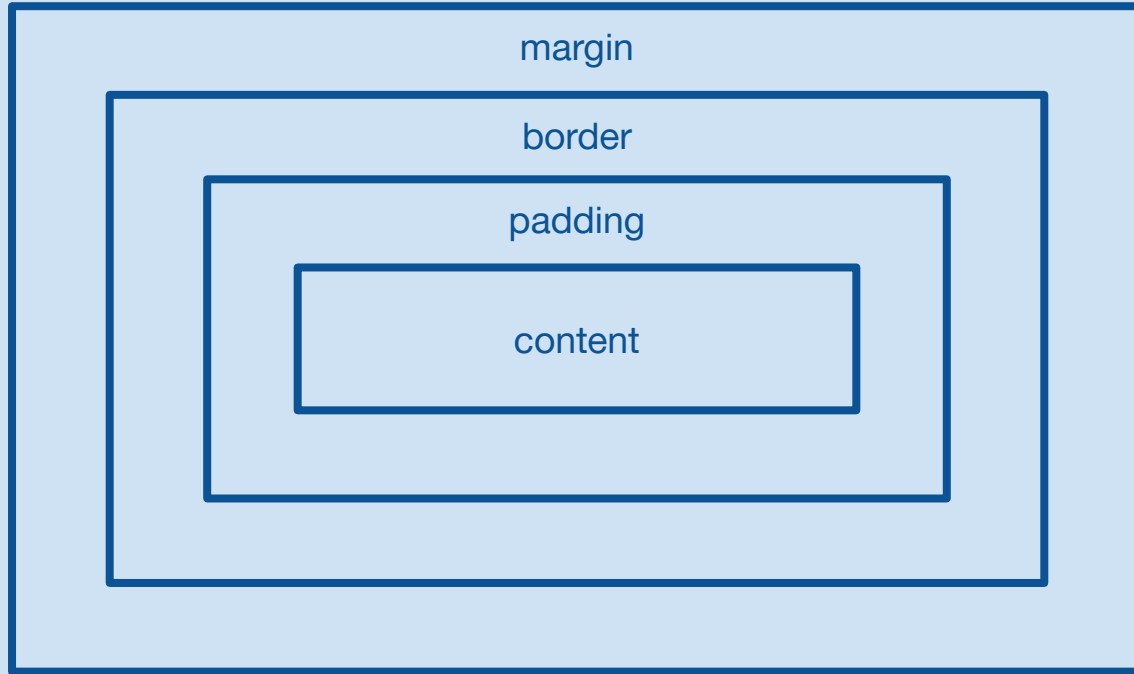


Page Layout - Best Practices

August 30

Today's Goal: break down and simplify CSS so that you know ***what*** you actually want to style and ***how*** you want to style it.

The Box Model



The Box Model: CSS box-sizing Property

Usually helps answer the question of: “Why the hell is my element not the size I expect it to be?”

`box-sizing: content-box;`

- Default style for all box-like elements
- Content size is kept constant
- Box dimensions = content size + padding + border

`box-sizing: border-box;`

- Never applied by default, must always be explicitly applied in CSS
- Overall box size is kept constant
- Content size = box dimensions - padding - border

Flow Direction and Writing Direction

block elements follow the page's flow direction while inline elements follow the page's writing direction.

inline-block elements follow writing direction like inline elements do, but give you more flow direction related control.

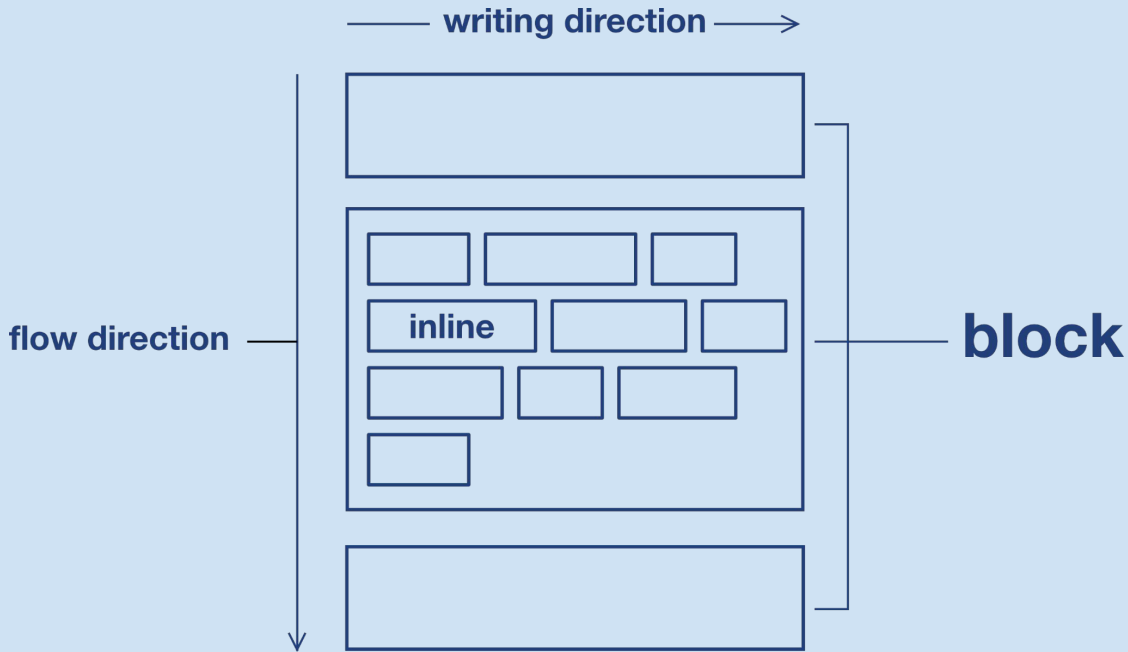


Image credit: every-layout.dev

Composition

DRY - Do Not Repeat Yourself

Recreating every single small piece not only leads to more work for you, but also leads to files that are larger and more complicated than they need to be, which makes your work harder to understand, share, and maintain.

Break down your layouts into as many tiny reusable pieces as you can, so that you create as many things as you can only once.

Responsive Design

- Media queries - generally breakpoints that allow your site to look vastly different (but still nicely designed) on different platforms.
- Flexible designs:
 - rem and/or em instead of px - allows for “hard values” flexible measurements based on the device or container’s base font size.
 - % instead of px - allows for “soft values” flexible measurements based on the device or container’s size.
 - min-width, min-height, max-width, and max-height CSS properties - when used in the right combination can let containers be flexible in their sizing up to and down to certain points. Hint: combine hard values in the min/max-width/height properties with soft values in the regular width/height properties for a nice result.
 - Flexbox, AKA `display: flex` - the king of all flexible containers. We will cover this more next class