**CSS Design Patterns**

*Techniques to build beautiful UIs in CSS.*

Do you envy how Software Developers have this book called “Design Patterns” written by the famous “Gang of Four”? These principles now hold for about 30 years, and they are now the foundation of any programmer’s education. I have been developing software for over 20 years and love the frontend just as much as I love writing backend code. But how long have I wished for the frontend to have clear design patterns that don’t depend on the library or framework we are using? It was only some years ago that I realized that we already have that, but it seems to me that nobody collected and categorized the patterns. So, here’s my try to organize and list things that are — in my opinion — current state-of-the-art design patterns. I don’t give examples to any of them in this article; it shouldn’t be difficult to use your favorite search engine to find them. I appreciate any comment from you. What do you think about that? Have you found a similar list? Do you miss something on the list? What do you think about the categories?

Here we go:

**Layout Patterns**

Layout patterns are a collection of CSS properties and values that can help you arrange elements in your document.

* **The “Holy Grail” Layout**  
  A three-column layout with a header and footer.
* **Multi-Column-Grids**Any technique to position DOM elements in a grid. For instance, an image gallery.
* **Aspect Ratio**Display an element with a specific aspect ratio. E.g., padding-hacks or using the newer aspect-ratio property.
* **Centering**  
  Alignment of elements.
* **Clamping**  
  Stopping elements from exceeding a specific dimension.
* **RAM**The abbreviation stands for **r**epeat, **a**uto-fill / **a**uto-fit, and **m**in-max. It is a pattern for arranging elements in a grid.
* **Same Height Columns**  
  Display elements that are next to each other with the same height.
* **Sticky header / Sticky footer**  
  Keep an element within a specific position of the current viewport.

**Component Patterns**

Components are ready-to-use UI structures. When you see a website, you can quickly identify its components, e.g., a navigation bar, cards, or lists.

Any Pattern can be further categorized into Elements (or Atoms), Forms, Components (or Structures or Molecules), Sections, and Overlays.

**Elements**— the basic building blocks

* Arrow
* Avatar
* Badge
* Card
* Corner Ribbon
* Dot Leader
* Drop Area
* Keyboard Shortcut
* Overlay Play Button
* Price Tag
* Progress Bar / Progress Indicator
* Tooltip
* Pagination

**Forms**— Any elements regarding form input

* Button
* Chip
* Custom Checkbox
* Custom Radio Button
* Floating Label
* Input Addon
* Radio Button Group / Option Group
* Switch / Radio Switch
* Rating
* Search Box
* Slider

**Components**— Compound elements

* App Bar
* Breadcrumb
* Circular Navigation
* Cookie Banner
* Drawer
* Folder structure
* Full-Screen Menu
* Lists
* Menus
* Rails (e.g., Navigation Rail)
* Snackbar
* Tables
* Tabs
* Timeline
* Wizard

**Sections**— Bigger Chunks of a Page

* Fading section
* Feature Comparison
* Full Background
* Non-linear Background
* Pricing Section
* Video Background

**Overlays**— Components that overlay the current viewport

* Backdrop
* Banner
* Dialog
* Modal
* Notification
* Popover

**Scoping Patterns**

Patterns dealing with specificity issues, general scoping, and separation of concerns.

* CSS Modules
* Shadow DOM
* BEM
* CSS in JS

We could name so many more, especially in the last category. However, I think that these are the most common ones. Did I forget something? Do you believe some are too specific and could be solved in combination with other patterns I mentioned? I appreciate any comment!

That’s all, folks; thank you for reading!

Here are some resources that I used to create the lists:

* <https://tailwindui.com/components>
* <https://material.io/>
* <https://csslayout.io/>

20