

Do we really need a design system?

which is made from scratch by us?

The problem?

The problem?



Wireframe

The problem?



Wireframe



Mockup

The problem?



Wireframe



Mockup



Final Product

The problem?

- Design consistency
- Design and development alignment
- Tracking designers changes
- Accessibility (a11y) & Testability
- DX
- ...



Wireframe



Mockup



Final Product

I'm
Amirreza Safehian
Frontend developer at Riverlane



JavaScript and NodeJS Cambridge
November event

Table of Contents

- Definition of a design system
- Good to know principles
 - Atomic design
 - Design tokens
- Build one
 - To reinvent the wheel
 - Not to reinvent the wheel
- Tools and examples
- Live coding (hopefully)
- Key Takeaways
- QnA

Definition of a design system

Definition of a design system

A design system is a collection of **reusable components** that can be used to build web applications and websites. It helps teams create consistent, high-quality user interfaces **quickly** by providing reusable components for developers to use in their projects. It also makes it easier for designers to **collaborate** with developers on creating new features because they have clear documentation about how each component should behave and look.

What is the Definition of a design system in the context of web development?

✓ Searching for: **definition of design system in web development**
✓ Generating answers for you...

A design system is a collection of reusable components that can be used to build web applications and websites. It helps teams create consistent, high-quality user interfaces quickly by providing reusable components for developers to use in their projects. It also makes it easier for designers to collaborate with developers on creating new features because they have clear documentation about how each component should behave and look ¹.

In other words, a design system is a visually well-structured compilation of logically consistent reusable elements, such as branding, rules, and style and usage guidelines. It is used to implement and organize an effective design interface and experience ².

Design systems are widely used in web development workflows. Some popular design systems include Google's Material Design, Apple's Human Interface Guidelines, Microsoft's Fluent Design System, and Salesforce Lightning Design System ¹.

I hope this helps! Let me know if you have any other questions.

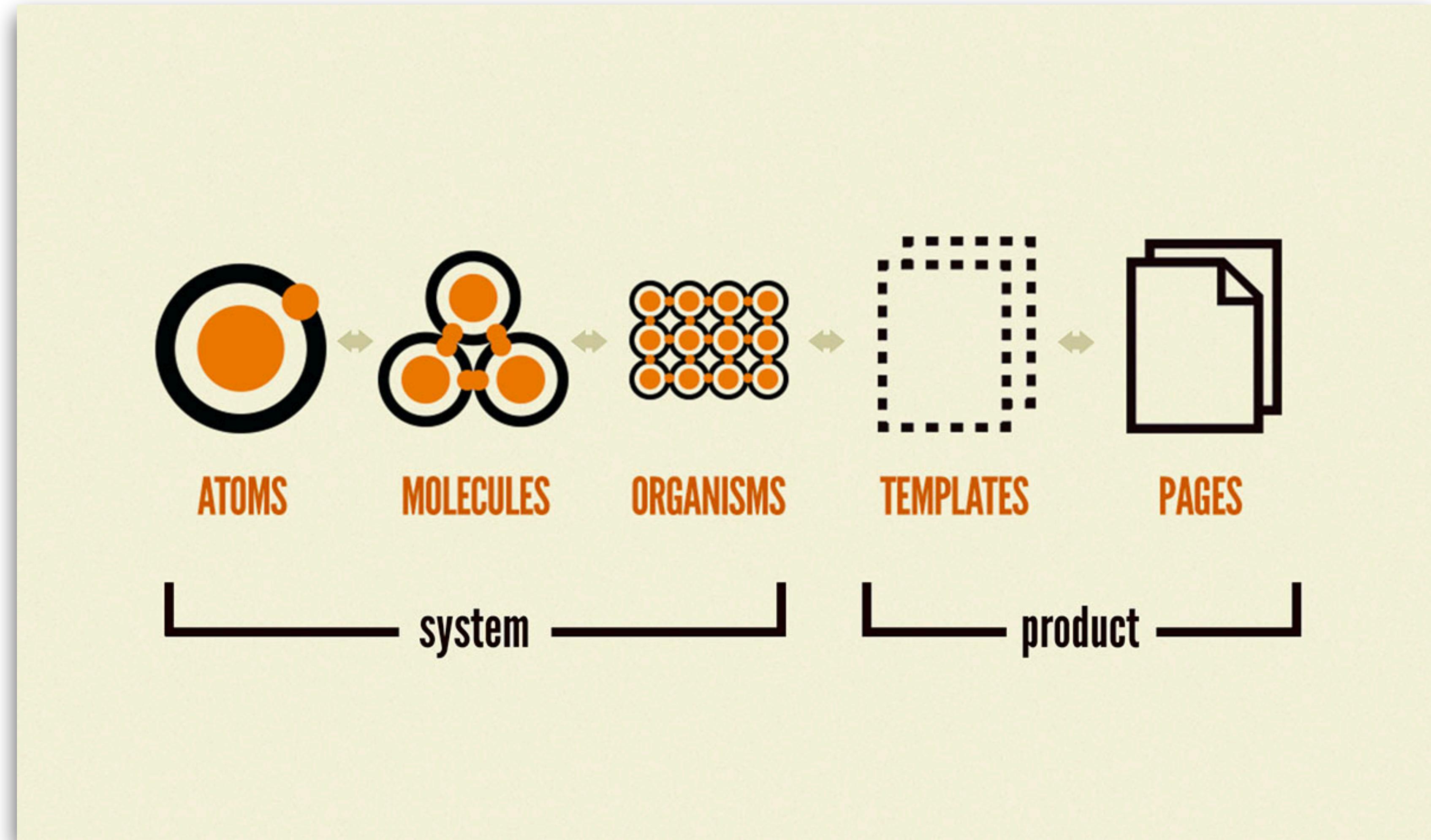
Learn more: [1. bootcamp.uxdesign.cc](https://bootcamp.uxdesign.cc) [2. designrush.com](https://designrush.com) [3. blog.hubspot.com](https://blog.hubspot.com) [+1 more](#)

1 of 5

Good to know principles

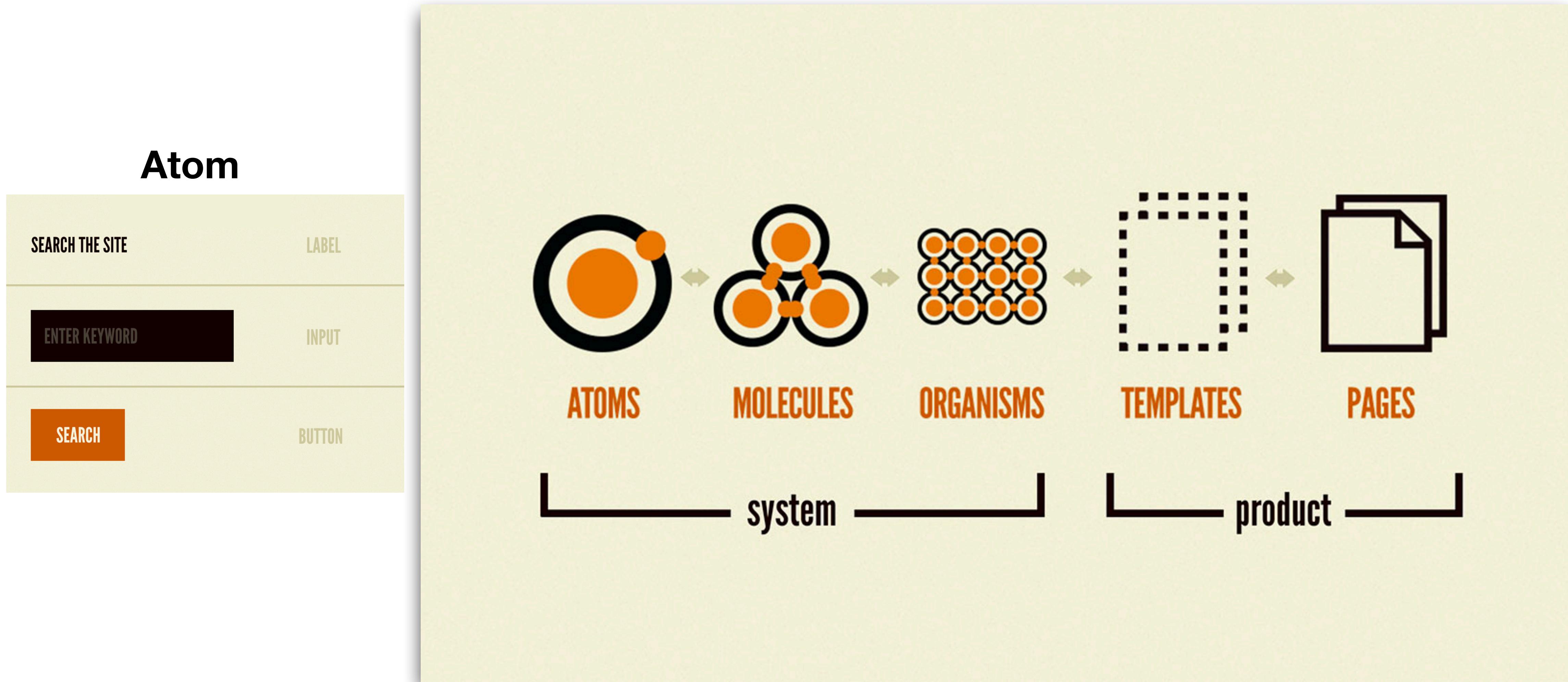
Good to know principles

Atomic Design



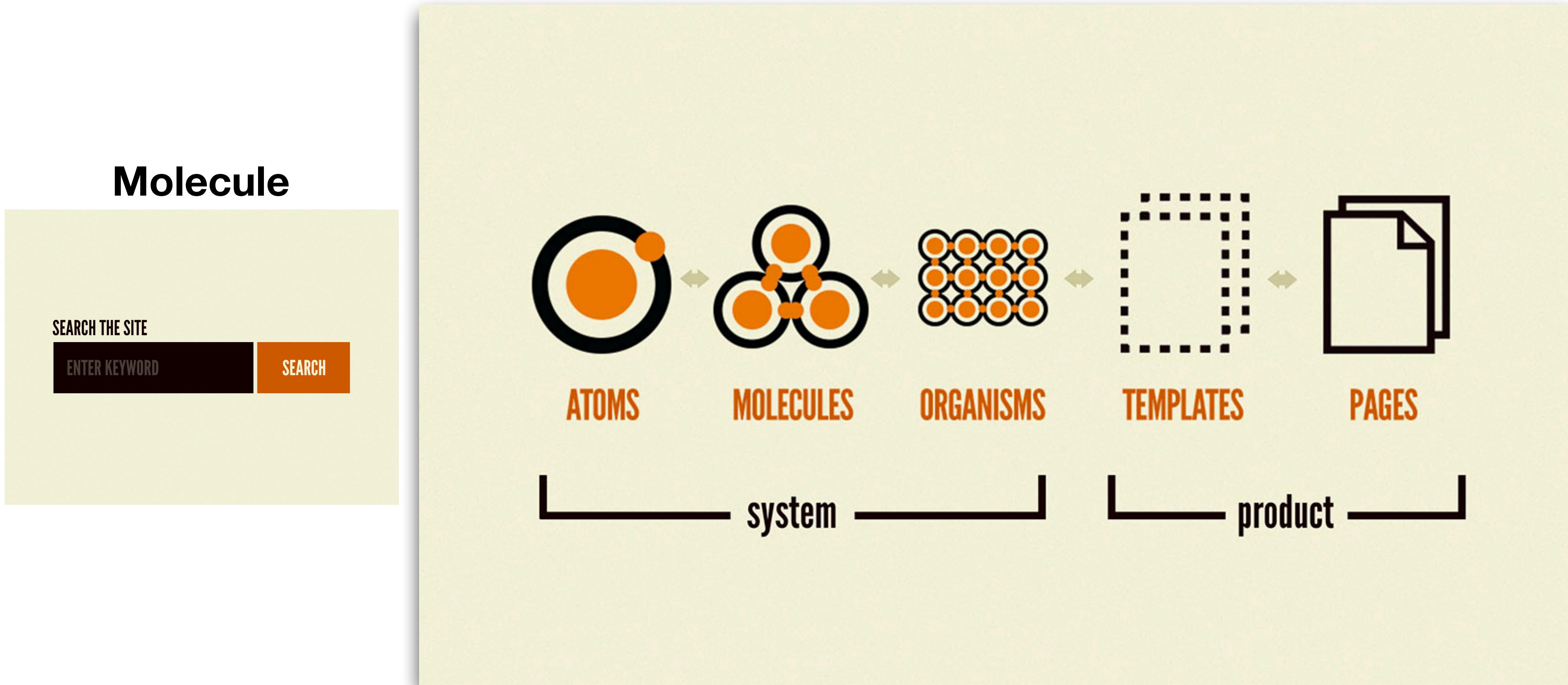
Good to know principles

Atomic Design



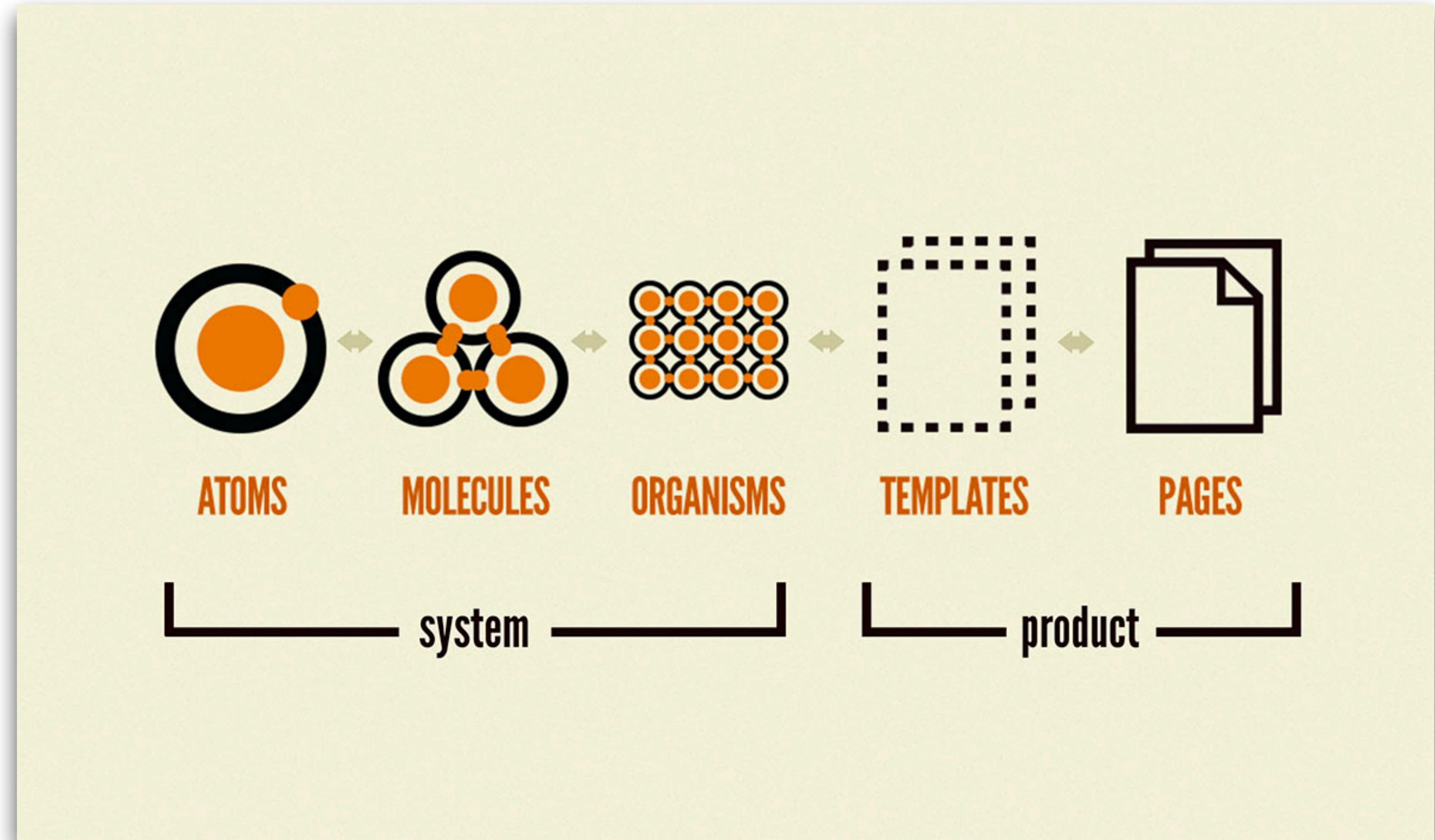
Good to know principles

Atomic Design



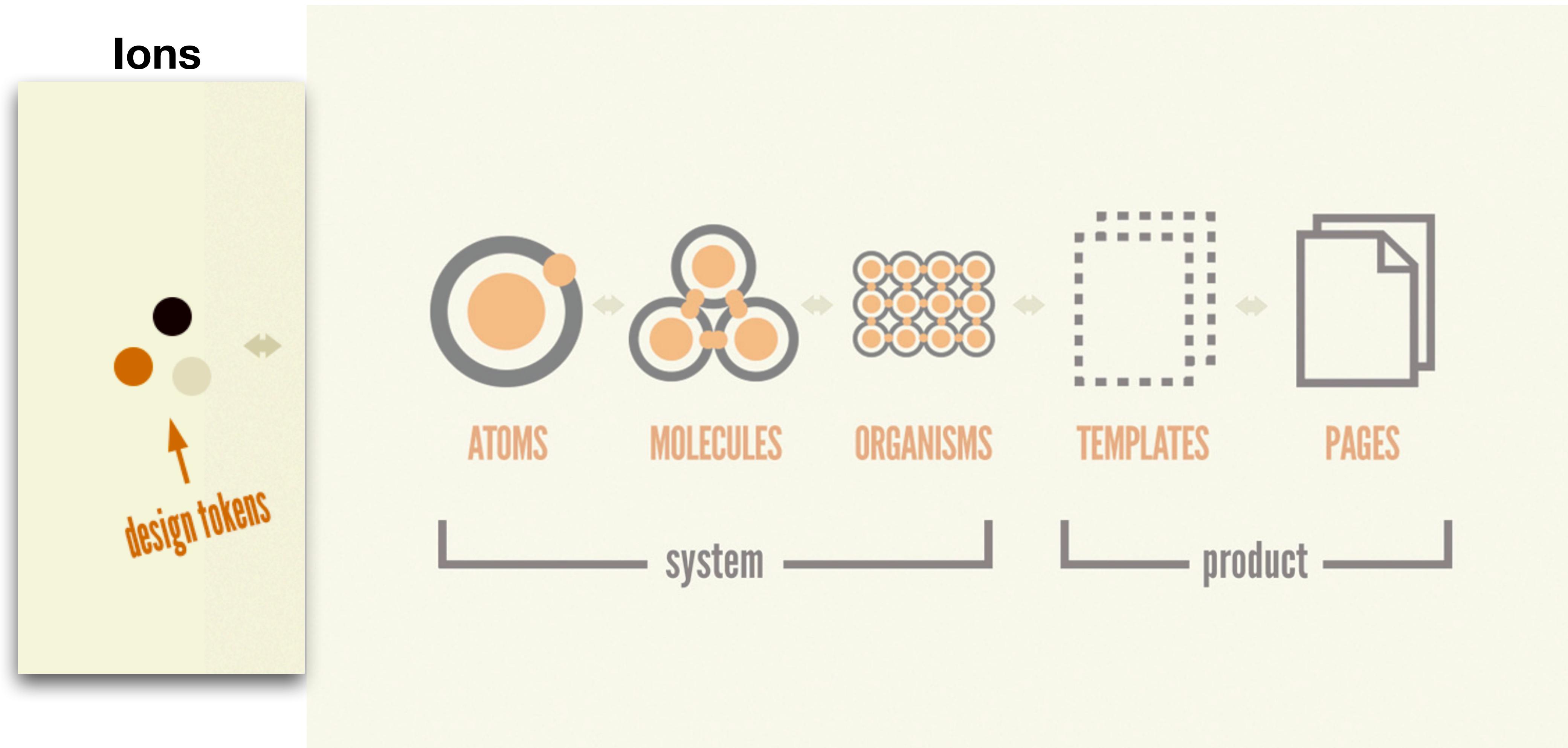
Good to know principles

Atomic Design



Good to know principles

Design Tokens



Good to know principles

Design Tokens

Design tokens represent the small, repeated design decisions that make up a design system's visual style

Good to know principles

Design Tokens

Design tokens consist of values needed to construct and maintain a design system, such as spacing, colour, typography, object styles, animation, and more.

Good to know principles

Design Tokens

Jina Anne [@jina · Follow](#)

Design Tokens are a methodology. IMHO, saying “design tokens are just variables” is like saying “responsive design is just media queries”. It’s a technology-agnostic architecture and process for scaling design across multiple platforms and devices, including native, and more.

8:42 PM · Nov 14, 2018

163 Reply Copy link

[Read 8 replies](#)

Good to know principles

Design Tokens

Context

*The most **broad** idea.*

- A component
- A system constant (e.g., a border, a background)
- A color

Common unit

*The most **consistent** idea.*

- Sizing
- Spacing
- Styling
- Text

Clarification

*The most **specific or detailed** idea.*

- T-shirt size
- Index/value
- User-initiated state

checkbox

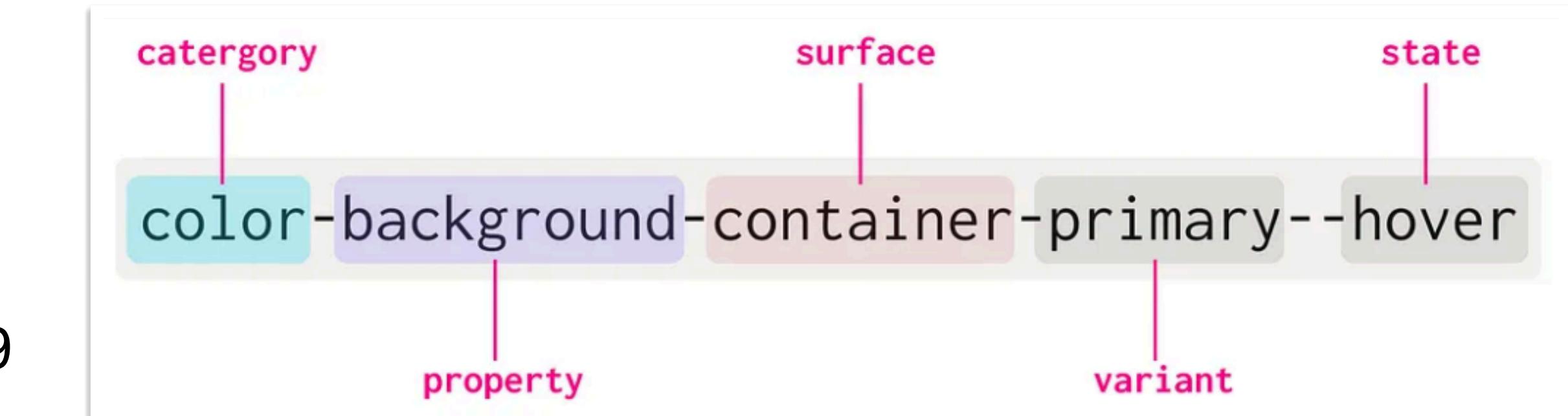
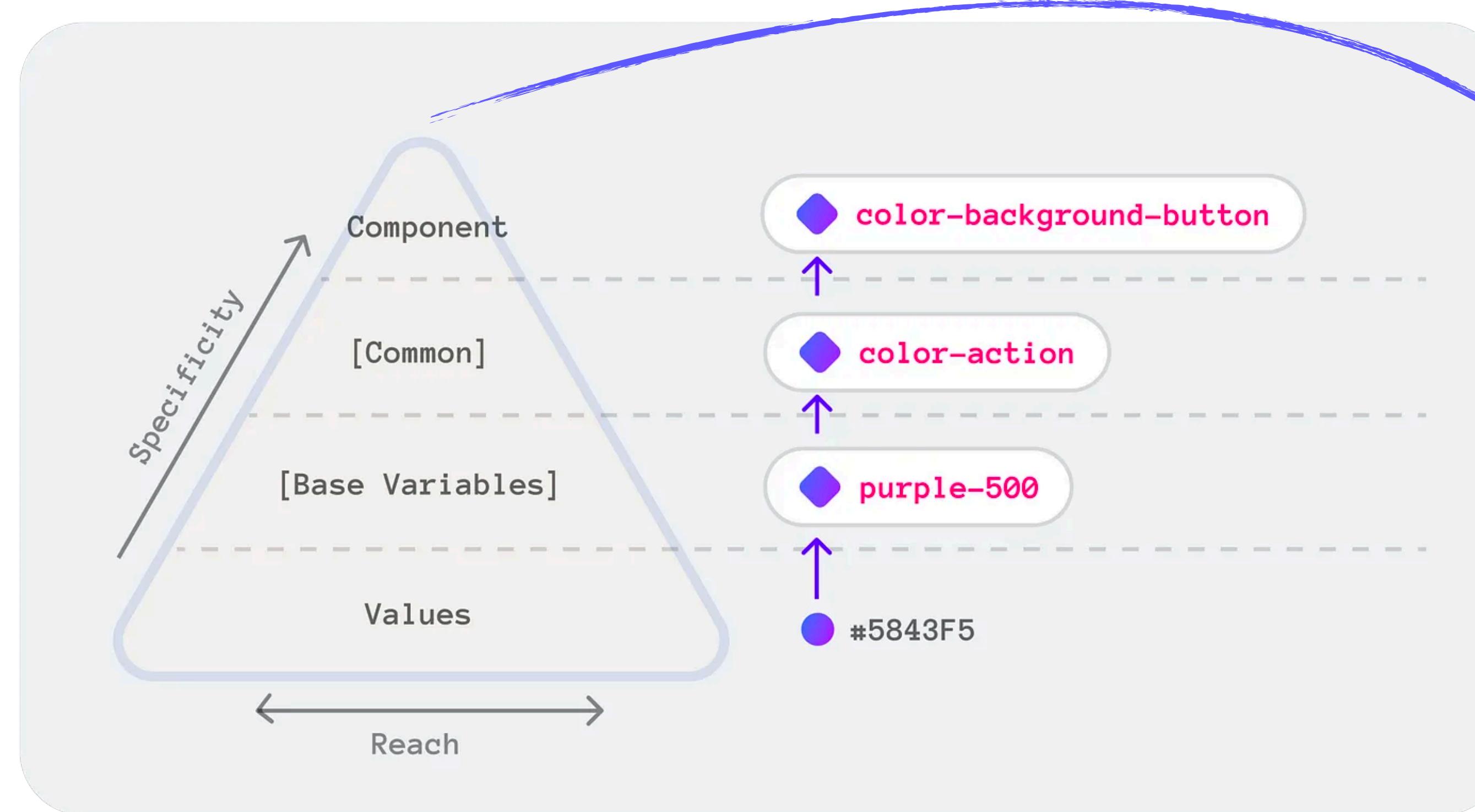
checkbox - control

checkbox - control - size

checkbox - control - size - small

Good to know principles

Design Tokens



Good to know principles

Design Tokens

domain .	category .	usage .	component .	variant .	state .	mode .	media-query
foundation	color	text	input	primary	default	light	small
--web	typography	background	button	secondary	hover	dark	medium
---specific-domain	shadow	icon	link	tertiary	active		large
--app	spacing	border	checkbox	success	focus		xlarge
	opacity	etc.	card	warning	disabled		
	border		modal	error	etc.		
			etc.				

colors-background-button-primary-hover

Build a design system

Build a design system

Why I might need to reinvent the wheel?



Build a design system

Why I might need to reinvent the wheel?

I just want to experience
making it from scratch



Build a design system

~~Why I might need to reinvent the wheel? Or I don't?~~

I just want to experience
making it from scratch

But you can experience it the
better way

Don't reinvent the wheel!

IBM



Build a design system

Why I might need to reinvent the wheel?

I just want to experience
making it from scratch

It takes more time to
modify existing component

But you can experience it the
better way



Build a design system

~~Why I might need to reinvent the wheel? Or I don't?~~

I just want to experience
making it from scratch

But you can experience it the
better way

It takes more time to
modify existing component

But I bet you're not following the
correct API approach

Don't reinvent the wheel!

IBM



Build a design system

Why I might need to reinvent the wheel?

I just want to experience making it from scratch

It takes more time to modify existing component

They don't fulfil all the requirements

But you can experience it the better way

But I bet you're not following the correct API approach



Build a design system

Why I might need to reinvent the wheel? Or I don't?

I just want to experience making it from scratch

But you can experience it the better way

It takes more time to modify existing component

But I bet you're not following the correct API approach

They don't fulfil all the requirements

Maybe you're not choosing the one that you should

Don't reinvent the wheel!

IBM



Build a design system

Why I might need to reinvent the wheel?

I just want to experience making it from scratch

It takes more time to modify existing component

They don't fulfil all the requirements

I'm not bound to any specific technology

But you can experience it the better way

But I bet you're not following the correct API approach

Maybe you're not choosing the one that you should



Build a design system

Why I might need to reinvent the wheel? Or I don't?

I just want to experience making it from scratch

But you can experience it the better way

It takes more time to modify existing component

But I bet you're not following the correct API approach

They don't fulfil all the requirements

Maybe you're not choosing the one that you should

I'm not bound to any specific technology

You're always bound to specific technologies anyway

Don't reinvent the wheel!

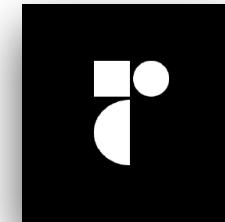
IBM



Tools and examples

Tools and examples

Radix UI



The screenshot shows the official website for Radix UI (radix-ui.com). The top navigation bar includes links for Themes, Primitives, Icons, Colors, Documentation, and Playground, along with a search icon and a user profile icon.

The main content area features a large, bold heading "Start building your app now" with a subtitle explaining the library's purpose: "An open source component library optimized for fast development, easy maintenance, and accessibility. Just import and go—no configuration required."

A code editor window displays a snippet of JavaScript/TypeScript code:

```
import '@radix-ui/themes/styles.css';
import { Theme, Button } from '@radix-ui/themes'

export default () => (
  <Theme>
    <Button>Hey 👋</Button>
  </Theme>
)
```

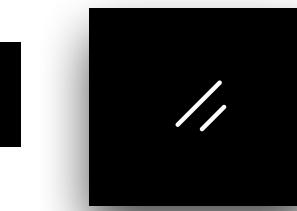
Below the code editor are two buttons: "Get started →" and "Playground".

The "Playground" section contains several examples of UI components:

- Your team**: A list of team members with their names, emails, and profile pictures. An "Invite" button is present at the top right of the list.
- Sign up**: A form for entering an email address and password, with "Create an account" and "Sign" buttons.
- Your company card**: A card for Sophie Johnson with a phone number (4929 3849 5027 1846) and a date (01/27 999). Buttons for "Freeze" and "Delete" are visible.
- Notifications**: A section for managing notification settings. It includes sections for "Comments" (Push, Email, Slack), "Favorites" (Push, Email, Slack), and "New documents" (Push).
- Invoice paid**: A confirmation message stating "You paid \$17,975.30. A receipt copy was sent" with a checkmark icon.

Tools and examples

Shadcn UI



The screenshot shows a web browser displaying the Shadcn UI website at ui.shadcn.com. The page has a dark background. At the top, there's a navigation bar with links for 'shadcn/ui', 'Documentation', 'Components', 'Themes', 'Examples', and 'GitHub'. A search bar says 'Search documentation...'. Below the navigation, a banner reads 'Introducing Style, a new CLI and more. →' and features the text 'Build your component library.' in large white font. It describes the components as 'Beautifully designed components that you can copy and paste into your apps. Accessible. Customizable. Open Source.' There are 'Get Started' and 'GitHub' buttons. The main content area shows a detailed view of a 'Dashboard' component. The dashboard has a header with a user profile for 'Alicia Koch' and navigation tabs for 'Overview', 'Customers', 'Products', and 'Settings'. It includes a search bar and a date range selector ('Jan 20, 2023 - Feb 09, 2023') with a 'Download' button. The main content area displays four cards with metrics: 'Total Revenue \$45,231.89' (+20.1% from last month), 'Subscriptions +2350' (+180.1% from last month), 'Sales +12,234' (+19% from last month), and 'Active Now +573' (+201 since last hour). At the bottom, there are 'Overview' and 'Recent Sales' buttons.

Tools and examples

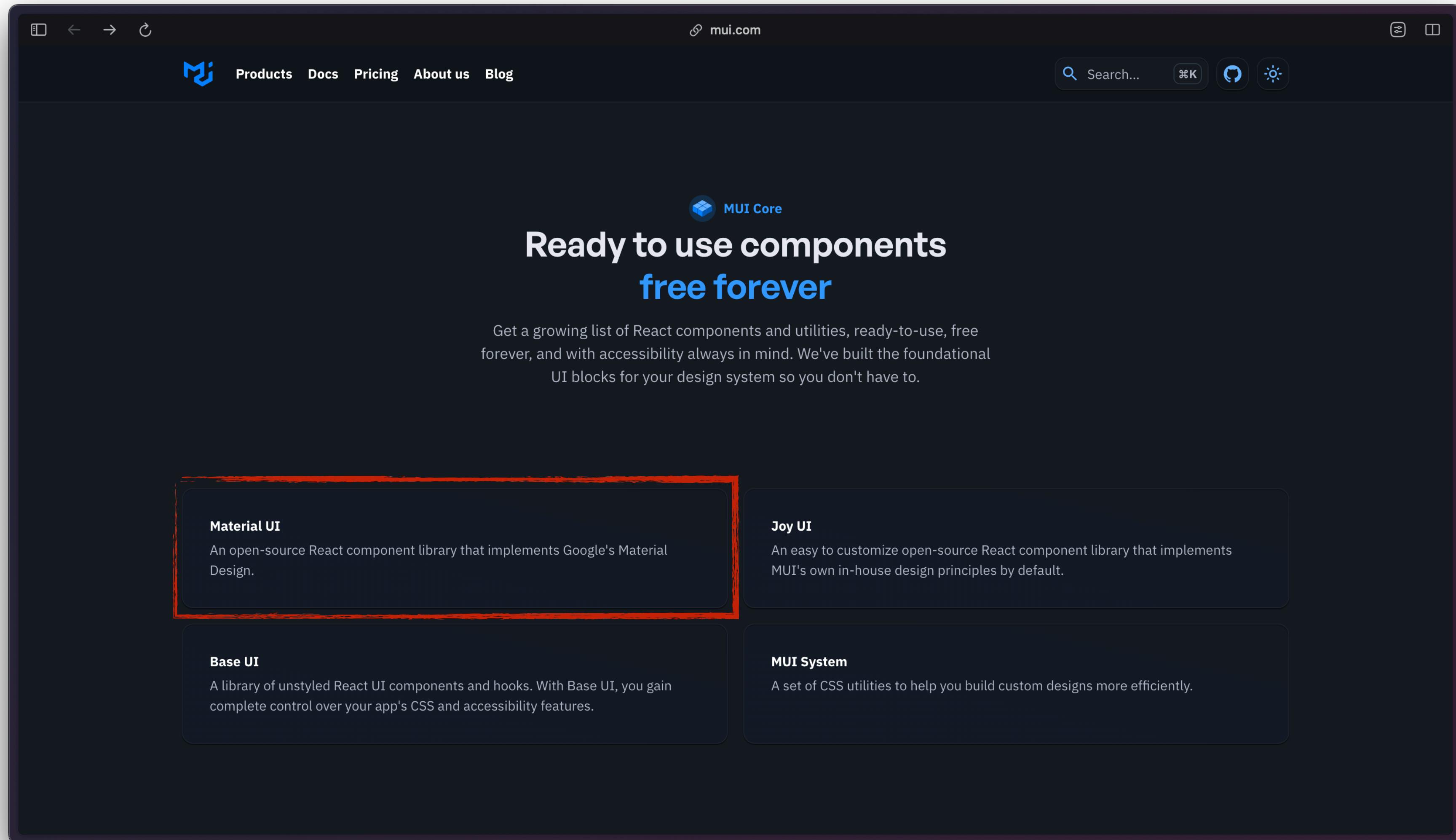
Material UI (React)

The screenshot shows the Material UI Core documentation website at mui.com/material-ui/core/. The page title is "Material UI - Overview". The main content area features a large heading "Move faster with intuitive tools" and a sub-section "MUI stands in solidarity with Ukraine.". A sidebar on the left contains a navigation menu with items like "Getting started", "Components", "Customization", "How-to guides", "Experimental APIs", "Discover more", "Migration", and "Templates". A note at the bottom of the main content states: "Material UI v5 supports Material Design v2. Adoption of v3 is tentatively planned for Material UI v6—see [the release schedule](#). You can follow this [GitHub issue](#) for future updates." The footer of the page includes sections for "CONTENTS", "Diamond sponsors" (Octopus Deploy, doT), and a "Become a Diamond sponsor" button.

The screenshot shows the main Material UI website at mui.com. The top navigation bar includes links for "Products", "Docs", "Pricing", "About us", and "Blog". The main content area features a large blue callout box with the text "Customize every button and chip instance primary color" and a progress bar indicating "60%". To the right, there's a sidebar with a section titled "Install one of our production-ready libraries to get your next app started inevitably successful:" which lists "Material UI", "Base UI", and "Joy UI". Below this are sections for "Joy UI", "Material UI", and "Base UI", each with a preview image and some text. At the bottom, there's a "Contemplative Reptile Sounds of Nature" section with a video player interface.

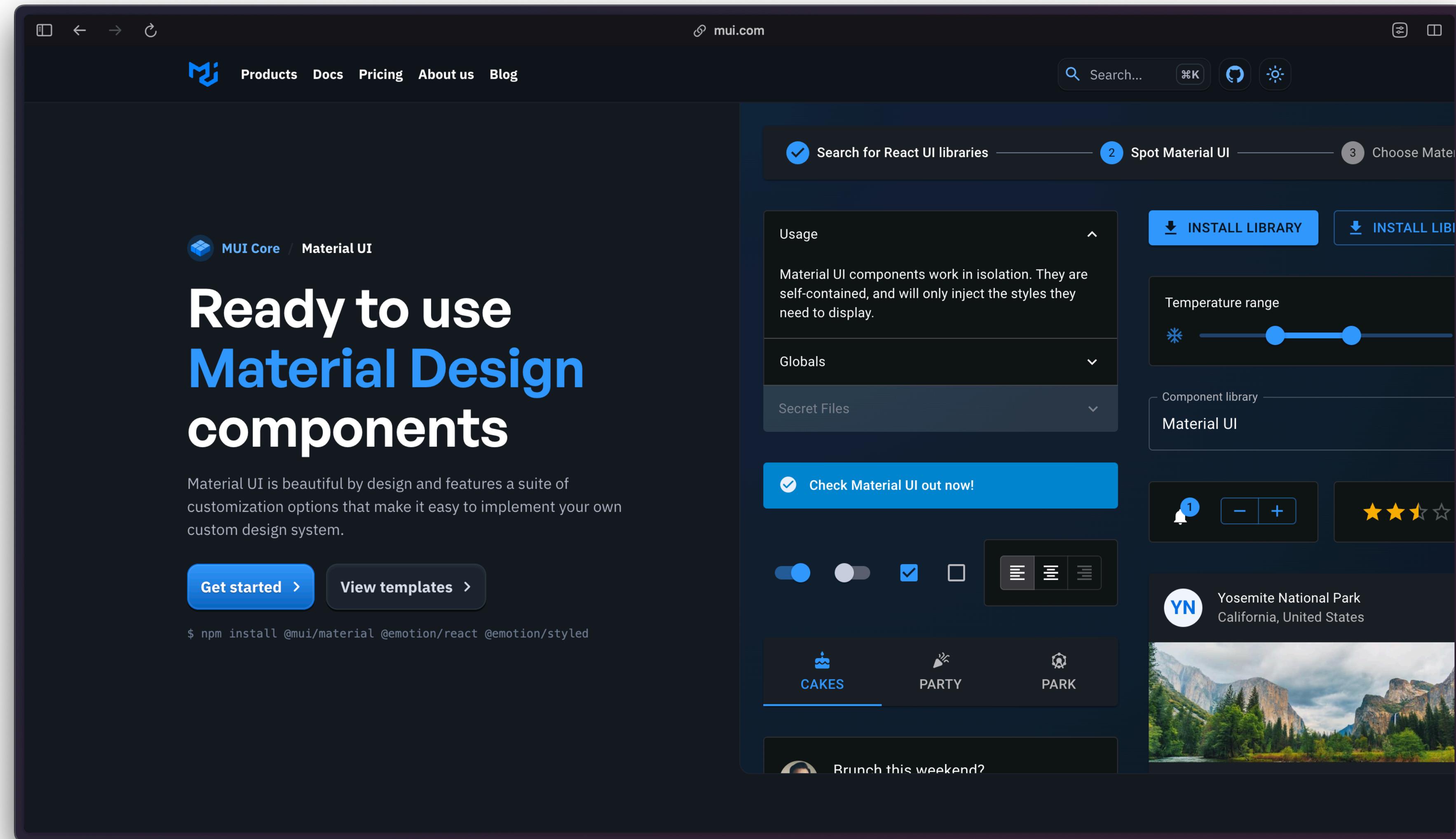
Tools and examples

Material UI (React)



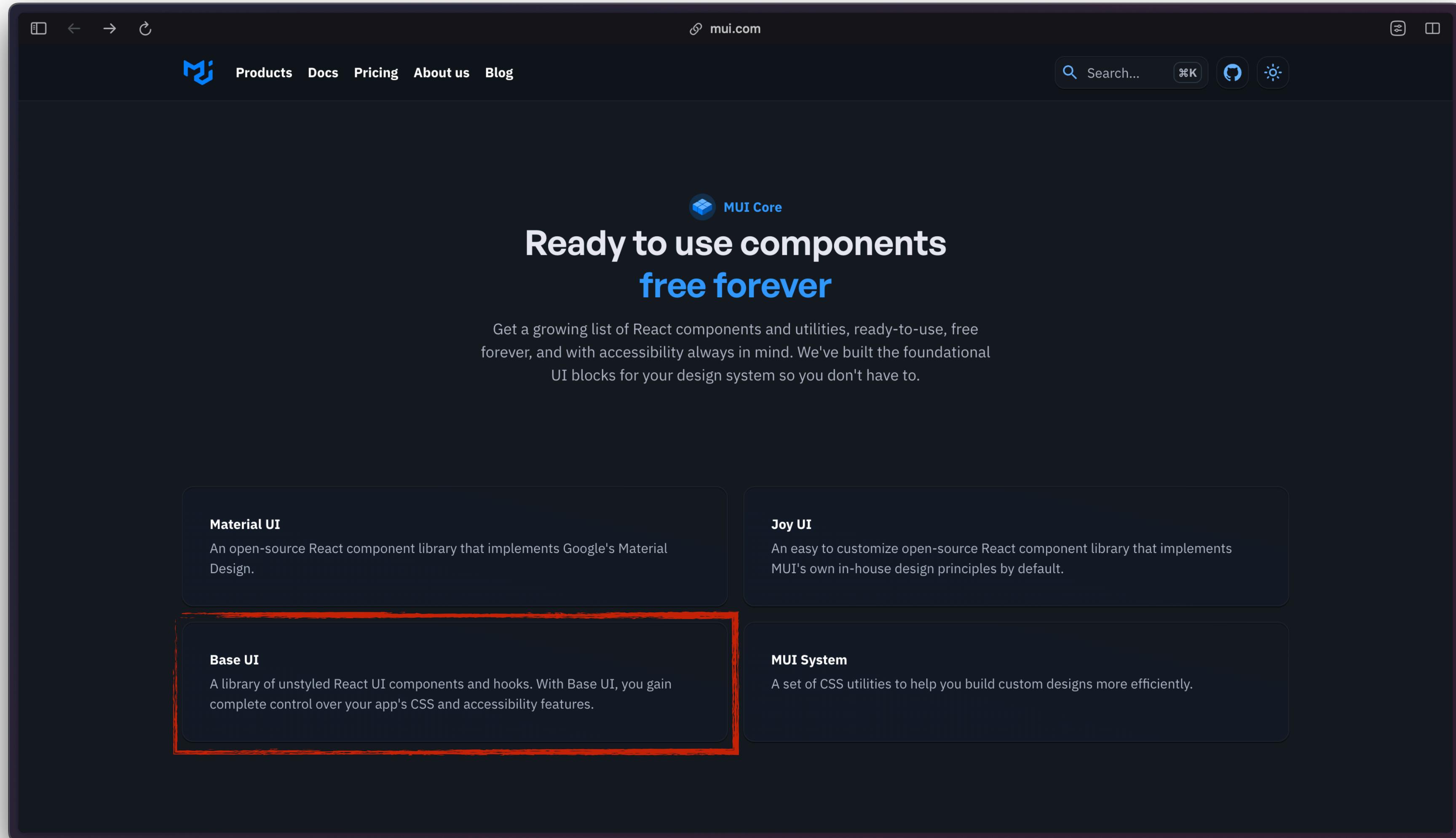
Tools and examples

Material UI (React)



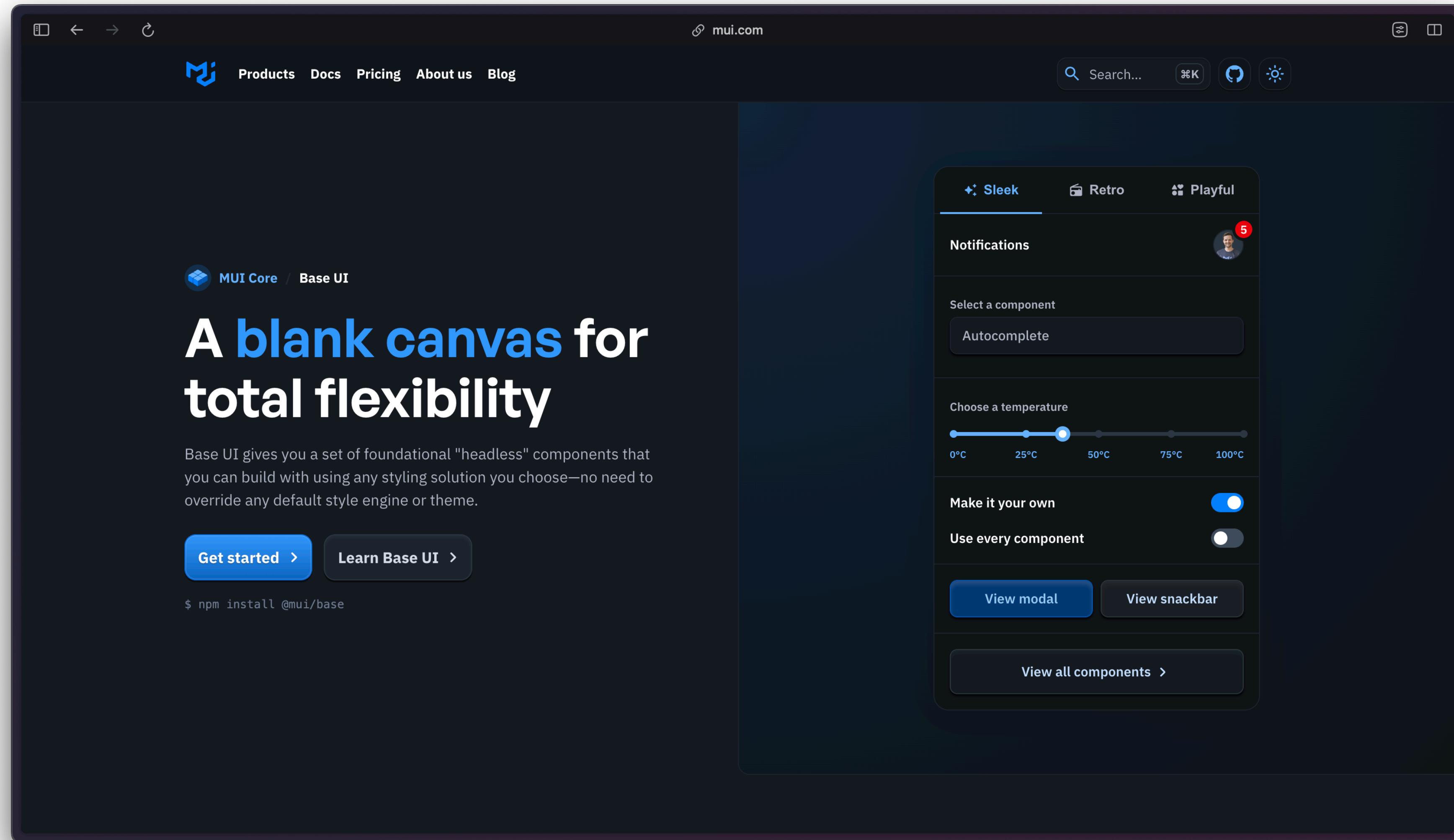
Tools and examples

Material UI (React) - Headless



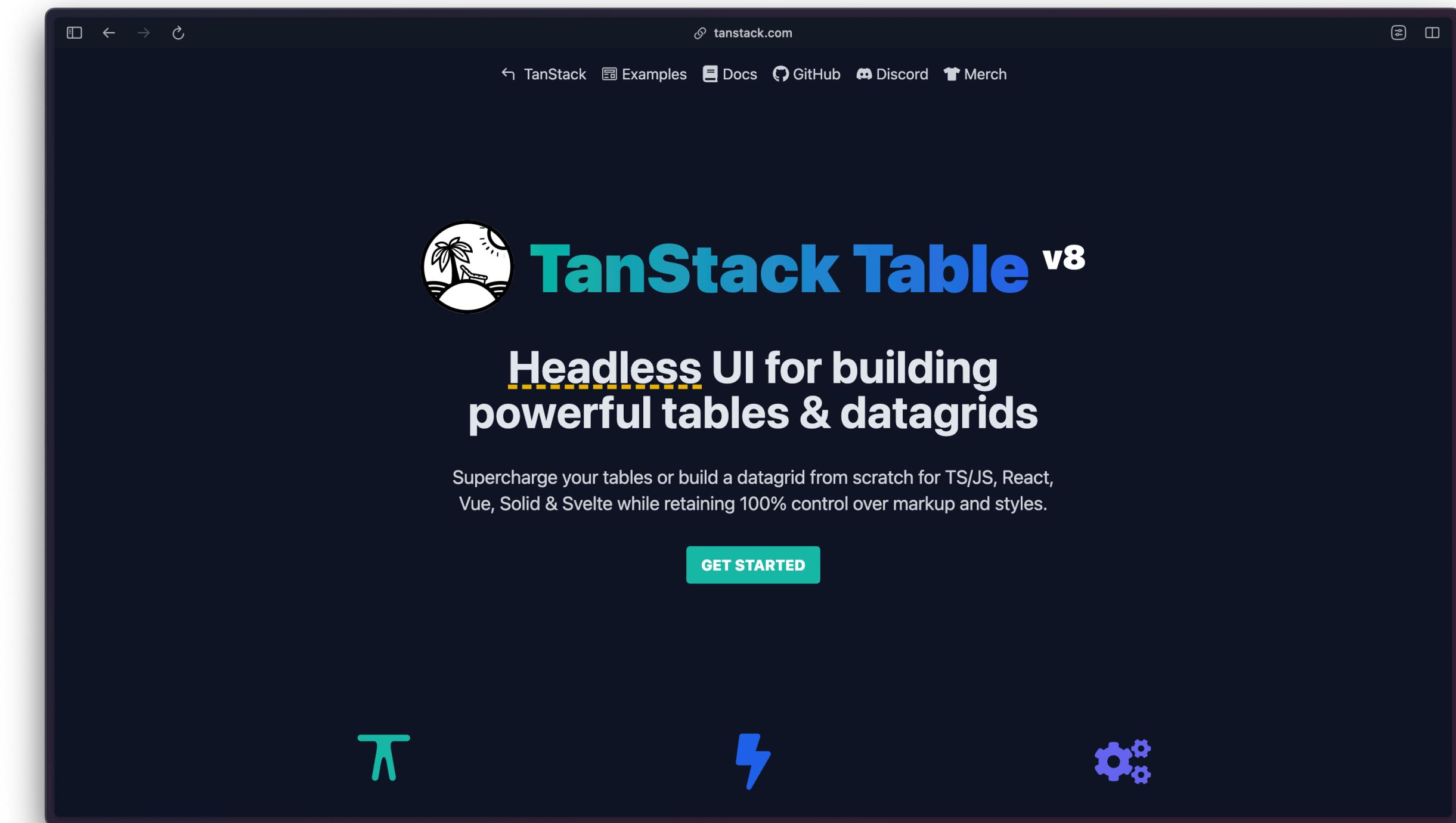
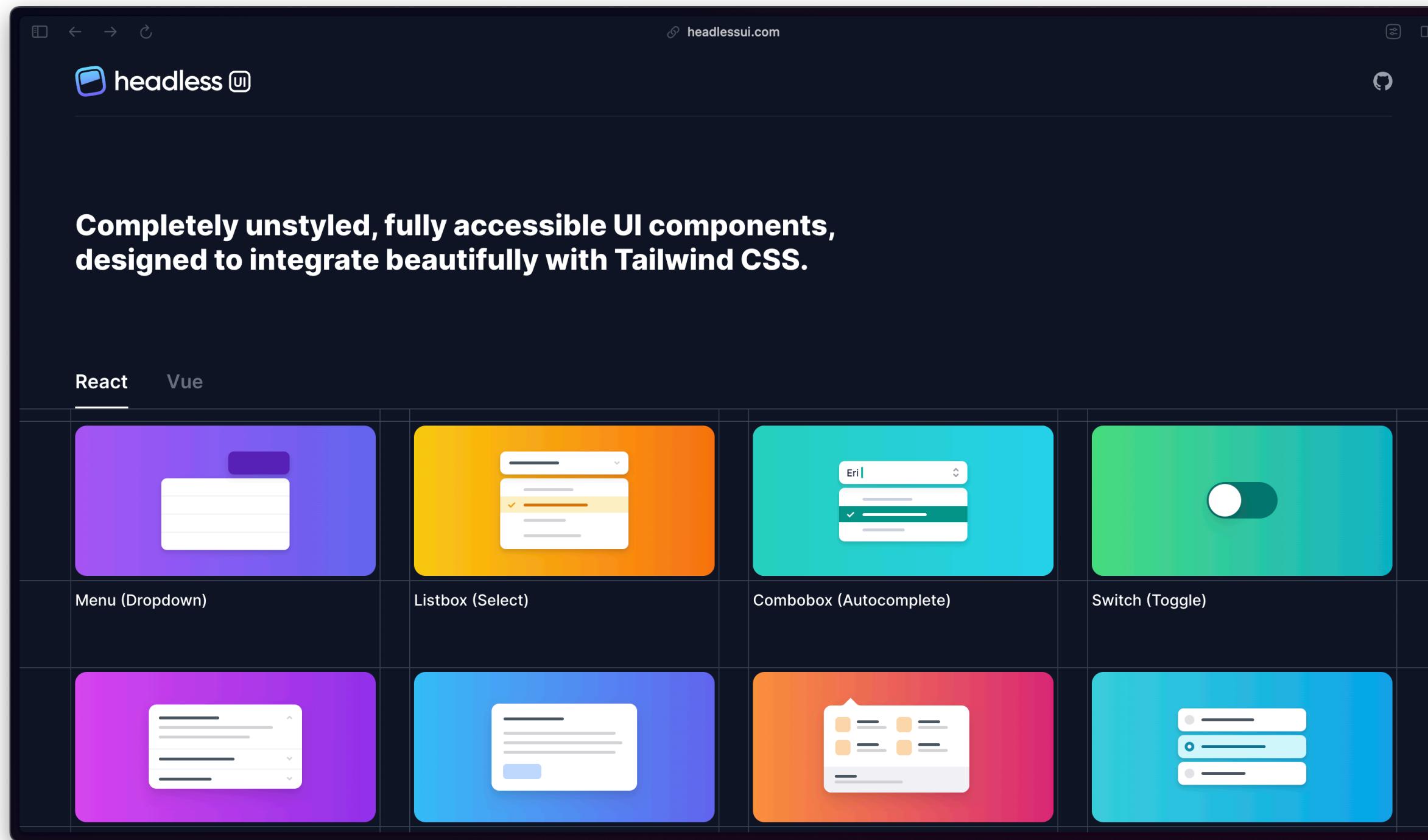
Tools and examples

Material UI (React) - Headless



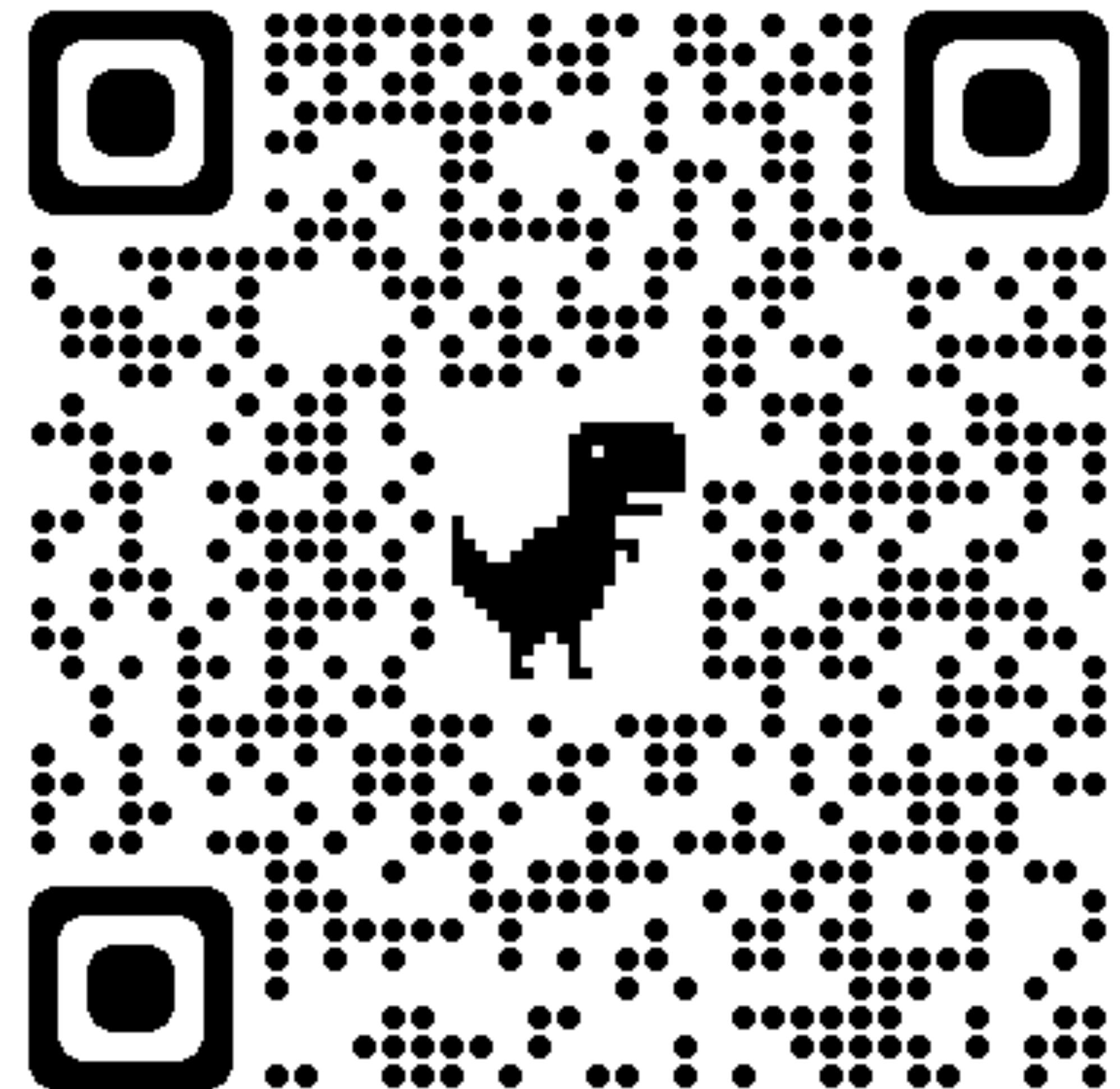
Tools and examples

Other headless design systems



Some coding...

If we have enough time for it



Key Takeaways

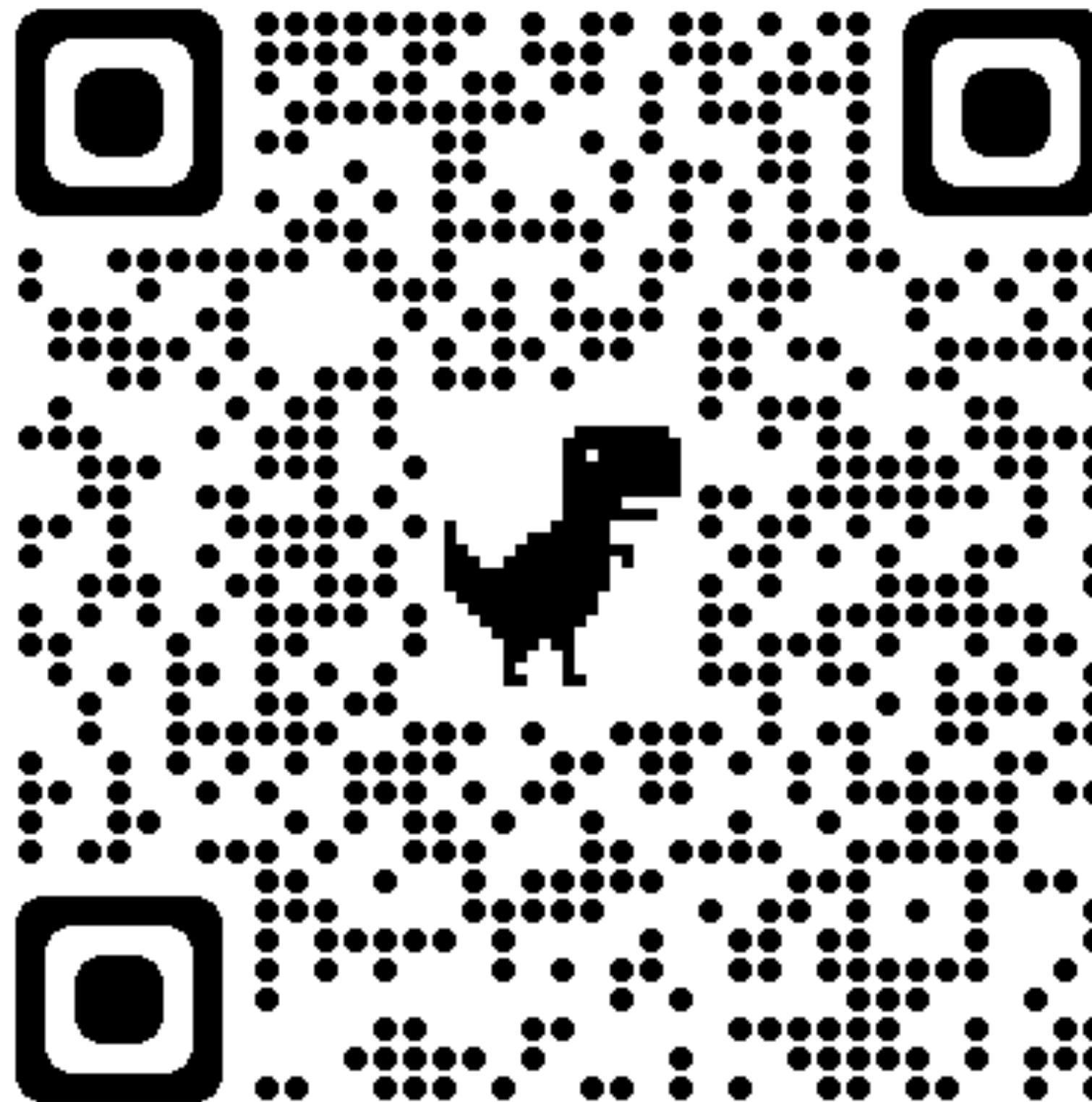
Key Takeaways

- You might not need a design system for your product right now, but you need to think about that! (You can even document the decision)
- If you need to have your design system, most likely, you won't need to build everything yourself from scratch.
- There are so many options to modify and use, if you're unsure, either one is probably fine.
- We need to find a language to make frontend/product design communications as efficient as possible.

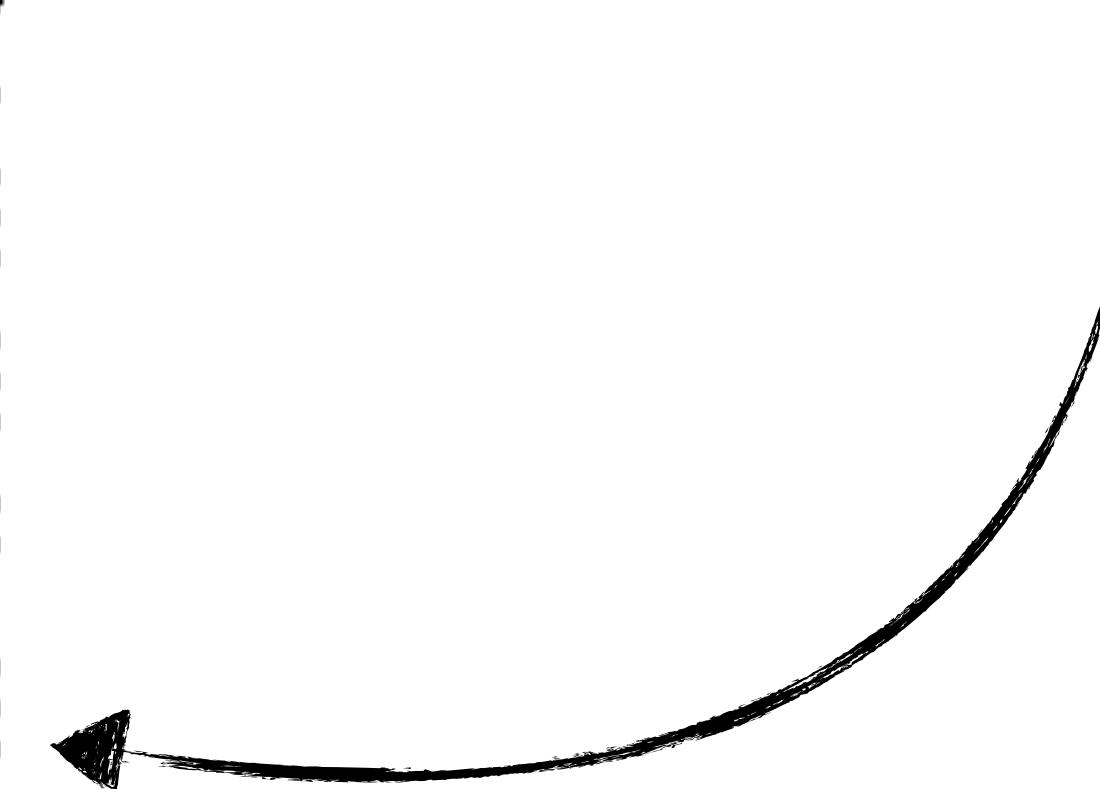
QnA

Any Question?

You can just ask, or find me around if anything.



Link to the presentation and repo



safehiana@gmail.com

References

- <https://bootcamp.uxdesign.cc/design-systems-what-they-are-how-to-use-them-and-the-benefits-daf89b5a479b>
- <https://bradfrost.com/blog/post/extending-atomic-design/>
- <https://ui.shadcn.com/>
- <https://www.radix-ui.com/>
- <https://mui.com/>
- <https://tanstack.com/table>
- <https://uxdesign.cc/design-tokens-cheatsheet-927fc1404099>
- <https://spectrum.adobe.com/page/design-tokens/>
- <https://github.com/dipsaus9/design-system-monorepo>

Images:

- <https://www.pinterest.com/pin/763852786783416303/>
- <https://4dproducts.co.uk/bad-product-design-how-can-you-avoid-poorly-designed-products/>
- <https://www.invisionapp.com/inside-design/6-design-dev-memes/>