



dAVIT dJATKO

2 years vashoj co a tym?

User images via
gravatar.com

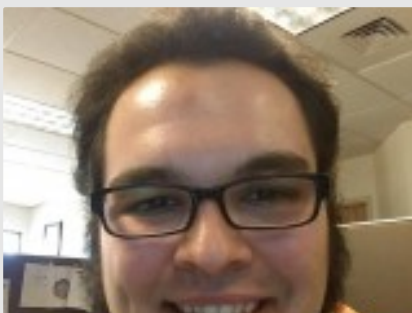
oNDREJ KONRÁT

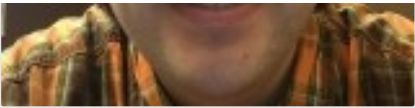
Any body browser



Josh McCullough

For your final example, how would you make the content (center row) take up all available space, so that at minimum, the footer is pinned to the bottom of the window - but if the content area has more content, the footer will push below, allowing scrolling. This can be accomplished by setting a min-height on the content row: $\text{calc}(100\% - \text{header-height} - \text{footer-height})$ but it requires hard-coding or JS to accomplish AFAIK.





Alan Carpenter
Front-End
@ Lawrence at the point of using flex does it not become a problem already? I think the grid solution could be solved with nth-child. Then using media queries to make appropriate adjustments based on the user's screen.



Andy Maleh
Front-End
Perfect Product Flexbox Layout (using justify-content: space-between and flex products):
See the Pen [Edged Row Wrap Flexbox](#) by Andy Maleh (@AndyMaleh) on CodePen.
Though to be honest, I don't like that I had to use fillers for the Flexbox implementation to ensure the last row is spaced evenly.
It would be mighty nice if they offer Flexbox row selectors for multi-row wrap flows.
Here is an alternative implementation with display: inline-block:
See the Pen [Edged Row Wrap Inline Block Layout](#) by Andy Maleh (@AndyMaleh) on CodePen.



Nafis
Front-End
Not working on iPad, iPhone, desktop safari also. Any solution?





Hubert Hubendubel
4
Your last example only works with no content. If you put some text in Aiskel the 3 column layout is gone.



User images via gravatar.com

PaulOB
4
@Hubert: Yes the 3 col layout needs this added.

```
div[id="x1"] and (min-width: 600px) {  
  .aside {  
    flex: 1 0 0;  
  }  
}
```

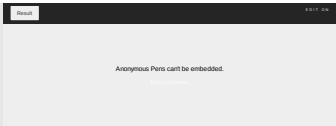
(mentioned in a while ago in an earlier post and assumed someone would update the demo.)



Kazi
4
@Josh McCullough: Its pretty simple to achieve that, better and easier then ever before. Just use the flex property and set it to 1, for e.g.

```
.someClass {  
  flex: 1;  
  color: #e6e6ba;  
  padding: 2rem;  
}
```

flex is a very powerful property and can be used in the shorthand of flex: 1 1 auto; (grow, shrink and basis) - using just flex: 1 tells it to take all the remaining space thus making the footer stick at the bottom. Look an eye out for grid to make a proper entry into the browsers and we would be having magic on our plates in terms of layouts.
See it live in action:



Alex2
Well, it's bad on many levels. Too verbose, hard to manage, it already creates "frameworks" around it, just to make it manageable. 25 years ago we already had tools, WYSIWIG IDE's and ways to define UI and "responsive" views.... For goodness sake, can we come back to roots and come up with simple and effective markup language with UI tools and *plain* resizing rules for view elements?

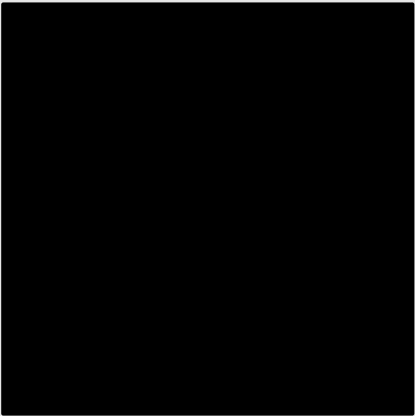


Chris
Regarding the 'flex' property
Saying that the 2nd and 3rd parameters: '<flex-shrink>' and '<flex-basis>' are optional is slightly misleading because it implies that '<flex-grow>' (the 1st parameter) is never optional. The truth is, '<flex-grow>' is optional as long as '<flex-basis>' is present (and obviously when the value is 'none'). It's only required when '<flex-shrink>' is present.





Chris
oops, sorry, my comment was for the whole board, not just to your post.

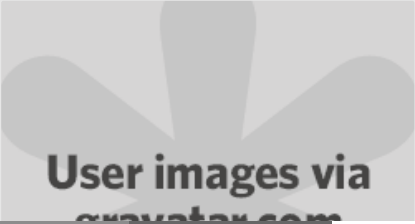


Kuhan
`<article class="main">
<aside class="side"></aside>
</article>`
when this article and aside come as html tag, I never know this



Jacob Duhal
Hey Chris,
Thank you so much for the comprehensive write up.
I just updated Firefox to v20 on a mac and now all of the flex-box demos aren't working. Everything still looks great in Chrome.
Anyone else having this problem?

Reply :





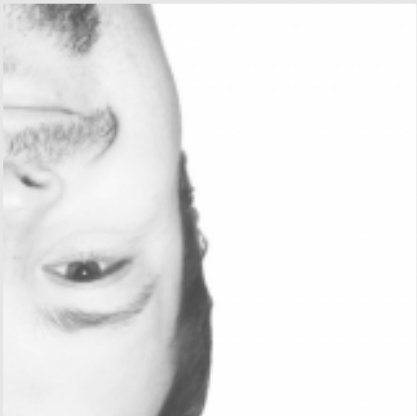
Andreas

Issues with Ch 34.0.1847 on OSX 10.9.2
Thanks for the webpage Chris!



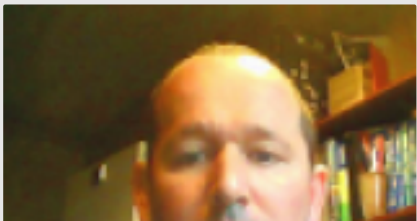
sl01k

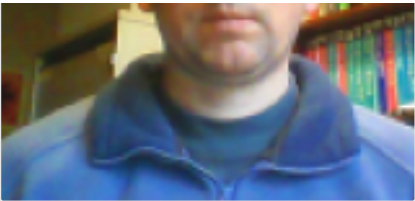
FF 2-21 (old) - (old means the old system from 2009 (e.g. display box)



Robert Fauver

The demos are using the new fontbox specs which requires FF 22+





Peter Lord
Not working for me on ubuntu 14.04 with firefox 29



Coolcat007
The only thing I don't understand is why the use of prefixes is needed if the syntax doesn't differ from the recommendation.
I think what would be enough is (using the above example):

```
.wrapper {
  display: -webkit-box;
  display: -ms-box;
  display: -ms-flexbox;
  display: flex;
}

.item {
  -webkit-box-flex: 1 200px;
  -ms-box-flex: 1 200px;
  flex: 1 200px;

  -webkit-box-ordinal-group: 2;
  -ms-box-ordinal-group: 2;
  -ms-flex-order: 2;
  order: 2;
}
```

At the moment this is not supported, but I think it should be because everything that was left out here had the recommended syntax. The prefixes still should be available if needed, but it shouldn't be necessary.



Tom L
Good explanation of the need for multiple vendor prefixed rules here:



Billy Wenge-Murphy

Vendor prefixes aren't just about syntax differences. 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 IE5. Once everyone has a correct implementation, then the prefixes can be dropped. Otherwise, the most popular browser's implementation of the feature becomes the de facto standard even if it's the most broken (again, IE5).



Daniel

2012

Regarding the example with the 8 items of fixed dimensions to be evenly distributed - using the justify-content: space-around; rule

I'd really like to use this, however it's not doing exactly what I want. 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. (right)

Is there any way for items on the last row to be placed/aligned underneath the elements from the previous row (left-right)?

Reply :



Coolcat007

This is something that can be done with the grid layout module, but it is not supported by the browser yet.
You could always use tables and calc()



Daniel

@Coolcat007 You mention that this can be done with tables and calc() - is this so - even if you have a dynamic number of items?? If it is - any chance of a fiddle / codepen?

Thanks!

User images via gravatar.com

Coolcat007

Sorry, I misunderstood your question. For a dynamic number of items, this won't work without JS or php.
This is indeed a thing that could be added.
Something like align-items:main-ais /cross-ais could be a great addition.





Catalin Rosu

I think the browser support table is missing a cell, the one with Opera's version 12.1+.
The table, as it is now, shows "Any" for it. It's funny! :)

Reply :



Chris Coyer

Fond, thanks!



Tim McKay

Works beautifully in Chrome and Safari, but I'm on Firefox (21) and it's not working well at all. It doesn't allow the paragraphs to break. It's like it's treating the displayflex as displayinline-flex. The only way I've been able to get around this is to change the aboutconfig of Firefox to multline, but visitors won't have that set by default. Has anyone had any luck with this? Currently I'm using flexbox for webkit and equalize.js for other browsers.

Reply :





Coolcat007
I think that's because flexbox isn't fully supported by firefox until v22. That's why i'm running the v22 beta at the moment. You can always use the display:box until FF22 is released.



Daniel
Am I crazy enough to use this in production? I have a really awkward situation and I can't use display: table. It messes up with the fluidity of the images.

Reply :



Kevin L.
You can use flexbox in production pretty well as long as you're using a sound way to detect less-than-ideal support for flex-wrap w/ modernizr and use a ratio-based grid system like Singular/tygs as a fallback.
An example: <http://www.mindor.com/ga/9783325> [buggle the flexbox and .noflex option.
It's a sound strategy to the extent you can use flexbox first towards planning for the layout and quickly create the fallback with a ratio-based grid system.
As long as you're considerate enough to have a style guide that documents documenting how a particular component ought to look if it in fact differs from both, you should be fine.





Wolf

4 years ago

Firefox is now unaffiliated in Firefox 22.

Reply



Tri Noentsie

4 years ago

I found a compass compatible [minors](#)

Reply



tinabeans

4 years ago

In your second Codapen example (both the blue navigation bar), I couldn't figure out why the flow-direction: column doesn't seem to kick in at the smallest screen width. 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 hard-coded height?

Reply

User images via gravatar.com

Jay

That's because the code for max 600 width is missing a few flex column or 4p; if you are using Firefox, it only contains one for web-ik. Once I added that in, it does it nicely in my FF.



skitterm

Thanks for the post, I found it highly insightful.

Reply :



Ankur Oberoi

Something weird is going on in the first example's pen (<http://codepen.io/httpGravdipen/1K0j>). I tried recreating it on CodePen and noticed it wasn't working, even when I 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 `justify-content: space-around;`, what is actually rendered on the page is `width: 100%; justify-content: space-around;`. Turns out prefix free was turned on in the CodePen config for the Sizer panel.

Even if this was CodePen's prefix-free doing the work for me, mixing vendor prefixed rules and non-prefixed rules that the preprocessor transforms should be a no-no.

Reply :





Ryan Boag

1 JULY 24, 2013

Nice post Chris, I like how thorough and detailed you are. Too bad we don't use SASS, we rely almost solely on LESS. We would love to use Flarebox 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. Do you, or anyone else, know of any good JS polyfills or plugins or solutions to get this to play cross-browser nicely? Otherwise, how long (in your opinion) until we can 'realistically' use this without a lot of cross-browser headaches?

Reply



Dario Grassi

1 AUGUST 13, 2013

Great post Chris. I think that flexbox capability to order items will be useful in RWD.

I've got only a question. When you define main-axis you say that its direction depends on the justify-content property, but isn't the flex-direction property that defines if flex items are layed out as a row or as a column? Am I misunderstanding something?

Reply



Chris Coyier

1 DECEMBER 22, 2015

When you define main-axis you say that its direction depends on the justify-content property, but isn't the flex-direction property that defines if flex items are layed out as a row or as a column?

You're correct, that was wrong in the article and is fixed now.



ZippZipp

4 months ago

Hey, anybody knows real site that using flexbox?

I know that SBT try to use it some time ago, but returns to React.

Reply :



Jacob Duball

4

Hey ZippZipp,

I tried to build my personal portfolio site with flexbox a few months ago, but got super frustrated with the syntax. I just found this SaaS helper https://raw.githubusercontent.com/compas-flexbox/master/extensions/compas-flexbox-stylesheets/_flexbox.css, which is working really well so far. I'm hoping to launch my new site in the next 2 weeks using flexbox for everything except IE &9.



Johnny Calderon

10/10/2017 10:00 AM

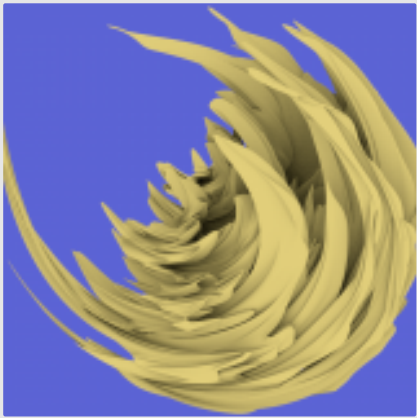
I would like to find one too, but older browsers just make it a big pain... I'd rather use floats to keep the headache away and less code.

Just yesterday I was checking my browsers support and I saw that flex is now un-prefixed in these versions, but unfortunately not everybody has updated browser versions.

IE11
Mozilla Firefox 27.0.1
Chrome 32.0.1700
Opera 23.0

Safari still uses the rule "display: -webkit-box"

Safari 5.1.7



Eden

1 did a school project using flexbox (with help from Autoprefixer): [colong.github.io/ASH20](#)



David

10/10/2017 10:00 AM

main axis - The main axis of a flex container is the primary axis along which flex items are laid out. Beware, it is not necessarily horizontal; it depends on the **justify-content** property (see below).

I think you mean **flex-direction**.

flex-direction (Applies to: parent flex container element)

flex-direction: row | row-reverse | column | column-reverse

row (default): left to right or right to left in rtl

row-reverse: right to left or left to right in rtl

column: same as row but top to bottom

column-reverse: same as row-reverse but top to bottom

I think in **column-reverse** you mean but bottom to up





SelenIT
12/12/2018 11:20:13
Firefox 22+ has unprefixed flexbox, but, unfortunately, it still doesn't support flex-wrap property (and hence flex-flow shorthand). So the wonderful example with 3-column layout reducing to 1 column on narrow screen in Firefox looks really messy. But it's possible to create its simplified analog that works in both Chromium-based browsers and Firefox 23+: <http://codepen.io/jamon/jan/pd9w>

Reply :



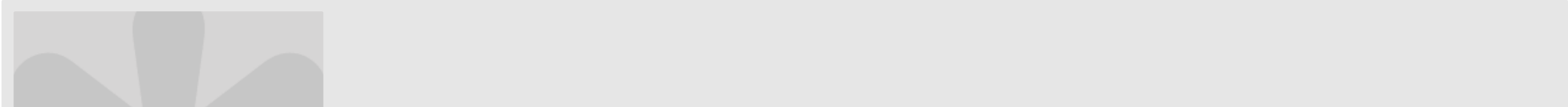
Jack Calder
12/12/2018 11:20:13
Wow, its really the one the best post i ever read on this topic. The steps which you have mentioned are really perfect.

Reply :



Arthur
12/12/2018 11:20:13
Hey, Chris! Looks like "flex-wrap" incorrectly works in Firefox and Opera at both tomato blocks and very last demos do not work in these same workarounds already.
And thank you so much for your website! :)

Reply :



User images via gravatar.com

SeleniT

• 10/05/2013

Yes, only latest Chromium-based browsers like Chrome, Opera 16+ etc. seem to support multi-line flexboxes currently. As a workaround, you can use nested flexboxes in combination with media queries, as in [my comment](#) above (it's not as flexible as true multi-line flexboxes, but still better than nothing) or use graceful degradation to old techniques like inline-blocks.



paceaux

• 09/10/2013

I've found that, in Chrome 29, <div> /> and <table> do not respect @font-face. Anyone else observed this, or have an idea as to why?

[Reply](#)

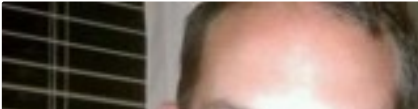


Grant

• 09/10/2013

Flexbox is what CSS has been sorely lacking since its inception - an easy way to create flexible web page layouts without the need for floats, clears, margin: 0 auto and JS tricks. I look forward to the day when flexbox is supported by a big enough share of the browser market to put into this all of our production sites.

[Reply](#)





Aron Duby

Thanks for the awesome tutorial, just managed to use the knowledge to make a sweet way to build tournament brackets! You can check out the codepen at <http://codepen.io/qbz/>

Reply :



Uncle Jesse

I find myself doing a Mr. Burns "excellent", as I'm pretty excited about [93199:154861](#) : STRETCH

Reply :



Randy Burleson

I am trying to make my video rich website "FLEX". The site scales ok but the Vimeo / frame videos do not. I was trying to use the FLEX.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. Using Firebug plug in the Firefox browser I saw this code about Flex Box... How do I modify this to make the videos Flex?

```
<!DOCTYPE html>
<html class="no-js" no-flexbox flexbox-legacy canvas canvastext webgl no-touch geolocation postmessage no-websocket database indexdbd hashchange history draganddrop websockets rgba hls multiplebgs backgroundsize borderimage borderradius boxshadow textshadow opacity csanimations csscolumns cssgradients no-cssreflections csstransforms csstransforms3d csstransitions fontface generatedcontent >
<head>
<body id="body" class="no-site-editor allow-collapse-transition">
</html>
```

Reply :



Reblutus
NOVEMBER 20, 2013
I really like this post. It got me started with my project.
I had a problem with Firefox like other users here but came over it by wrapping the columns/rows in more container like a user suggested.
I have another problem though. 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.
On my side I had a different problem with IE: the columns were showing but the items in them had no height! So everything collapses for no reason. Of course it's fine in Chrome and Firefox (25)

Reply :



Britton
NOVEMBER 21, 2013
There is a typo with the portion on flex grow. It doesn't inhibit understanding the content, but it would be nice if you fix it.

Reply :



Jesse
NOVEMBER 21, 2013

Reply :



mono

2 · 10/20/2019 10:00 PM

Chris, I couldn't vertically align some content in print media, do you know where I could find more information about this kind of support?

My test looks something like this,

```
css:
@page {
  size: US-letter;
}
article {
  page-break-after: always;
  text-align: center;
  display: flex;
  flex-direction: row;
  align-items: center;
  justify-content: center;
}
article:last-child {
  page-break-after: avoid;
}
```

```
html:
<body>
  <article>
    <h1>
      Hello
    </h1>
  </article>

  <article>
    <h1>
      Hello 2
    </h1>
  </article>
</body>
```



Carlos

4 · 10/20/2019

Not all browsers support paged media, does the paged media example work without the flexbox?



Michèle
#OCTOBER 22, 2022
What does 22+ (new), in the Firefox support table mean?

Reply 1



Benjamin Charity

Meaning version 22 of Firefox which is the newest version at the time the article was written.



Chris Coyer

And the + means "and up"



Andy
10/07/14, 2014
Nice tutorial. Has anything changed this this tutorial was published?
Also it doesn't work for me in IE 10.

Reply :



Chris Coyer
10/07/14, 2014
IE 10 has the [browser syntax](#), so make sure you're prefiring for that. Autofrefixer does a great job of writing in the latest syntax and handling the fallbacks.

Reply :



Mark
10/07/14, 2014

Reply :



Brad Spencer
4 months ago
Having trouble with 2 flexboxes aligned horizontally when one is set in column flow and the other in column-reverse flow.
See pen: [Flexbox Alignment Sample](#)
How do I fix this? Thanks!

[Reply](#) :



Justin
4 months ago
Simple fix I think. You need to set the container(window) to flex so that your 2
-divs are flex children.

```
body { display: flex; }
```



Brad Spencer

Www, that was simple!

Thanks Justin.



Jan

RECEIVED 4, 2014

My

to your first example, the child element has been centered by (margin: auto);

This solution does not work in IE11 if the child element has no defined height, for example, if the height is determined by the content.

All other browsers behave as expected.

Reply

```
css
parent {
  display: flex;
  height: 300px; /* Or whatever */
}
child {
  width: 100px; /* Or whatever */
  height: 100px; /* Or whatever */ /* Doesn't work in IE11, if the height is determined by the content */
  margin: auto; /* Magic */
}
```



Andrew

RECEIVED 4, 2014

I messed with this a bit today. 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. Also, if I load the entire page via jQuery, as I've been doing lately, the same result... Instead of the framed environment you're getting I received flat little lines. 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.

Reply





Andrew
#12000000 12, 2016
Would you happen to know how I could code a horizontal split (like they have on Code Pen) that separates the top of the window and the bottom of the window and moves fluidly when the bar is moved, with Flexbox framework? Any help would be appreciated, thanks!

[Reply](#)



Chris Coyier
#12000000 12, 2016
The draggable bar isn't going to happen with just CSS, Flexbox or no, save for some super nifty hack using a [resizable browser](#) or something. On CodePen we use jQuery UI's 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. 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.



Andrew
#12000000 12, 2016
Do you know of any working examples of jQuery UI Draggable for a horizontal split pane? I've been messing with it for a couple of days now and can't seem to figure it out.





Gimm

FEBRUARY 20, 2014

Hi Chris,
I'm trying to make a div which its width auto grow with its contents.
Using this:

```
display: inline-flex;
flex-flow: column wrap;
```

There seems a bug that with the container's main size, please see this pen
[wrong main size when flex-direction is columns](#)

Reply 1:



Mumtaz

Can u check Safari....
flex property not supported.



Mark F. Simchock

#cssflexbox · 3d · 2014

Well played. Thanks Chris.

This will certainly be a great tool to have once it's better supported. For now it seems to me it's best to lean on `flex`, or just stick to a design / layout that can be manufactured with less buggy / if you will / off the shelf parts.

If design doesn't consider manufacturing then that's not design. That's art. There's a difference.

Reply



Evert

#cssflexbox · 3d · 2014

Perhaps not the best place to ask, but I am struggling with making a responsive flexbox fluid layout. 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 on 3 are easy, I just change the flex-direction from column to row. But how about 2)?

So basically it must look like:

A

B

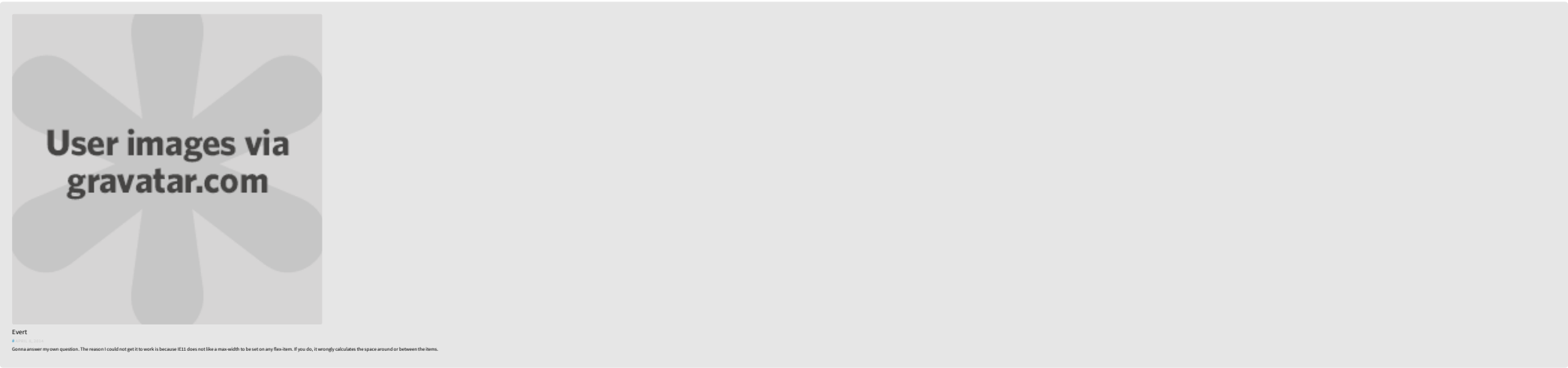
C

AB

C

ABC

Reply



Evert

#cssflexbox · 3d · 2014

Gonna answer my own question. The reason I could not get it to work is because IE11 does not like a max-width to be set on any flex-item. If you do, it wrongly calculates the space around or between the items.





Levi

Ever, I just ran into that same issue! I was loading my head against it for a good hour until I discovered that IE11 doesn't like max-width on flex items.



Jon

Great article, thanks. Regarding the the browser support table, I think that IE11 may have full support of the specification. But <http://msdn.microsoft.com/en-us/library/ee685027?v=vs.95> says otherwise. Thanks.

Reply :



Dwayne

Does using flexbox responsibly meaning coding the site via flexbox and usual css positioning methods as a fall back for browsers who dont support flexbox, coding the layout twice? Just thinking workflow wise...

Reply :





Michael Park
4 years ago
Thanks Chris! This is an excellent Flexbox reference. I have implemented a basic Holy Grail template: <http://meyerpark.me.uk/2014/03/26/a-holy-grail-flexbox-layout/>. Flexbox is a thing of beauty!

[Reply](#) :



Anton G
4 years ago
Nice job!
Thanks for sharing this.
I found this Polyfill for Flexbox, <http://flexjs.com/>

[Reply](#) :



Evert
4 years ago
Things I noticed using flexbox that are a real pain
Using margins & auto; on the flex container shrinks the container (and it's containing flex items) to the minimum width. It is no longer flexible/fluid.
Because of this, any fluid, centered layout must use justify-content: center() or space-between. But then the layout becomes "inflexible" (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 max-width on the flex container, but that cancels out the centering for some reason and the page flushes left. So the only other possibility is to set a max-width on one or more flex items...but those will break in IE11 because of some bug.
In short: flexbox will only work practically when using the full screen width and not limiting any flexible item with a max-width. As soon as you want to set a limit to any item, it falls apart.

[Reply](#) :





fred
4 months ago
I too see no other advantage for this than loading some lines in my media queries

Reply :



Stuart
4 months ago
This really annoyed me and was broken for a bit, so I wanted to share in case anyone ever comes across this in the future. If you need to support blackberry 7+, make sure you use

```
:-webkit-box-orient: vertical;  
-ms-box-orient: vertical;  
-webkit-box-direction: normal;  
-ms-box-direction: normal;  
-webkit-flex-direction: column;  
-ms-flex-direction: column;  
flex-direction: column;
```


...if you use row wrap, it doesn't wrap and just puts everything side by side. 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!

Reply :



Stuart
4 months ago
One last important thing to remember if you have to support blackberry 7+...make sure all child elements have 'float: none' applied to them...if floats are applied, they'll just not appear. I hope this helps!



Lauren
10/10/13, 2014
Buh, thanks so much, this helped me on Samsung Galaxy as well. Cheers!

Reply >



Ed
10/10/13, 2014
In the first line of the SASS mixin, shouldn't @ixin: !infix!() be just @ixin: !infix! ?

Reply >



Deepak
10/10/13, 2014
Ohh, this [example](#) does not work in IE11, could you please suggest, how I can have support on IE11



Chris
2
The wrapper default "mobile flex flow row wrap," only, add "flex flow row wrap," and it works in IE 11 and Firefox.



Najmul
2
Where are things:

box-orient
box-pack
box-align
box-flex
box-flex-group
box-ordinal-group
box-direction
box-lines





Chris Coyer
12/11/2014
These are deprecated properties. 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.



Abhishek Hingnikar
12/11/2014
Amazing writeup and excellently explained, you saved me fairly a LOT of time! would off spent learning all this combining all the broken and outdated articles over the web! :D thank you so much!

[Reply](#)



Scott
12/11/2014
This is a great article. I'd love to see the pens using the flex wrap updated with "flex-flow: row wrap;" added on-prefixed so they work in Firefox 28! But still a very good and informative article.

[Reply](#)





Gluten

4 years, 10 months

Is there a way to specify a minimum for inter-elements spacing when using `flex-wrap: wrap`? The only way I've currently found forces me to add a padding to the container which isn't ideal.

Reply :



BurninLeo

4 years, 10 months

To add spacing, use margin-right and margin-bottom. Give the container (the same, but) negative margin to still use the full width.



Ceah

4 years, 10 months

Please forgive my newbie ignorance.

I'm thinking that I would experiment with a background color of the site, then the container would be another color (centered) and then the flex items yet another color.

I get how to center the flex items themselves, but how would you center the container itself? And is that something one would even want to do?

Thanks

Reply :





Ceah
#1417, 2014
margin: 0px auto;
think I figured it out.....feet very dumb right now!



Guilherme Bruzzi
#1418, 2014
Hi Chris! Very nice article! But the last example "mobile first 3-columns layout with full-width header and footer" in my 34.0.1867.112 chrome didn't make the two sidebars half of the size of the main content.
I had to write:
@media all and (min-width: 800px) {
 aside-1 { order: 1; flex: 1 25%; }
 main { order: 2; flex: 2 1 50%; }
 aside-2 { order: 3; flex: 1 25%; }
 footer { order: 4; }
}
On the last media query in order to do that (<http://codepen.io/rlond>).

Reply :



Rory Matthews
#1419, 2014
Wow! I had bookmarked the article before and have come back to it today as a reference. Really like the re-hat, makes it even more useful! Cheers to you, Chris.

Reply :





Alan
6 years ago
Great work on the updated format! The guide was crazy informative before but now it's also a great cheat sheet when needed. Thanks!

[Reply](#)



Ry
6 years ago
Great guide, nice update! Has always been very useful.
One thing I've noticed missing here and almost every other flexbox guide is how to do flex-grow and flex-shrink in EJS.
flex-grow
--> flex-grow: 1; // or
flex-grow: 1 1 0;
flex-shrink
--> flex-shrink: 1; // or
flex-shrink: 1 1 0;
Would be great to have this footnoted somewhere.

[Reply](#)



Neil
6 years ago
@Ry, good point. I happen to use Autoprefixer, which added this (a specific property name in for me, I wouldn't have known otherwise.



Daniel Berthiaume

• MAY 15, 2014

I love all that can be done with the flex box model, now only if all the browser could support it the same way! How does the flexbox fall on browsers that don't support the CSS3?

Reply 1



Bob Prokop

• MAY 15, 2014

Thanks so much for updating this post – by far the easiest to understand guide to flexbox ever written. You deserve at least a six pack of *Budling Beck* for this one, Chris – (if that's still your brew of choice that is :)

Reply 1



stephen

• MAY 15, 2014

Hi,

I was hoping someone might be able to help me out (I'm pretty new to all of the programming stuff).

I created a flex box and arranged the items in it in a column layout. I then did "justify-content:center", but the elements stay on the left hand side of the screen, even though the width of the container is 100%. Is there an easy way to center everything in a container box when arranging elements as columns? Hope this makes sense.

Reply 1

Cheers,
Steve.



Jay
Hi Stephen, I believe that justify-content isn't to be used for this purpose. 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: center.



stephen
Hi Jay,
Thanks for getting back to me so quickly. Ah yes, I guess because I didn't set a height on the flexbox, I didn't see how the elements were centering vertically.
Thanks,
Steve.

Reply





Ethan
#1547 21, 2014
Does Compass support flex box? 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. Like @include display flex? I'm unable to get this working locally however. Ideas?

[Reply](#)



User images via gravavatar.com

Jozsef Remen
#1547 21, 2014
Forget about Compass and use Autoprefixer instead (with gulp/grunt). Personally, I just use it for vertical rhythm calculations now as Compass will be big no no for all kinds in C++.



Srbiotik
#1547 21, 2014
Hey I'm interested in why this background: darken(color, %) part of code is not working for me, I tried to do it by myself with my own examples and it didn't work so I pasted your code from codepen and it still doesn't work. I'm sorry if I'm asking a noob question and there is something obvious that I've missed? Because I haven't been clear that the example that concerns a making of dynamic navbar? Thanks a bunch!!

[Reply](#)





Chris Coyer
✎
That's a SaaS function.

[Reply ↗](#)



Srbotik
✎
Oh, sorry I forgot I'm using the latest version of Firefox and Chrome!

[Reply ↗](#)



Matt
✎
Bit of a long shot here, but do any Email clients support Flex box...? Would be useful in HTML emails to rearrange the order of elements.





Yehuda
10/16/10, 10:14
Testing flexbox in Safari now.
What works in all other browsers, either doesn't work in Safari, or doesn't work correctly.
Really frustrating...
The demos here don't work correctly either (especially the last one).

[Reply](#)



pankaj
10/16/10, 10:14
this property not working android 4.1.1 browser. How it will be work on mobile browser

[Reply](#)



Scott Vandehey
10/16/10, 10:14
I think the Support Chart is out of date for Safari. Should read:
6.1+ (iOS)
5.1+ (iOS)

[Reply](#)

According to <http://beta.caniuse.com/#feat=ch-flexbox>





Scott Vandehey
Similarly, Android 4+ (new), iOS 7.1+ (new)



Chris Coyler
Thanks Scott

User images via gravatar.com

Brian Hughes
How do you all know what works in which browser version? Where is Firefox standing now for support?
I just learned about Firefox yesterday so now I'm all anxious to learn more. I'm a little hesitant because of browser version support.

Reply

User images via gravatar.com

coolcat007

You can find more detailed information about browser support when you type in "caniuse flexbox" in google.

User images via gravatar.com

Stephen

Hi, I was wondering if anyone could help me out with a flexbox problem. I've set a container width to 100% and put six divs 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 row wrap). 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:

[View](#) [Edit](#) [Embed](#)

[Reply](#)

Any help would be much appreciated.

Thanks,

Steve.

User images via gravatar.com

Yehuda

try align-content: flex-start; on the container. 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 20%; on each sub item instead of specifying the width (again, it depends on what you want to do).



Saman

```
#divcontainer: {  
  align-content: flex-start;  
}
```

User images via
gravatar.com

Stephen

Hi there,

Thanks for both of the tips; the first one works well and solves the problem I was having.

If you have time, I was hoping you might be able to elaborate on the second one a little. 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's clearly taking up more than 20% of the container now? Anyway, don't mean to be lazy; I can look this stuff up tomorrow. Time for bed in the UK though.

Cheers again,

Steve.

Reply :

User images via
gravatar.com



NeedHate
2017-12-20 09:44
Guys, what about "order"? It doesnt look good in safari, even doesnt look anymore. B) how to make it work in safari?

[Reply](#)



Yehuda
2017-12-20 09:44
I gave up on Safari. Not supporting it on my sites.
You could just want to focus for it, but when I discussed it with my employer he said "no one uses it anyway".
@Stephen, play around with Ret: 1:1 20%

[Reply](#)



Johnny
2017-12-20 09:44
Safari 5.1.1
Works only this:
@Stephen, make it work!
And that's it. Nothing else can make work :(
I've read that this version of Safari is (old), but how it should be like?
Can't handle it...

[Reply](#)





wilbur
2017-01-27
In the first example (with the 6 orange squares)... is there a way to request the current number of columns and rows within a flexbox container? or at least the current number of rows (since the columns are not rigid)?
thanks!

Reply :



Pawel
2017-01-28, 2017
Hi
This guide is wonderful, seriously, guy who did this deserve a BIG nice glass of GOOD beer.
But I have issue:
I made a website, where container's div is flex and direction is column.
I made this container (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, I just want to use only flex box model.
Regards, mates!

Reply :



Pawel
2017-01-28, 2018
Ok, I got it, there was no question id Sorry. Thanks anyway! This is best place to learn CSS Tricks.
Regards again!



width: 100% and order: the last div in y

Reply:



JULY 2011

And why width....



JULY 2

I did it that way

<https://www.dropbox.com/s/yfawccmz7hzmikpdb/Zrztu%20krans%20r%202014-07-29%2011%3A57%3A31.png>

Do you see any issues that may be?



Hasschi
1000007 20, 2014
10017 size webkit justify content
justify content doesnt work

[Reply](#)



Gopinath
1000007 20, 2014
14
My requirement is need to alignment support all browser without use Javascript. Use only CSS/CSS3.
Note: Particular para line Margin top value support all browser (Mozilla, Chrome, Safari) as per match PDF. But IE-11 browser some different its will come_, in case for adjust IE-11 browser, at the time margin top value change another browser_. So, how to modify all browser requirement. If any possible on that particular IE-11 alignment modification style sheet.
Please give any solution that issue.

[Reply](#)



Anne

AUGUST 18, 2014

I am working with flexbox on a few different projects now and love it. Only downside is all the prefixes that you need.

For my projects I made a less mixin stylesheet that has been tested and works in the most recent browsers (latest version -3). Hoping to help some more people out I put it on my [github](https://github.com), so if you want a little help getting started you can grab it there github.com/annebosman/Flexbox.css

Reply :



Fredrik

AUGUST 22, 2014

I've been experimenting with `Flex-urap` recently, and found that Safari doesn't support it (on desktop or mobile), although it claims to, i.e. `Modernizr.Flexurap` is true. I've [filed a bug report with Modernizr](#) for this. 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.

Reply:



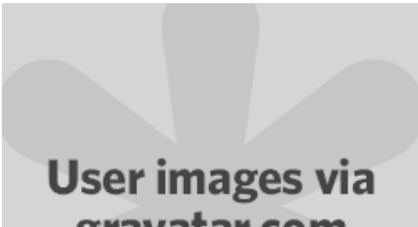
Lester

SEPTEMBER 8, 2014

it seems many properties aren't supported by safari: https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Flexible_boxes

something as important and necessary as wrap makes it a no-go for me (but i'm a new-b)

plus i think that, as great as it is [and C.C. knows how much i love him], combining old and new is still another hack that flex box was supposed to eliminate and i ain't got time for that!





Jarek

9/17/2022 1:20PM

[Reply](#)

Hi,
I'm trying to build simple layout. Could anyone help me with this? I was written some code reading article.
Want to know this.

[Try to open this \(I want to display in this way\)](#)

But new block number four is moved to center and on the bottom of block number two (whole layout). I want to get it on the right side of the block number two, but below of the block number three.
(I must remove because message was rendering in wrong way)

```
body
{
  class="flex-container"
  li class="flex-item1" /li
  li class="flex-item2" /li
  li class="flex-item3" /li
  li class="flex-item4" /li
}

ul
{
  flex-direction: row wrap;
  justify-content: space-around;
  align-items: stretch;
}

.flex-item1 {
  background: tomato;
  line-height: 50px;
  width: 650px;
  height: 50px;
  margin-top: 0px;
  color: white;
  font-weight: bold;
  font-size: 0.50em;
  text-align: center;
}

.flex-item2 {
  background: tomato;
  padding: 0px;
  width: 320px;
  height: 50px;
  margin-top: 0px;
  line-height: 50px;
  color: white;
  font-weight: bold;
  font-size: 0.50em;
  text-align: center;
}

.flex-item3 {
  background: tomato;
  padding: 0px;
  width: 320px;
  height: 50px;
  margin-top: 0px;
  line-height: 50px;
  color: white;
  font-weight: bold;
  font-size: 0.50em;
  text-align: center;
}

.flex-item4 {
  background: tomato;
  padding: 0px;
  width: 320px;
  height: 50px;
  margin-top: 0px;
  line-height: 50px;
  color: white;
  font-weight: bold;
  font-size: 0.50em;
  text-align: center;
}
```





Gabe S.
#107750593, 06/06/17
In case someone is trying to do a grid layout using flex, I found this helpful for aligning items in the last row: <http://codepen.io/Gabedjpen/pen/Dlmevs>

Reply :



Tyler
#107750593, 06/06/17
Flicker 'love this update' Sad to think we've still another few years out from implementing this without fallback support. :(

Reply :



Alex
#107750593, 06/06/17
I have a flexbox container.

Reply :

```
display: -webkit-box;
display: -ms-flexbox;
display: flex;

-ms-box-direction: column;
-webkit-box-direction: column;
-ms-flexbox-direction: column;
flex-direction: column;

-ms-box-orient: horizontal;
-webkit-box-orient: horizontal;
-ms-flexbox-orient: horizontal;
flex-orient: horizontal;

justify-content: space-between;
align-items: stretch;
align-content: space-between;
height: 100%;
width: 100%;
background-color: purple;
```

Inside this container, I have two items. A content area and a footer. 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 want the footer to have a set height of 50px and I want the content region to automatically fill the rest of the empty space.

What CSS is needed for the content area to fill the remaining space relative to the footer?

I want to be able to infinitely expand the browser window and always have my content area fill the empty space and I never want the footer to change size.

Any help would be greatly appreciated, thanks!



Alex

I figured it out. Here's the solution that I came up with:

```
.masterContainer > .content {
  flex-shrink: auto;
  flex-grow: 1;
  background-color: yellow;
}
.masterContainer > .footer {
  height: 50px;
  width: 100%;
  background-color: blue;
}
```



Ken

#OCTOBER 8, 2024

I started on an idea for HTML as a presentation format using flex.

<https://github.com/kenwright/html-as-presentation>

Reply





Richard C

4/11/2014 12:24 PM

Hi

Can someone point me to a tutorial or demo of using frame within a flexbox container. I have tried and it is failing to keep aspect ratios and the usual padding trick doesn't seem to work. Alternatively is there an easy solution you could give me here.

Many thanks

Richard C

[Reply](#) :



Dan

4/11/2014 12:25 PM

After reading your great article on how to use flex box, I came across this [article](#) that says don't use flex box for over all page layout.

Any comments on how valid the above article is, if it's valid is there any work around to still using flex box for page layout without the performance hit?

Thanks,

[Reply](#) :



coolcat007

4/11/2014 12:25 PM

I kind of agree with the article. Both Flexbox and Grid layout have their pros'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 screen sizes, but are optimized for different applications.

[Reply](#) :



User images via gravatar.com

Alex Wilkins

10/10/2008 11:20:14

The middle first 3 columns layout doesn't work when adding a paragraph to the aside. 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.

Reply



yan

10/10/2008 11:20:14

the initial value of 'flex-basis' is 'main-size', and if omitted in the shorthand property 'flex', it's value is '0%'.
<http://www.w3.org/TR/CSS3/flexbox/#shorthand-flex-basis>

Reply



ArleyM

10/10/2008 11:20:14

I just read that too, but when I was tinkering with it in Chrome only auto worked!





yan

The specification says `flex: auto` is `flex: 1 1 main-size`, to be distinguished from `flex: 1 1 auto`. Currently only Firefox 34+ support 'main-size'.
<https://developer.mozilla.org/en-US/docs/CSS/flex-base>



fantasai

This is currently under discussion, like it says in the big red box there. We're actively looking for feedback on that issue at the moment, so please let us know if any!



fantasai

This is a pretty good quick guide. Just a couple things noticed from a skim:

It's not very clear how 'order' actually works. 'order: 3' doesn't mean "put it at the third position", it means "put it after any items with 'order' less than 3 and before any items with 'order' greater than 3". We (the Firefox spec editors) strongly recommend not using the longhands of 'flex' unless you really, really want to cascade in flex settings from some other style rule, so I'd suggest somehow discouraging the use of 'flex-grow/shrink/basis' here (or, preferably, leaving it out/in an advanced section). The shorthand resets things in appropriate ways, and will therefore result in fewer cascading errors. Please use the shorthand!

Reply



Neil

• 10/10/2016

I tend to think of flex "order" as a index for flow items. Maybe this will help others to visualize it this way also.



vroom

• 10/10/2016

There is currently a crippling bug in Firefox that makes any non-trivial implementation of flex unfeasible. Needing a few flex'd containers causes Firefox to become unresponsive. https://bugzilla.mozilla.org/show_bug.cgi?id=1382780

Reply



neomwired

• 10/10/2016

Lots of bugs with it on iPad too, so it's pretty much unusable currently



Phil

4 NOVEMBER 20, 2019

Thanks for the article, helped me a great deal bringing my LESS implementation and flower package up to date!
(free to use at <https://github.com/philipprichter/cach-floc> or via <http://browser.js/search?q=cach-floc#20190908>).

[Reply](#) :



Paul

4 NOVEMBER 20, 2019

Hi, I'm looking for the way to do a fullscreen menu for my website with flex, with a header on the top and the rest of the space with only 6 big responsive buttons. I've tried many things and I've check many websites. I would appreciate any help. Thanks in advance.

[Reply](#) :



Justin T

4 NOVEMBER 20, 2019

This is one of the best code tutorials I've ever seen. Kudos for taking the time to make this super intuitive.

[Reply](#) :



Luke
#comment 12, 2015
This is going to be an amazing feature right now. 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.

Reply :



KillerDesigner
#comment 13, 2015
Sean Fioritto (a/p) produced a great video (and a book!) on Facebook, entitled "Sketching with Facebook", if anyone is interested. The video lesson link: <http://www.sketchingwithfb.com/facebook/> and a free lesson too! <http://www.sketchingwithfb.com/facebook-lesson/>
Enjoy!

Reply :



Matthew Dixon
#comment 14, 2015

Reply :

So I was wondering, is there a good way of making the child elements of the flex grid not automatically span to the full width of the page. Only specifying widths every time is not very effective. No one should have to add a `width: 250px;` to every element within if they want it to behave properly. Thanks



Michael C.
#cssflexbox · 12/1/2015
Tons of love to Flexbox which just saved my weekend. 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 row-major format to column-major format. So I used Flexbox to lay out the columns in left-to-right (row) direction, and then lay out each child in each row in top-to-bottom (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 "column-reverse". 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. Woo hoo!!!!

Reply ↩



Ville Vanninen
#cssflexbox · 1/1/2015
Thanks Chris! I made a flexbox robust config thingy / cheat sheet for quick copy & paste, based on your article. I've been using it a lot for my own projects, might be useful for others too. <http://apps.worflower.fi/css-cheats/transform-flexbox> (also on github if anyone cares to fork/improve/whatever <https://github.com/vukanninen/css-cheats>)

Reply ↩



marco
#css #css3
Great work man this inspired me this little css library
<http://hachach.github.io/css3floatables/>

[Reply](#) :



Jared Proke
#css #css3
Your first example at this link (<http://codepen.io/Hugobiraudel/pen/LAC/>) does not work in IE 11. It doesn't seem to like `width: flex: flex; flex-direction: row; or just flex: flex: row wrap; works though.`

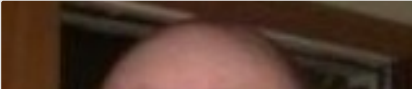
[Reply](#) :



Pikosan
#css #css3
hey guys, need help here. I wrote 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). Here is the code:

```
@media screen and (min-width: 1440px) {  
  .wrap{width: 100%; margin: 0 auto;}  
  #about{width: 500px;}  
  #container  
  {  
    overflow: hidden;  
    margin: 0 auto;  
    margin-top: 70px;  
    width: 1000px;  
    padding: 0;  
    justify-content: space-between;  
    list-style: none;  
    display: -webkit-box;  
    display: -ms-flex;  
    display: -webkit-flex;  
    display: flex;  
    -webkit-flex-flow: row wrap;  
    padding-right: 5px;  
    padding-left: 5px;  
  }  
  #skills{width: 250px; float: left; }  
  #portfolio{ margin-left: 7px; width: 250px; }  
  #certificates{width: 250px; float: right; }
```

[Reply](#) :





Alex

For starters, you don't need floats. That is the whole point of Flexboxes. So you don't have to use floats. Trying to rid of the float declarations and playing around some more...

Also, keep in mind that every set of flex items needs a flex container. It's not redundant to see something like

```
<div class="flex-container">
  <div class="flex-item">
    <div class="flex-container">
      <div>
        </div>
      </div>
    </div>
  </div>
```

Having flex boxes is how you keep consistency across browsers but it can get really confusing really quick. Especially when you get like 8 levels deep.

You also are probably missing LOTS of vendor prefixes to get it working properly across all browsers.

For instance, you might want to take a look at the classes that I use in my projects to see what you are missing

```
.flex-it {
  display: -webkit-box;
  display: -webkit-flex;
  display: -ms-flexbox;
  display: flex;
  -ms-box-orient: none;
  -webkit-box-orient: none;
  -ms-flex-direction: row;
  -webkit-flex-direction: row;
  -ms-flex-wrap: none;
  -webkit-flex-wrap: none;
  flex-direction: row;
  flex-wrap: none;
}

.flex-row {
  -ms-box-direction: row;
  -webkit-box-direction: row;
  -webkit-box-orient: horizontal;
  -webkit-flex-direction: row;
  -ms-flex-direction: row;
  -webkit-flex-direction: row;
  flex-direction: row;
}

.flex-col {
  -ms-box-direction: column;
  -webkit-box-direction: column;
  -webkit-box-orient: vertical;
  -webkit-flex-direction: column;
  -ms-flex-direction: column;
  -webkit-flex-direction: column;
  flex-direction: column;
}

.flex-align-between {
  -webkit-box-align: center; space-between;
  -webkit-align-content: space-between;
  -ms-flex-align: center; space-between;
  align-content: space-between;
}

.flex-align-center {
  -webkit-box-align: center; center;
  -webkit-align-content: center;
  -ms-flex-align: center; center;
  align-content: center;
}

.flex-align-start {
  -webkit-box-align: center; flex-start;
  -webkit-align-content: flex-start;
  -ms-flex-align: center; flex-start;
  align-content: flex-start;
}

.flex-align-items-start {
  -webkit-box-align: flex-start;
  -webkit-align-items: flex-start;
  -ms-box-align: flex-start;
  -ms-flex-align: flex-start;
  align-items: flex-start;
}

.flex-align-items-center {
  -webkit-box-align: center;
  -webkit-align-items: center;
  -ms-box-align: center;
  -ms-flex-align: center;
  align-items: center;
}

.flex-start-all {
  -webkit-box-pack: justify;
  -webkit-justify-content: flex-start;
  -ms-flex-pack: justify;
  justify-content: flex-start;
  -webkit-align-items: flex-start;
  -ms-box-align: flex-start;
  -ms-flex-align: flex-start;
  align-items: flex-start;
  -webkit-box-align-content: flex-start;
  -webkit-align-content: flex-start;
  -ms-flex-align-content: flex-start;
  align-content: flex-start;
}

.flex-align-items-stretch {
  -webkit-box-align: stretch;
  -webkit-align-items: stretch;
  -ms-box-align: stretch;
  -ms-flex-align: stretch;
  align-items: stretch;
}
```

```
.flex-justify-between {
  -webkit-box-pack: justify;
  -webkit-justify-content: space-between;
  -ms-flex-pack: justify;
  -ms-justify-content: space-between;
  justify-content: space-between;
}

.flex-justify-center {
  -webkit-box-pack: justify;
  -webkit-justify-content: center;
  -ms-flex-pack: justify;
  -ms-justify-content: center;
  justify-content: center;
}

.flex-justify-start {
  -webkit-box-pack: justify;
  -webkit-justify-content: flex-start;
  -ms-flex-pack: justify;
  -ms-justify-content: flex-start;
  justify-content: flex-start;
}

.flex-justify-end {
  -webkit-box-pack: justify;
  -webkit-justify-content: flex-end;
  -ms-flex-pack: justify;
  -ms-justify-content: flex-end;
  justify-content: flex-end;
}

.flex-wrap {
  -ms-flex-wrap: wrap;
  -webkit-flex-wrap: wrap;
  -webkit-flex-wrap: wrap;
  -ms-flex-wrap: wrap;
  -ms-flex-wrap: wrap;
  flex-wrap: wrap;
}

.flex-items-auto {
  -webkit-box-orient: auto;
  -webkit-flex-basis: auto;
  -ms-flex-basis: auto;
  flex-basis: auto;
  -webkit-box-flex: 1;
  /* OLD - iOS 6-, Safari 3.1-6 */
  -ms-box-flex: 1;
  /* OLD - Firefox 18- */
  -webkit-flex: 1;
  /* Chrome */
  -ms-flex: 1 0 auto;
  /* IE 10 */
  flex: 1;
}
```



Neil

4 12, 2016

If you have the option to use Autoprefixer, this could help a lot with the vendor prefixing.

<https://github.com/postcss/autoprefixer>

User images via
gravatar.com



Gabe

10 NOVEMBER 4, 2020

I've taken the navigation layout above and put it in the header of the header, aside, main, aside, footer, layout.

What I want to do is fix the navigation/header and have it the width of the page with the other elements remain in their position below the header.

Here's my pen



[Reply](#)



User images via
gravatar.com

ionlyseespots

10

Does this help? I put your HTML5 within the Ambient framework.



User images via
gravatar.com

oneblackswan

10

It's great that you have given the html, css and result, but I used yours exactly and it's fine on my laptop, but on my Android phone the header, main, aside, aside2 and footer are all on the same line (both portrait and landscape). I find a difference between resizing my laptop monitor and actually viewing it on other devices.

[Reply](#)



ionlyseespots
10/05/2017 at 10:00
I can potentially tag it as an issue in within Ambient.



Peter I.
10/05/2017 at 10:00
Hi everyone,
I've been working on this layout which I managed to work perfectly in modern Firefox & IE browsers, but it's not working as expected in Chrome and Safari (which leads me to believe I'm not implementing the flex box correctly).
Any advice would be greatly appreciated... I've tried all manner of logic in cluding flex box within a flex box to make this work... perhaps it's a limitation of the way flex box is being implemented in webkit browsers or vice versa.
I've posted the hard file here: http://www.danagross.com/test_layout.html
In Safari and Chrome, the contents do not fit perfectly in the browser window, and the footer div tag is not visible at all.

Reply :





Pavle

• 1 comment on 2018

I noticed when declaring flex property for parent that hold some elements (for example ul is flex, li are flex items (they are inline or inline-block)), when I set to some flex item margin-right:auto, it push all other elements to the edge of the parent container?

Reply :

Neil

• 1 comment on 2017

Thanks, as always, for a very informative post. It really fast-tracked my understanding of using the flexbox model.

One of the hardest things to wrap my head around was the flex-grow, flex-shrink and flex-basis properties. Not so much the concept of what they were, but how the actual values played out.

My basic assumption at first was that I set the flex-basis to a static size, say 200px, and flex-grow of item X to "2" and the other items in this container to "1", that the width of item X would be exactly 2 times the width of any of the others. This was not the case. It was always greater than 2 times.

After looking a little closer at the numbers it was applying, the first thing I noticed was that the flex-grow/flex-shrink is a ratio of these values amongst all children in that flexbox for that specific property. The grow and shrink values have nothing to do with each other.

As in the example given above, the ratio would be 2:1 for item X's width to the flex-basis value. 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 flex-shrink).**

That being said, the key is that if you subtract the basis width from each item width, then the remaining width will follow the ratio.

Now, if you are not setting the flex-basis properly manually, then the default will be "0%" and the ratio is closer to being what you would think, but there is still a minimum width on these elements that is factored into the ratio calculation as described above.

Hopefully, because flexbox is being used, the ratio won't need to be exactly correct and the layout will still look and work great. That's the whole point of flexbox, right?

I just wanted to share this extra information with those who like to understand where the numbers are coming from when it doesn't come out as you may have thought at first.

Reply :



Clare

• 1 comment on 2018

I'm having this problem and it's sooo confusing!





Ivan Kleshnin

4 months ago

Warning! Description of justify content / align-items is incorrect. Behavior of the last two changes depending of flex-direction. Article says it should be independent. "This defines the alignment along the main axis." No! If flex-direction = column, that will align items along the cross axis. To align items along main axis you'll need to change align-items instead.

[Reply](#)



Chris Coyler

4 months ago

When you change the flex-direction, you're changing the main axis. That's how I think about it anyway. Flex-direction:

establishes the main axis

[Reply](#)



Vova

4 months ago

Here, "Let's try something else. Imagine we have a right-aligned navigation on the very top of our website, but we want it to be centered on medium-sized screens and single-column on small devices. Easy enough."

The navigation don't works in Chrome 41.0.2272.101 m

[Reply](#)





Glenn Dixon
Just inherited a project with over a thousand products in dozens of categories/sub-categories. Alignment was all wonky. Just fixed it by adding TWO flexbox items into CSS. This site rocks!

Reply :



Alan
I have seen this code in the wild but it seems like a bad idea.

```
+ {  
  min-width: 0.1 auto;  
  flex: 0.1 auto;  
}
```

Can you help me understand why this is or isn't bad.
Thanks

Reply :



Knight Yoshi
It's not really 'bad' per say, it's just cross-browser for IE. It's ugly code, most people use a post CSS processor like Autoprefixer.





lauchoverit

I'm new to flexbox and certainly don't want to spread my noob confusion, but I noticed a couple things:
* Regarding this image - <http://www.w3.org/TR/css3-flexbox/images/autosize-flex.png>. Initially I thought this was super helpful. However, when I looked at where it's used in the w3 spec, it doesn't actually talk about using "auto" as a value for flex-basis at all (just a value for the "flex" shorthand), it just has it in the image for some reason - <http://www.w3.org/TR/css3-flexbox/#flex-property>
* Then I found this section of the spec, and I took like using "auto" as a value for flex-basis is to debate - <http://www.w3.org/TR/css3-flexbox/#flex-basis-property>
Questions:
* Should we avoid using "flex-basis: auto" for the time being? And if so, should there be a note accompanying that image?
* Am I right in thinking that the w3 spec is a bit confusing/disorganized in those places? Worthy of me sending a comment/email to somebody?
Lastly: Very, very grateful for this post. Thanks!

Reply



Michael
I just do this most of the time:
flex: 0 0 auto; or flex: 0 0 25%; or flex: 0 0 100px;
I think it's easier just to use the shorthand property, and have a play with the values.



Marc
Originally "auto" meant "content" or natural size. Now auto means look at the height/width property and a new value of "content" has been added. Chrome is still treating "auto" like "content". Firefox and IE are not.



netdog
#1760, 5, 2015
Why did you add the classes?

```
<
header class = "header"> header </header>

Do not write now (hard),
write correctly is necessary on.
<

header> header </header>
I tried to remove the ALL classes, but the site is broken. I do not understand.
```

Reply :



Rahul Kumar
#1760, 5, 2015
These cs are like readymade ui bootstrap components or angular stuff. They work off the shelf. Web pages development are becoming breezy now, given most of the common burden is taken by the framework. Love it, thanks!

Reply :





Michael
17 Feb 24, 2023
What bothers me, is if you use either flex-direction: row; or flex-direction: column; It dictates what property you use to center objects horizontally. Maybe I don't just understand the logic.

Like
Dislike
Report
Get on

Reply :



Andy Mahesh
17 Feb 24, 2023
I think align-items and justify-content got mixed up in the example shared. Also, you the container article is missing a height, which ends up in confusing the result of applying align-items and justify-content as the same in that special case. Here is an example that might help clear this up for you I hope:
See the Pen [W7mndu](#) by Andy Mahesh ([@AndyMahesh](#)) on [CodePen](#).

Reply :



Kevin
17 Feb 24, 2023
Another great article!
Using this page as a guide and reference, I created a web-app based log in template that looks like a phone-app. 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.
Here's the result: <https://github.com/ken49133/projects/blob/main/login.html> (complete w/ "hamburger" nav)
Here's the folder (sorry, no pics): <https://github.com/ken49133/projects>
It uses a full mix of css flex props including a flex column w/ nested rows and nested traditional css (no floats)
Thanks for another great article, Chris!!

Reply :



Leonard Berman

17/06/2016, 10:30AM
Thanks for posting. Very interesting. I'm using the hamburger from your fiddle. Is there a particular attribution you would like?



PaulOB

17/06/2016, 10:30AM

I notice that the 3 column demo at the end is not working and should there be more content in the sidebars than the one word shown then the columns stretch to 100% width and break the layout.

The width of the side columns need to be set.

e.g.

```
media all and (min-width: 600px) {  
  .side {  
    flex: 1 0 0;  
  }  
}
```

[Reply](#)





Chris Clapp

I really like the concept of flexbox, but with needing to support IE's, looking for a way to do that with a graceful fallback or is it better to just style it "traditionally" for no flexbox (using Modernizr)?

Reply :

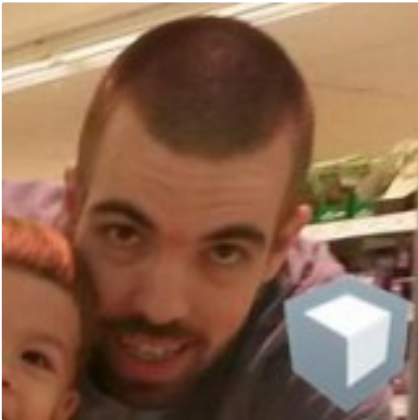
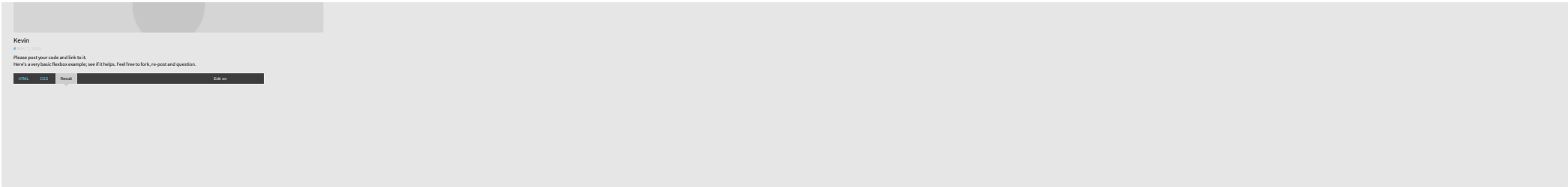


Kyle

Total noob when it comes to flexbox, but I was wondering something. Is it possible to have, in a list comprised of multiple rows, the first row "space-around," 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 "space-around" 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.

Reply :





Marc Dix
#css #flexbox
When using the flex shorthand in Safari 7 (1.6) (`flex: 1`) without specifying the third parameter (`flex: 1 1 1`), Safari will compute the value `flex: 1 1 1` and wrapping via `flex: 1 1 1` is not going to work. In order for Safari to wrap via flexbox `flex: 1 1 1` must be `flex: 1 1 1` (which is Safari's default value). So, if you use the shorthand and don't want an initial size for your flex item, set the third (or the second parameter if you leave out shrink) to 'auto' (i.e. `flex: 1 auto` or `flex: 1 1 auto`).
You can check this behaviour this codepen: <http://codepen.io/marc/dip/p/10000>

Reply



Phil Zito
#css #flexbox
Good article, I just shared on Twitter. Really like how you formatted it, the other articles on the flex box suck compared to yours.

Reply





Chris Deacy
10/02/17, 10:10
This is super helpful. Thank you sir.

Reply :



knazark
10/02/17, 10:10
Custom Flexbox Grid using Bootstrap mixins (SASS)
<https://www.creativeclouds.com/blog/programming/custom-flexbox-grid-using-bootstrap-mixins-sass/>

Reply :



JHoni
10/02/17, 10:10
I have your? Resolved my problems... :)
- Jonathan
flex-wrap: nowrap; / wrap; / wrap-reverse;
}

Reply :



User images via
gravatar.com

JULY 10, 2011

@Alex: maybe i
don't see it right



JULY 25, 2021

It's worth noting

```

.menu-item {

```

This works fine

©media all and

```

menu-item {
  font-size: 1.2em;
}

```

3

Journal of Management Education 34(1)

3

1

This takes acco



JULY 27, 2011

Having just refe



Alban
4 months ago
Great really great guide, well explained with examples. Thank you very much.

[Reply](#)



Kp
4 months ago
In the event anybody is having issues getting it to work on Firefox for the 2nd example (tomato background)
Put the flex items into their own container with no other element in them.
Add "flex-flow: row wrap;" to flex container
Hope this helps.

[Reply](#)



VNTproYKT
1/10/2017 22:10:13
Wow, this article is the coolest material about Flexbox.
People, now I need help with this:
<http://stackoverflow.com/q/32228436/239067>
Share please!



Antonio
1/10/2017 22:10:13
Wow, it could've better explained! thanks a lot!



Seasalt
1/10/2017 22:10:13
Thank you for the tutorial, 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 chat's container and the centre's content) I was trying to update to use flex like in the tutorial, but it's not worked. It looks like the centre might be the cause. Any ideas? Here it is on Codepen: <http://codepen.io/seasalt/pen/0p22mG>



Reply :

Reply :

Reply :



Tadeusz Łazurski
SEPTEMBER 24, 2015
Hello, I have stumbled upon this interesting [StackOverflow question](#) re "justify-content: flex-start" and "margin: auto" on a container. I don't know the answer and I wonder if there is any solution to this.

[Reply](#) :



Idanow
SEPTEMBER 24, 2015
I think for "align-content", the container should already has been propped up by some elements or in a fixed height. Can tell the reader of this in advance.

[Reply](#) :



Dan
SEPTEMBER 24, 2015
Chris, can you give us an example of what are small-scale layouts and large-scale layouts? I don't completely understand the Note about the best use for Flexbox vs. Grid.

[Reply](#) :





Ed

• 2019-03-20 10:00

Excellent article. This is the best explanation of Flexbox I have seen so far.

Reply :



User images via
gravatar.com

Enriep Yrekcaz

• 2019-03-20 10:00

Thanks so much for the article! 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 small scale 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 in flux. 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.

Thanks again! Keep up the great work!

Reply :



Paul Brady

• 2019-03-20 10:00

To make Flexbox play nicely with iPhone/iPad, add the following metatag...

Cheers!

Reply :





Paul Brady

4 years ago

(Or, correctly...)

To make Firefox play nicely with iPhone/iPad, add the following metatag...

```
<meta name="viewport" content="width=device-width">
```

Cheers!

Reply



Mirko

4 years ago

The 2nd example works fine without flexbox, with "display:inline-block". Less code and it works even with old browsers. See <http://codepen.io/antonjans/Vdub8P/edit#src=110>

Reply



Dan

4 years ago

Try adding a background color to the `...navigation` `div` and you will see that they are not the same. Using `inline-block` keeps you dependent on the browser default use of extra space left and right of inline `div` elements. This rendering can be fixed by floating the `div` elements, but `flexbox` is a nicer (modern) way of achieving that effect.





Matt G

OCTOBER 13, 2015

<http://daveedlively.com/blog/love/>

[Reply](#)



User images via
gravatar.com

Alex

OCTOBER 13, 2015

flex-basis seems to have got some updates, **width: auto** no longer works, **width** means look at width / height property (previously main-size), new **content** keyword means automatic sizing (previously auto)

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

[Reply](#)



Chris Coyler

#

Indeed! Updated.



User images via



VOVB
@vovb1979 1, 2015
order default: 0

[Reply](#) :



Vince
@vince1979 1, 2015
I have `float:wrap;` `wrap;` and `align:center;` `float:left;` set and the way I understand it from this guide, the last row should be aligned to the left of the other flex items, but it's not. It's centered. I must be misunderstanding something. What am I doing wrong?
L'entente est que si une commande aligne des flex-items, elle les aligne tous par rapport à une seule ligne de base. Si vous utilisez `float:left;`, les éléments sont alignés à gauche. Si vous utilisez `float:right;`, les éléments sont alignés à droite. Si vous utilisez `float:center;`, les éléments sont alignés au centre.

[Reply](#) :

Thank you.



Vince
@vince1979 1, 2015
I figured out that `align:center` is only for the cross axis. 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.



Coolcat007

Something that approaches what you try to do is this:

```
div {
  float: left;
  width: 200px;
  height: 200px;
  margin-right: 10px;
}
```



Coolcat007

If you use `space-between`, it also seems to align left



Why I gave up as a webdesigner

Yes, let's make CSS even more complicated! We crams more and more stuff into HTML and CSS, but forgets that people wants to settle and work with it and not study new tags/definitions each day.

Reply :



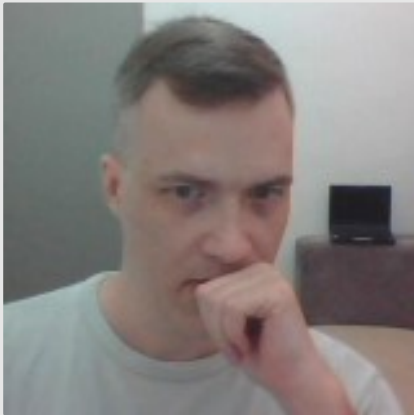
Coolcat007

December 6, 2007

The proposed changes to CSS were initiated years ago, along with the introduction of HTML5. 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. Therefore these new tags were added to simplify web structure/layout, rather than to complicate it.

Take for instance flexbox. 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.

And as a matter of fact, you are still free to use CSS2 and HTML4 if you wish. Nobody is stopping you, but you deny yourself some awesome tools if you do



Vince

The reason that I enjoy working with the web is that it's always growing. There's always something more to learn.

The same is true for any technology or even life in general, really. Without new features and new capabilities, we atrophy and fail to realize our full potential.

I suspect that relatively few people want to settle for what we have now and just work with that.

The W3C isn't a single person who has neglected you, or any of us. 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.

It should probably be noted that the W3C documents recommendations, not requirements. Everything's optional.

You may have been moved by PPK's article [Stop pushing the web forward](#)

I found this counterpoint by Bruce Lawson enlightening: [On PPK's incantation on new browser features](#)

Both of those articles, and more linked to at the end of Bruce Lawson's article, were written by people much smarter than me.

I had to say it, but the frustration expressed by PPK and many others strikes me as very similar to my daughter's frustration with going to school. After all, she already has TV, YouTube, and all the toys she needs at home :)





JANUARY 12, 2016

Very nice/helpful site. Tiny error on <https://css-tricks.com/snippets/css/a-guide-to-flexbox>.

Under "justify-content", bullet item "flex-end: items are packed toward to end line" does not make sense, must be typo.

Just fyi, no reply needed.

Reply:



JANUARY 12, 2016

in the first example there is missing the non prefixed `flex-flow: row wrap;` so right now it's only working in chrome

Reply:



JANUARY 18, 2014

I much preferred the old layout for this article. Seeing the parent and child examples side by side meant it was easier to compare behaviours and to pick the right approach.

Summary



Are you a developer? Try out the [HTML to PDF API](#)



Louisa

#1050497, 2, 2020

Hello,

I want to put a link on images with are in a container flexbox, before putting the link, flexbox work, but after putting the link, it doesn't, WHY? I suppose it's a problem with my html? Can somebody can give me an example about how to do?
Thank you so much!

Here is my html code

```
<div id="pagecontainer">

  <a href="lorettastrong.htm"> </a>
  <a href="page2.html">  </a>
  <a href="page2.html">  </a>

  <a href="page2.html">  </a>

  <a href="page2.html">  </a>

  <a href="page2.html">  </a>

</div>
```



Vince

#1050498, 2, 2020

@Louisa You didn't include your CSS code, so it's impossible to tell what's wrong. I put your HTML into a Pen and set `pagecontainer { display: flex; }` and it works fine.

If you want help, you need to post your CSS code as well. Better yet, post your question and all related code to a site like [Stack Overflow](#) that's designed for questions and answers

Here's my example. I replaced the images with images from [unsplash](#) just to give me something to look at, ...





Stephen
#1 (2016-07-12, 2:03)
According to canvas.com, flexbox is supported in iOS6.* via the webkit prefix:
<http://caniuse.com/feat/flexbox>
According to cani-6ids, iOS support for flexbox is 7.0.1+ -.
I just got a defect ticket for iOS7 where flex doesn't work. So, is the above table wrong?

[Reply](#) :



Proov
#1 (2016-07-12, 2:03)
This guide is my flex bible ! I use it almost once a day ! Many thanks to you Chris !
Anyone know if there is a printable version ? a kind of cheat sheet ?
Thanks :)

[Reply](#) :



Savita
#1 (2016-07-12, 2:03)
Hi , I need to align all elements inside flex container to each other. 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 other, is that possible?

[Reply](#) :





Darren
27/03/2017 04:30:00
This is an excellent guide and I pretty much learned how to layout a page in about an hour using this. I cannot wait to test it out more and see how it all works in different scenarios. Thank you Chris & Team!

[Reply](#)



Joefel
28/03/2017 04:30:00
I am trying to replace a grid layout where I used display: table and table-cell to align content vertically with flexbox. My problem with flexbox is, that I can not get a second child item to align vertically. You can see this in action here: <http://codepen.io/joefel/pen/0gDzwe>. 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!

[Reply](#)



Alexandar
04/03/2017 03:00:00
Who ever wrote this article forgot to put information that flex shrink if put to 0 prevents them to shrink and maintain its original size. This information could have saved me 4 hours of work.

[Reply](#)





Taswell
10000 10, 2018
My boss says Facebook is stupid. She said "shee-lax" or something is better can u confirm??

Reply :



Regis Philbert
10000 10, 2018
Your boss seems pretty talkative when attempting to balance the effectiveness of Facebook and made up a word/service to better enhance that fascinating critique.

Reply :



Jim
10000 10, 2018
Sometimes I smile when reading these articles (and this one is just fine BTW) but I remember back to the dark ages when one could code a fairly decent web page on a single sheet of paper whereas now, it takes endless articles to even understand the coding and then one ends up with megabytes of code ... and it's still just one page but a lot "prettier" (and requires up to 10000 the bandwidth and server storage LOL). And we call it progress.

Reply :





Jim
2 1 month ago
P.S. It reminds me of when C++ came out, and "Hello World" went from 4 or 5 lines of code (C) to 4 or 5 pages of code (C++). LOL I had to swap my PC for a PC-XT to get a hard drive. :_:"



Li
1 1 month ago
If it takes you "megabytes" of code to make a page, you're either doing something very wrong or you've got much more than "just a page", such as a very complex system of scripts or similar.



Syed Asad Abbas
1 1 month ago
Can We make fixed navigation while creating layout of our navigation with flexbox

Reply :





Chris

4 years, 10 days

Regarding the first property

Saying that the 2nd and 3rd parameters `<F2Bx:10F310x>` and `<F2Bx:30B310x>` are optional is slightly misleading because it implies that `<F2Bx:0F000>` (the 1st parameter) is never optional. The truth is, `<F2Bx:0F000>` is optional as long as `<F2Bx:30B310x>` is present (and obviously when the value is none). It's only required when `<F2Bx:10F310x>` is present.

Reply :



Chris Simeone

4

This is an awesome post. 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? It's in the middle below 100px and you'll see what happens.

4 years, 10 days

2

Lorem ipsum dolor sit amet consectetur adipiscing elit.

3

Lorem ipsum sit amet consectetur adipiscing elit. Doloremque, nunc, Nuncit.

Thanks,

Chris

Reply :



Chris Simeone

4

Maybe I need to vertically center the icon to the text instead of the other way around?

4



Henry van Megen
4 years ago
Thanks for the awesome writeup!

Reply :



Martin Karluki
4 years ago
Awesome!

Reply :



Zjaan
4 years ago
I just started to learn HTML & CSS. Thank you for the information you have put together. I wish there was more "Complete Guides" like this out there. This is just brilliant.

Reply :



Jaime
2017-11-20 10:00
And keep coming back to this site. Thanks so much.

Reply :
2017-11-20 10:00
And keep coming back to this site. Thanks so much.



Jannes!
2017-11-20 10:00
Just wanted to say thanks! This is my go to site whenever using flexbox.

Reply :



Ole
2017-11-20 10:00
One of the examples (Numbered Tomato boxes that wrap) uses webkit flex flow, instead of just flex flow, so the example becomes specific to webkit only.

Reply :



Otto Nascarella
2 years ago
Hi people,
This article has been my "chest ache" for the low standard.
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!
John link that shows bug: <https://john.corn/browsers/1>
bugzilla link: https://bugzilla.mozilla.org/show_bug.cgi?id=984889

Reply :
0
0



Otto Nascarella
I meant to say "a bug on Firefox that does not allow elements to be flex containers."



JULY 5, 2016

for com

Regards,

Results

We highly encourage you:

Markdown is supported, so you can write inline code like `<div>this</div>` or *multiline* blocks of code in in triple backtick fences like this:

[Submit Comment](#)