



CSS Grid or Flexbox

Compare and Combine

Aaron LuBean (alubean@ChurchOfJesusChrist.org)

Mary Ebert (tikvahh@ChurchOfJesusChrist.org)

FamilySearch Web Developers

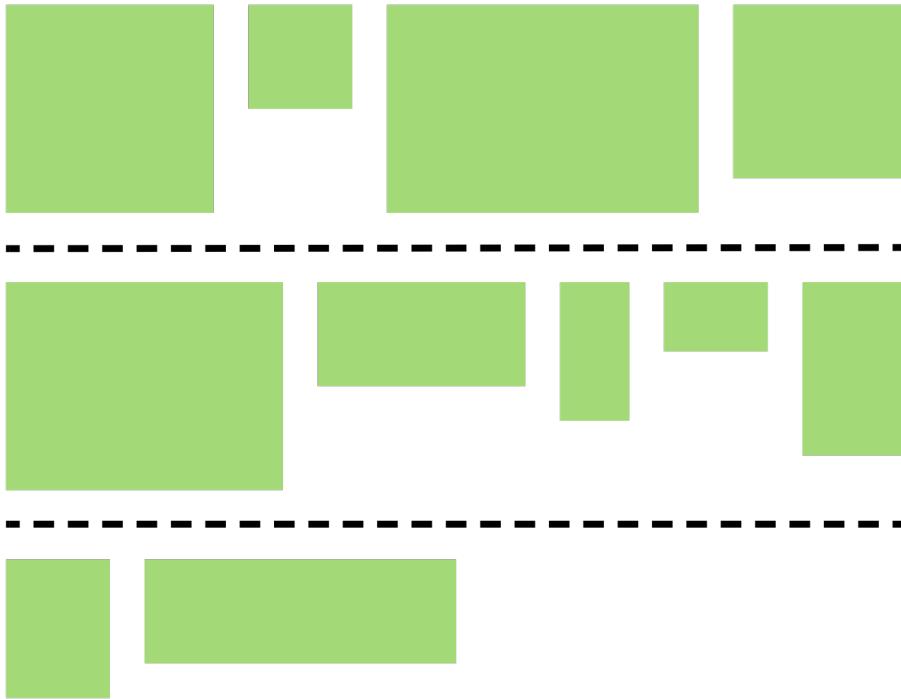
October 8, 2019

Flexbox or Grid for layout?

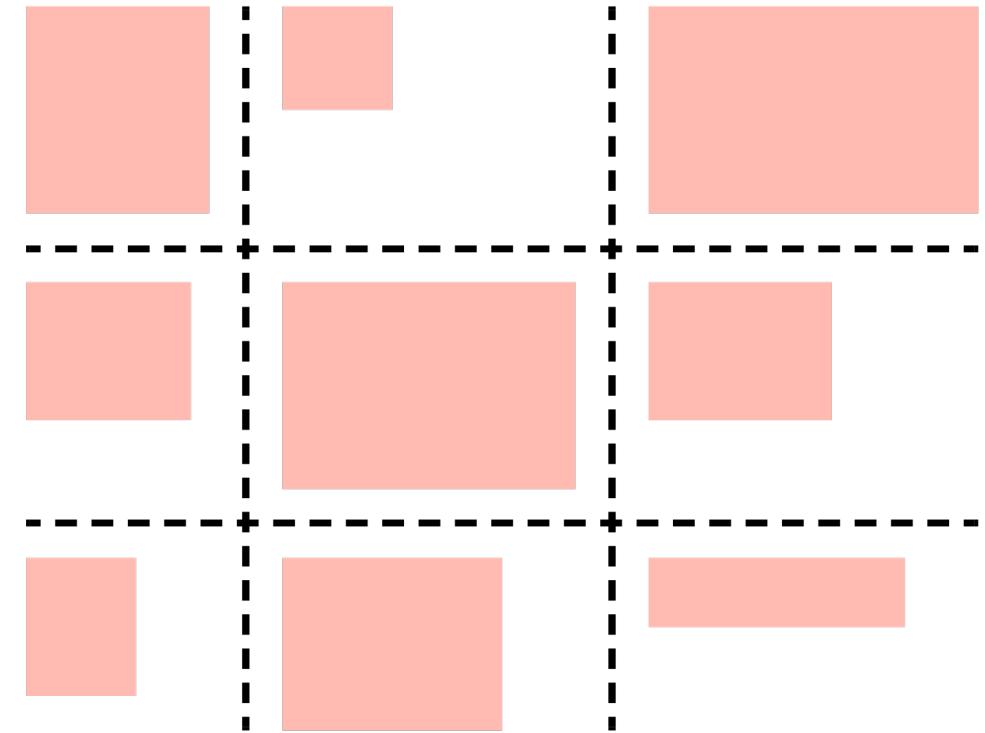
- Flexbox
 - one-dimensional, content-first approach
- Grid
 - two-dimensional, layout-first approach

Flexbox vs. Grid

Flexbox lines 'em up in one direction

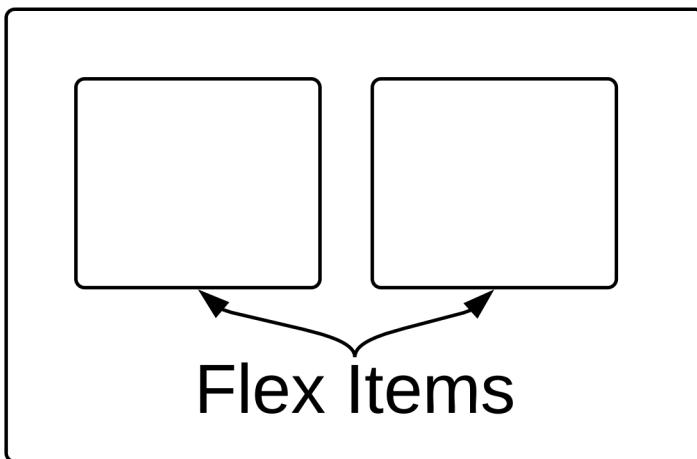


Grid lines 'em up in two directions



Flexbox Anatomy & Terminology

Flex Container



■ Container – `display: flex;`

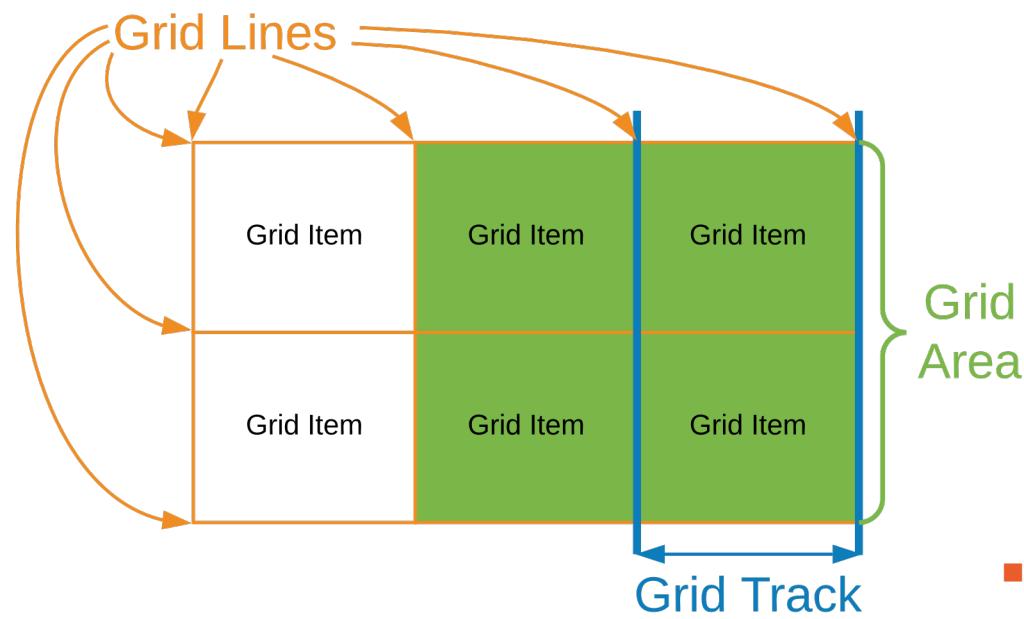
- `flex-direction: row | row-reverse | column | column-reverse`
- `flex-wrap: nowrap | wrap | wrap-reverse`
- `justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly`
- `align-items: stretch | flex-start | flex-end | center | baseline`
- `align-content: stretch | flex-start | flex-end | center | space-between | space-around`

■ Flex Item – direct child of container

- `order: 0 | <integer>`
- `flex: 0 1 auto | <grow> <shrink> <basis> | none`
- `align-self: auto | flex-start | flex-end | center | baseline | stretch`

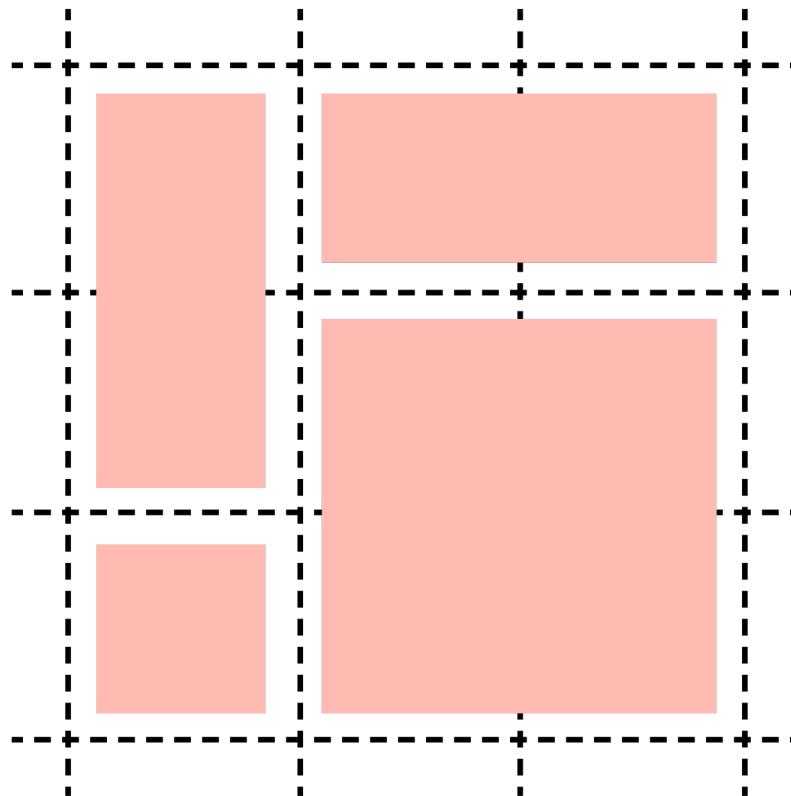
Grid Anatomy & Terminology

Grid Container



- Container – `display: grid;`
 - `grid-template-columns: <track-size> ... | [line-name] <track-size> ...`
 - `grid-template-rows: <track-size> ... | [line-name] <track-size> ...`
 - `grid-template-areas: "<grid-area-name> | . | none | ..." ...`
 - `grid-auto-columns: <track-size> ...`
 - `grid-auto-rows: <track-size> ...`
 - `grid-auto-flow: row | column | row dense | column dense`
 - `grid-gap: <grid-column-gap> <grid-row-gap>`
 - `justify-items: start | end | center | stretch`
 - `align-items: start | end | center | stretch`
 - `justify-content: start | end | center | stretch | space-around | space-between | space-evenly`
 - `align-content: start | end | center | stretch | space-around | space-between | space-evenly`
- Grid Item – direct child of container
 - `grid-column: <start-line> / <end-line> | <start-line> / span <value>`
 - `grid-row: <start-line> / <end-line> | <start-line> / span <value>`
 - `grid-area: <grid-area-name> | <row-start>/<column-start>/<row-end>/<column-end>`
 - `justify-self: start | end | center | stretch`
 - `align-self: start | end | center | stretch`

Column/row span & layering in Grid



Grid Special Functions and Keywords

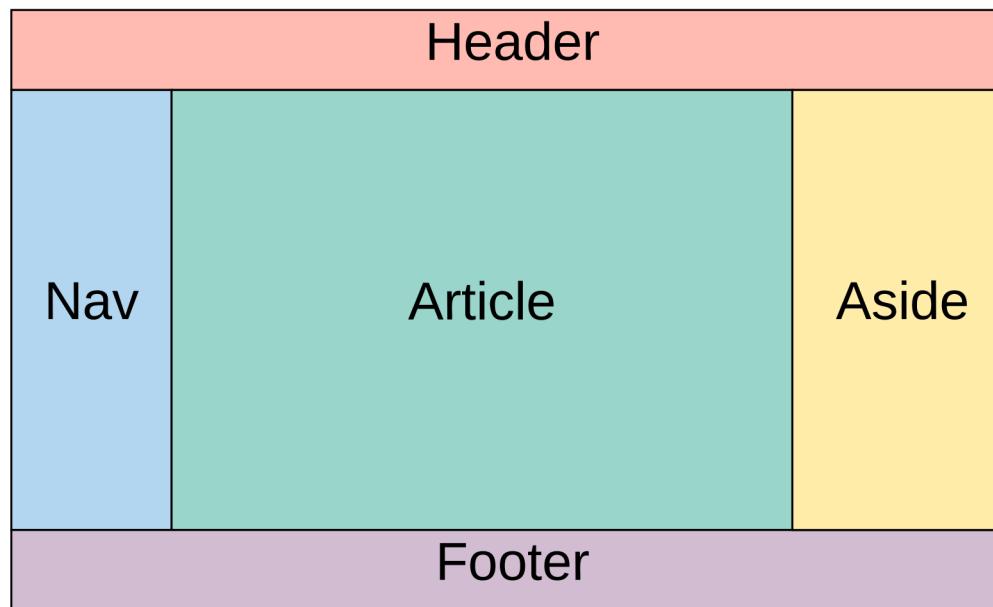
- `fr` length unit
- `min-content`, `max-content`, `auto`
- `minmax()`
- `repeat()` function w/ `auto-fill` & `auto-fit`

Key Differences

- Approach
 - Flexbox: one-dimensional, content-first approach
 - Grid: two-dimensional, layout-first approach
 - Explicit Layouts Easier
 - More Concise Media Queries
- Layering & Spanning
 - works in Grid, hacked in Flexbox
- Grid Has Gap
- CSS Animations – better support in Flexbox

Combining Grid and Flexbox

- Can Grid and Flexbox be used together?
- Let's do a combined Holy Grail layout



Code & References

- Slides and examples: <https://github.com/aaronlubean/grid-v-flexbox>
- Wrapping in Flexbox vs. Grid: <https://codepen.io/aaronlubean/pen/WNeWXrr>
- Flexbox Playground: <http://the-echoplex.net/flexyboxes>
- Holy Grail – Flexbox: <https://codepen.io/aaronlubean/pen/LYPvWWV>
- Grid Playground: <https://gridbyexample.com>
- Holy Grail – Grid: <https://codepen.io/aaronlubean/pen/ZEzwwEW>
- Track Spanning in Grid: <https://codepen.io/rachelandrew/pen/oXKgwa>
- Layering in Grid: <https://codepen.io/rachelandrew/pen/KpOwQW>
- Responsive Gallery w/o media queries - Grid: <https://codepen.io/aaronlubean/pen/vYBqMaE>
- Animating grid-gap: <https://codepen.io/matuzo/pen/rmQvMG>
- Animating grid-template-rows/columns (Firefox Only): <https://codepen.io/michellebarker/pen/oJmZKK>
- Holy Grail – Grid & Flexbox: <https://codepen.io/aaronlubean/pen/MWgLxbQ>
- <https://css-tricks.com/snippets/css/a-guide-to-flexbox>
- <https://css-tricks.com/snippets/css/complete-guide-grid>
- <https://cssgrid.io>



Thanks & Questions



SORT