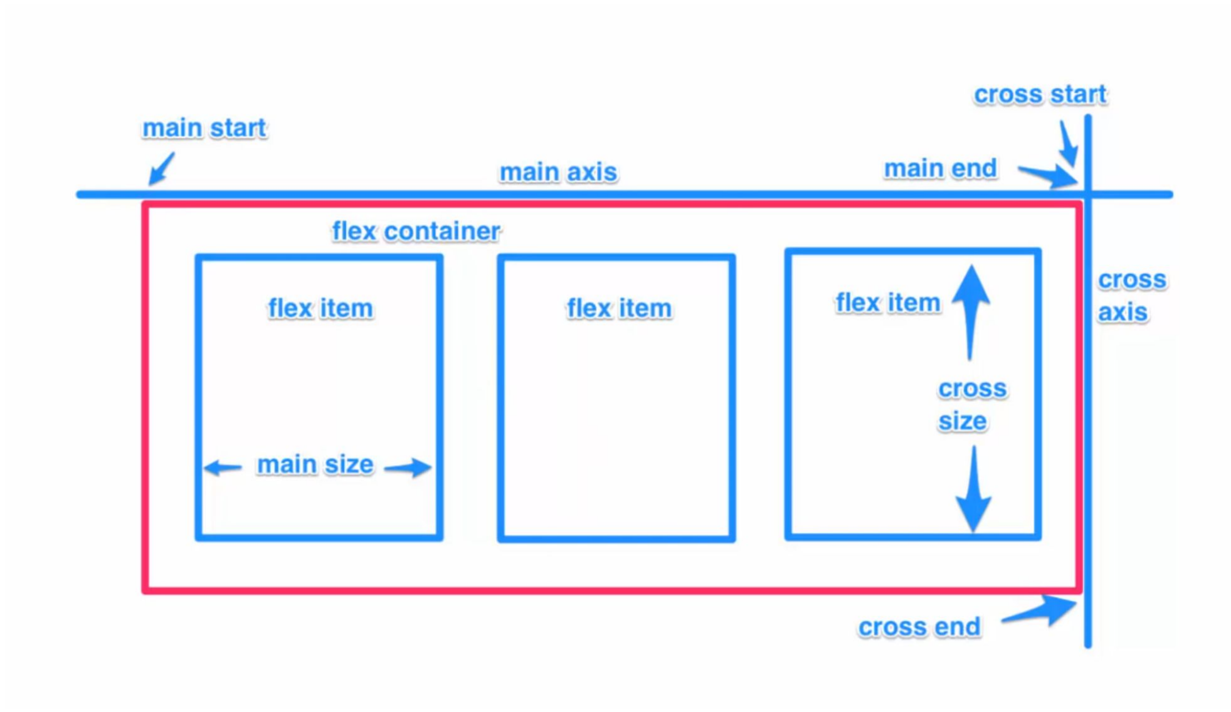
Layout: Flex and Grids

Hunter Jorgensen

How to effectively layout with modern CSS

- Last few years, flex (or flexbox) and grid have been released and are effective ways of laying out a website
- Previously developers needed to use float, calculate margins/width/height/etc
- Flex and grid allow for easy responsive sites
- They can be nested with each other and within themselves

Flex


cross start
main start
main axis
main end
flex container
flex item
flex item
flex item
cross axis
cross size
main size
cross end

Flex Attributes and Values

Container Properties

- display: flex/inline-flex
- flex-direction: row/column;
- flex-wrap: wrap, wrap-reverse
- flex-basis: {number} (basically width of items)
- justify-content: flex-start, flex-end, center, space-between, space-around
- align-self: flex-start, flex-end, center
- align-items: flex-start, flex-end, center
- align-content: flex-start, flex-end, center

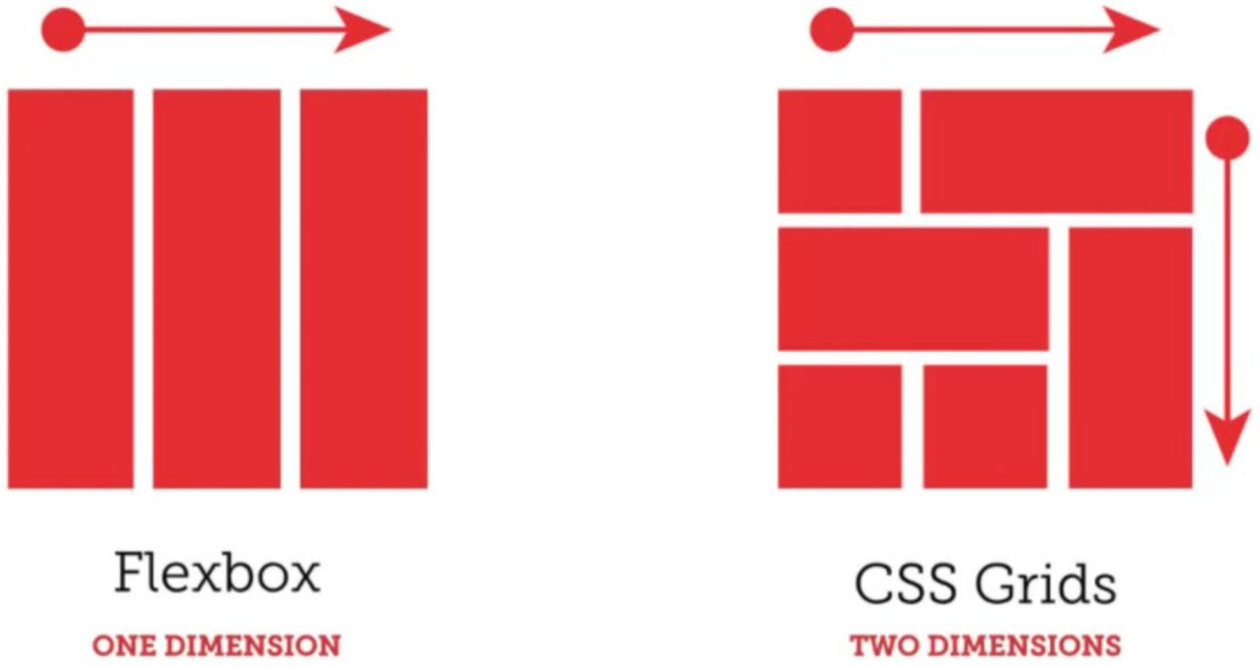
Item Properties

- flex-grow: <number>
- flex-shrink: <number>
- flex: <number> <number> <number>
- order: <number>

Flex Demo

Demo Link

Grid



Grid Attributes and Values

Container Properties

- display: grid
- grid-template-columns: {list of integs}
- grid-template-rows: {list of integs}
- grid-column-gap: {number}
- grid-row-gap: {number}
- grid-gap: {number}
- grid-template-areas: list of strings
- grid-auto-rows: number
- grid-auto-columns: number

Item Properties

- grid-column: number range
- grid-row: number range
- grid-area: string

Can also do align/justify similar to flex

Grid Demo

Demo Link