

## Flexbox Offers Responsiveness

Flexbox is CSS designed to provide smooth functionality for arranging containers and items on a web page that are responsive to changes in screen size. Its structure is also simple to manipulate in `@media` declarations. It is comprised of two basic components. Flexbox containers, and items which go into a container. While understanding these concepts is of utmost important, and understanding container was no problem, getting information on “item” was a bit elusive. As I understand it now, an item is simply any element, whether an element used as a container (e.g. `div` or `section`) or an element used as a page component (e.g. `h1` or `p`).

The basic declaration for flexbox is either `display: flex;` or `display: inline-flex;` The difference is comparable to `block` vs `inline-block` declaration ([Stackoverflow](#)).



Image 1

Flexbox provides easy ways to align items within a container. For instance, to have items pile up on top of each other, use the `flex-direction: column;` property/value in the container declaration. To display items side by side simply use `flex-direction: row;` Additional values offer greater flexibility in handling order. There is also a `flexbox-wrap` property that instructs the container to let its items wrap or not wrap. This is one of the things that makes it simple to adjust `@media` settings. For instance, set `nowrap` for a desktop and `wrap` for a mobile can flow left to right (or right to left if you like) or stack content nicely in the display adjusting for each media.

The “Complete Guide to Flexbox” ([CSS-Tricks](#)) and “Basic concepts of flexbox” ([MDN web docs](#)) are both great references if you are interested in developing a flexbox web page. The latter does a better job in explaining the css declarations for both containers and items, and has a nice menu for selecting specific properties. The former is a good deep read and offers quicker look-up once you get the hang of it.

My CSS skills are marginal, and it took me a while to get up to speed. After 15-20 hours working with it, I was getting pretty good results. I highly recommend for anyone interested in responsive web to take a look.

## Works Cited

MDN web docs: “Basic concepts of flexbox”,  
[https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Flexible\\_Box\\_Layout/Basic\\_Concepts\\_of\\_Flexbox](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox); not dated; accessed 24 February 2019.

CSS-Tricks: “A Complete Guide to Flexbox”  
<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>; updated 31 January 2019; accessed 20 February 2019.

Stack Overflow: “What’s the difference between display:inline-flex and display:flex”,  
<https://stackoverflow.com/questions/27418104/whats-the-difference-between-displayinline-flex-and-displayflex> ; posting dated 21 March 2018; accessed 24 February 2019.

## Illustrations

Image 1: CSS-Tricks: “A Complete Guide to Flexbox”  
<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>; updated 31 January 2019; accessed 20 February 2019.

## Internet Searches

Here’s the internet searches I did and the articles I accessed while researching this paper. I keep these largely for my own reverence, and offer them to you in the hopes it will make your life easier! :)

- [how do I use css flexbox](#)
  - [CSS Flexbox \(Flexible Box\) - W3Schools](#)
  - [A Complete Guide to Flexbox | CSS-Tricks](#)
- [browser flexbox support](#)
  - [Backwards Compatibility of Flexbox - CSS: Cascading Style Sheets ...](#)
- [browser flexbox shim](#)
  - [Normalizing Cross-browser Flexbox Bugs — Philip Walton](#)
  - [Using Flexbox: Mixing Old and New for the Best Browser Support ...](#)
- [what is a flexbox item](#)
  - [Basic concepts of flexbox](#)
- [flex vs inline](#)
  - [What's the difference between display:inline-flex and display:flex ...](#)