



CTD Intro Week 11

CSS Layout



- **Relative units are preferred for responsiveness**
 - **em**: Relative to the font-size of the element
 - 2em means 2 times the size of the current font
 - **rem**: Relative to the font-size of the root element
 - **ch**: Relative to the width of the “0” (zero)
 - **vw**: Relative to 1% of the width of the viewport
 - **vh**: Relative to 1% of the height of the viewport
 - **vmin**: Relative to 1% of the viewport’s smaller dimension
 - **vmax**: Relative to 1% of the viewport’s larger dimension
 - **%**: Relative to the parent element
 - **vi/dvi**: Relative to 1% of the inline dimension of the viewport. Adding ‘d’ makes it dynamic.
 - **vb/dvb**: Relative to 1% of the block dimension of the viewport. Adding ‘d’ makes it dynamic.
- **Absolute units (don’t work well with varying screen size)**
 - **cm**: Centimeters
 - **mm**: Millimeters
 - **in**: Inches (1in = 96px = 2.54cm)
 - **px**: Pixels (1px = 1/96th of 1in)
 - **pt**: Points (1pt = 1/72 of 1in)
 - **pc**: Picas (1pc = 12pt)

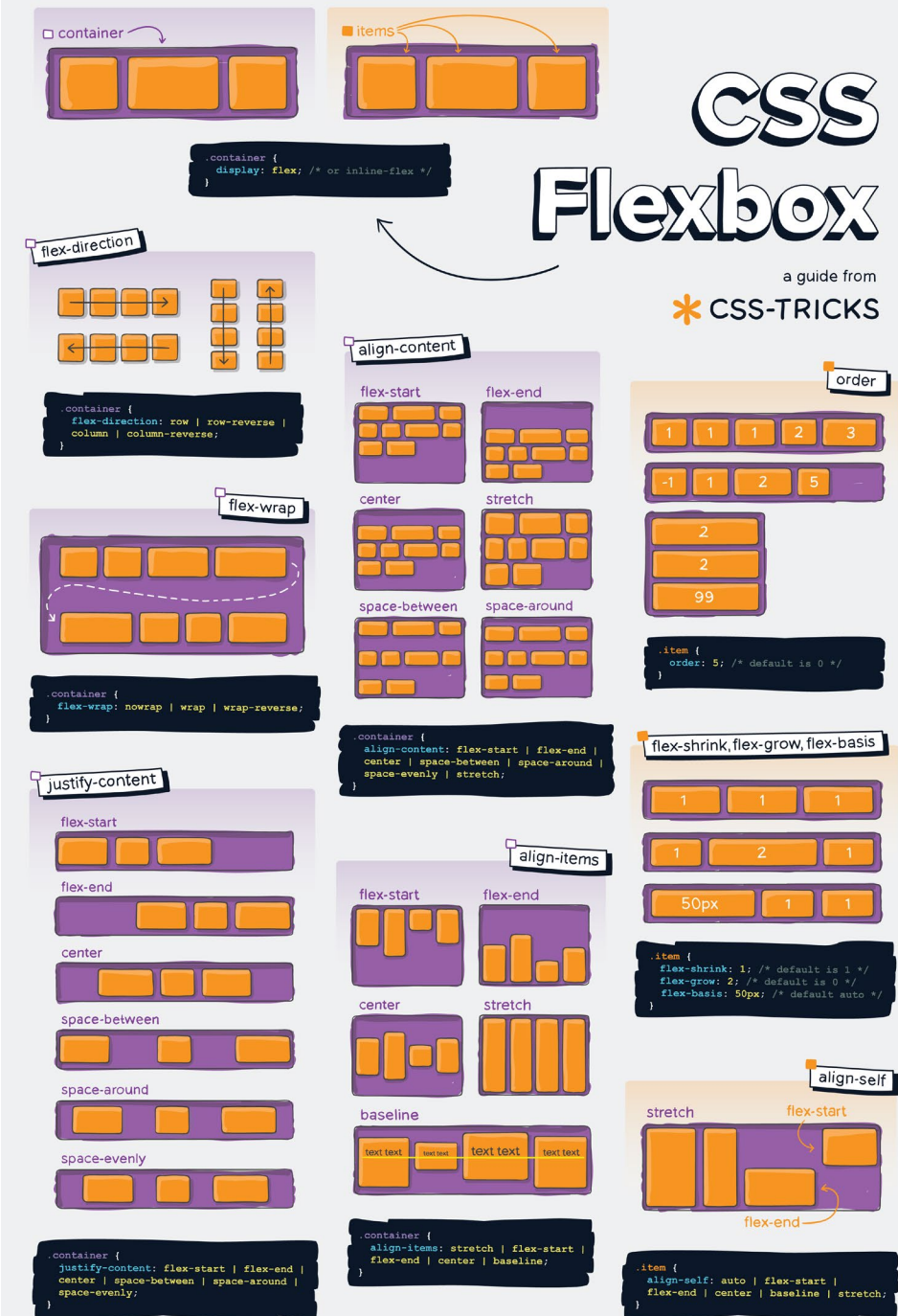
Navigation Menu

- How to link within a page
 - Add an ID to each target
 - Reference the id in the href (e.g. #about)
- Make it sticky
 - Use css position: fixed and top specifications
 - Alternative, position: sticky
- Fixed vs sticky
 - Fixed removes the element from the document flow, fixes it in the viewport
 - Need to deal with it being on top of other content
 - Sticky is relative until a threshold is reached and then becomes fixed in the containing element
 - More flexible, any element rather than the whole viewport
- Make sure all of the target headings show correctly
 - Apply scroll-margin-top to each section if necessary
- Make width 100% (optional design suggestion)
- Opaque background color (optional design suggestion)
 - So scrolled content doesn't show through
 - Match body background
- Pad the bottom of sections to avoid hitting the end of the document with nav links



Flexbox and CSS Grid

- Flexbox vs CSS grid
 - Flexbox is 1D
 - CSS Grid is 2D
- Grid for larger layouts
 - Full pages, landing pages...
- Flexbox for smaller scale layouts
 - e.g. nav menus
 - Specified via `display: flex` (not `flexbox`)
- They can be combined





Media Queries

- Mobile, Browser
- Example:
 - `@media only screen and (min-width: <some-width>px) {
<any styling override> }`
- Testing/debugging media aware layouts
 - Right-click->Inspect
- For next week's lesson



Questions

Previous lessons?