

1. NO MARGINS

When you look at a component, you see its natural borders. If something looks like "the space around it", then it's not part of the component. (Shadows and ripple effects are excluded, too.)

It's impossible to automatically rearrange or align components with margins, because even though they are technically aligned, their perceived boundaries aren't aligned.



The boundaries are obvious.



Possibly undesired negative space.

It's just a preview.

Get the rulebook at SpacingSystem.com

When each component has a single responsibility, be it acting as a UI control or providing spacing, it's easier to compose them, because they can be reused.

It aids in finding the right spacing token and makes it possible to globally update the spacing in some context without negatively affecting other contexts.

4. CENTRALIZED

There's one place where all the spacing values are defined. Components don't use any locally defined spacing values, like "0px" or "13px".

A single source of truth regarding spacing values helps both designers and developers find and update spacing tokens.

5. UNDERSTOOD

All the people involved in the design and the implementation of the user interface understand this spacing system.

A system is a set of rules. Rules work if they are followed. To be followed, they must be understood.

A COMPLETE EXAMPLE

HOW TO GET THERE?

WHAT CAN GO WRONG?

HOW TO DO IT IN CSS?

WHAT KINDS OF DESIGNS CAN BENEFIT FROM THIS SYSTEM?

Its focus on consistency makes it ideal for apps that often reuse components, are responsive, and display dynamic data. Configurable dashboards are a good example. One-off components don't call for consistency or a rule-based layout, thus don't require a systematic approach to spacing.

WHAT KINDS OF COMPONENTS CAN I APPLY IT TO?

The same principles apply both to tiny components like labels as well as entire screens.