OVHcloud

Ask the Expert

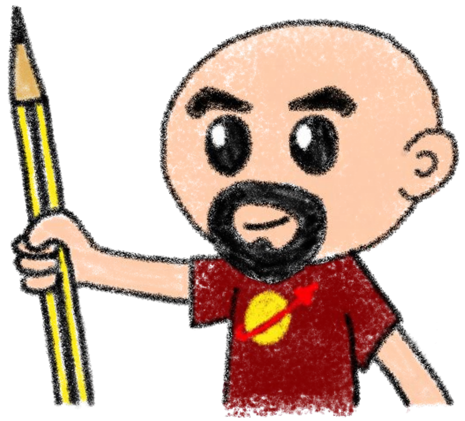
Futureproof Design Systems with Web Components

Horacio Gonzalez

2020-06-22

Who are we?

Introducing myself and introducing ~~OVH~~ OVHcloud



Horacio Gonzalez

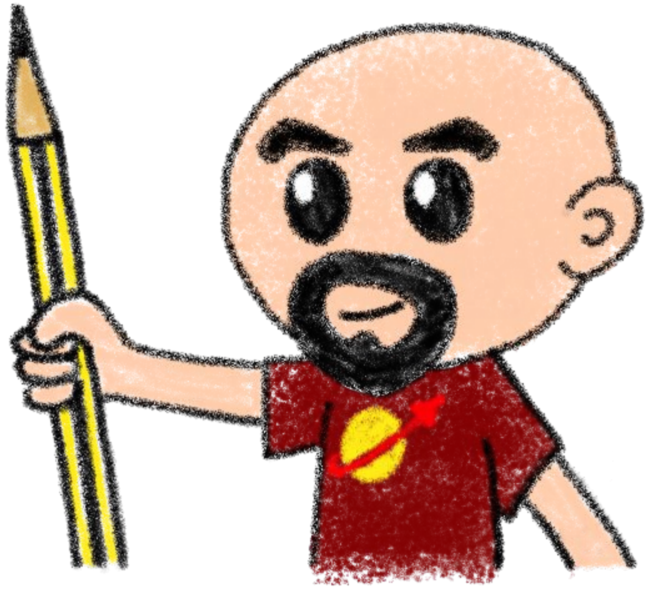
Ask the Expert

@LostInBrittany

Spaniard lost in Brittany, developer, dreamer and all-around geek

OVHcloud

DevRel Leader



DevFest du Bout du Monde

Finist Devs

Google Developers Experts 2019

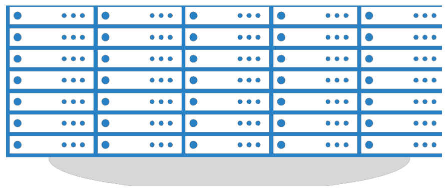
Web Technologies GDE

Flutter

OVHcloud: A Global Leader

200k Private cloud VMs running

1 **Dedicated IaaS Europe**



Hosting capacity : **1.3M** Physical Servers

360k Servers already deployed

Own **20Tbps** Netwok with **35 PoPs**

30 Datacenters

> **1.3M** Customers in **138** Countries

OVHcloud: 4 Universes of Products

WebCloud

Domain / Email

Domain names, DNS, SSL, Redirect

Email, Open-Xchange, Exchange

Collaborative Tools, NextCloud

PaaS for Web

Mutu, CloudWeb

Plesk, CPanel

PaaS with Platform.sh

Virtual servers

VPS, Dedicated Server

SaaS

Wordpress, Magento, Prestashop

CRM, Billing, Payment, Stats

MarketPlace

Support, Managed

Support Basic

Support thought Partners

Managed services

Baremetal Cloud

Standalone, Cluster

General Purpose	
SuperPlan	
Game	T2 >20e
Virtualization	
Storage	T3 >80e
Database	T4 >300e
Bigdata	T5 >600e
HCI	12KVA /32KVA
AI	
VDI Cloud Game	
Network	

VPS aaS

pCC DC

Virtuozzo Cloud

Wholesales

IT Integrators, Cloud Storage,

CDN, Database, ISV, WebHosting

High Intensive CPU/GPU,

Encrypt

KMS, HSM

Encrypt (SGX, Network, Storage)

Public Cloud

Compute

VM	K8S, IA IaaS
Baremetal	PaaS for DevOps

Storage

File, Block, Object, Archive

Databases

SQL, noSQL, Messaging,

Dashboard

Network

IP FO, NAT, LB, VPN, Router,

DNS, DHCP, TCP/SSL Offload

Security

IAM, MFA, Encrypt, KMS

IA, DL

Standard Tools for AI, AI Studio,

IA IaaS, Hosting API AI

Bigdata, ML, Analytics

Datalake, ML, Dashboard

Hosted Private Cloud

Hosted Private Cloud

VMware

SDDC, vSAN 1AZ / 2AZ
vCD, Tanzu, Horizon, DBaaS,
DRaaS

Nutanix

HCI 1AZ / 2AZ, Databases,
DRaaS, VDI

OpenStack

IAM, Compute (VM, K8S)
Stortage, Network, Databases

Storage

Ontap Select, Nutanix File
OpenIO, MinIO, CEPH
Zerto, Veeam, Atempo

AI

ElementAI, HuggingFace,
Deepopmatic, Systran,
EarthCube

Bigdata / Analitics / ML

Cloudera over S3, Dataiku,
Saagie, Tableau,

Hybrid Cloud

vRack Connect, Edge-DC, Private DC

Dell, HP, Cisco, OCP, MultiCloud

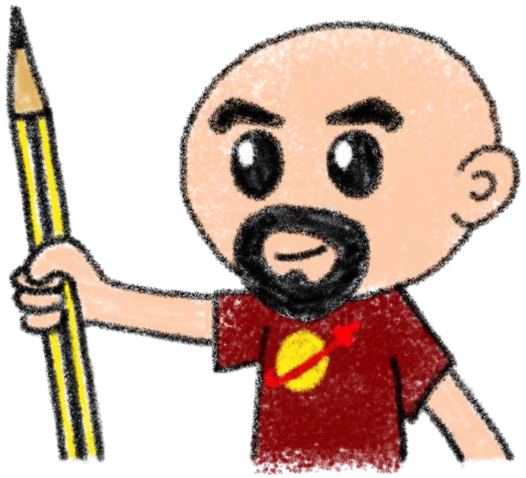
Secured Cloud

GOV, FinTech, Retail, HealtCare

Ask the Expert

Disclaimer

Before going further...

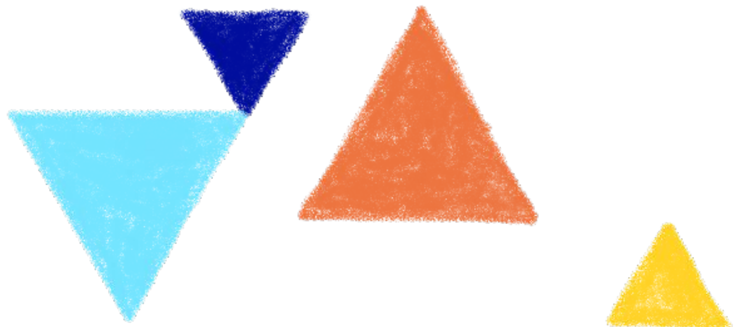


A talk for devs by a dev

I am not a designer, neither I play one on TV...

Design isn't only look and feel

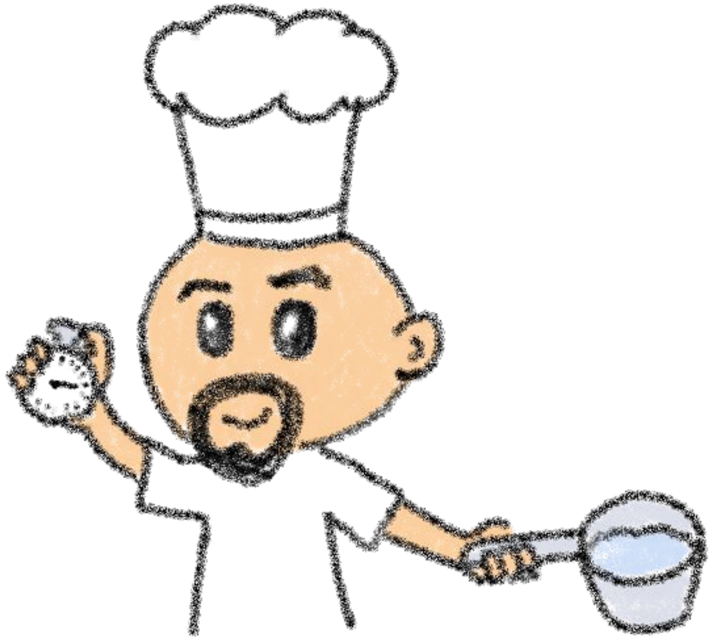
Look and feel must be a emanation of brand's **ethos**, **values** and **spirit**



Example

Ask the Expert

10 minutes
delicious dinners



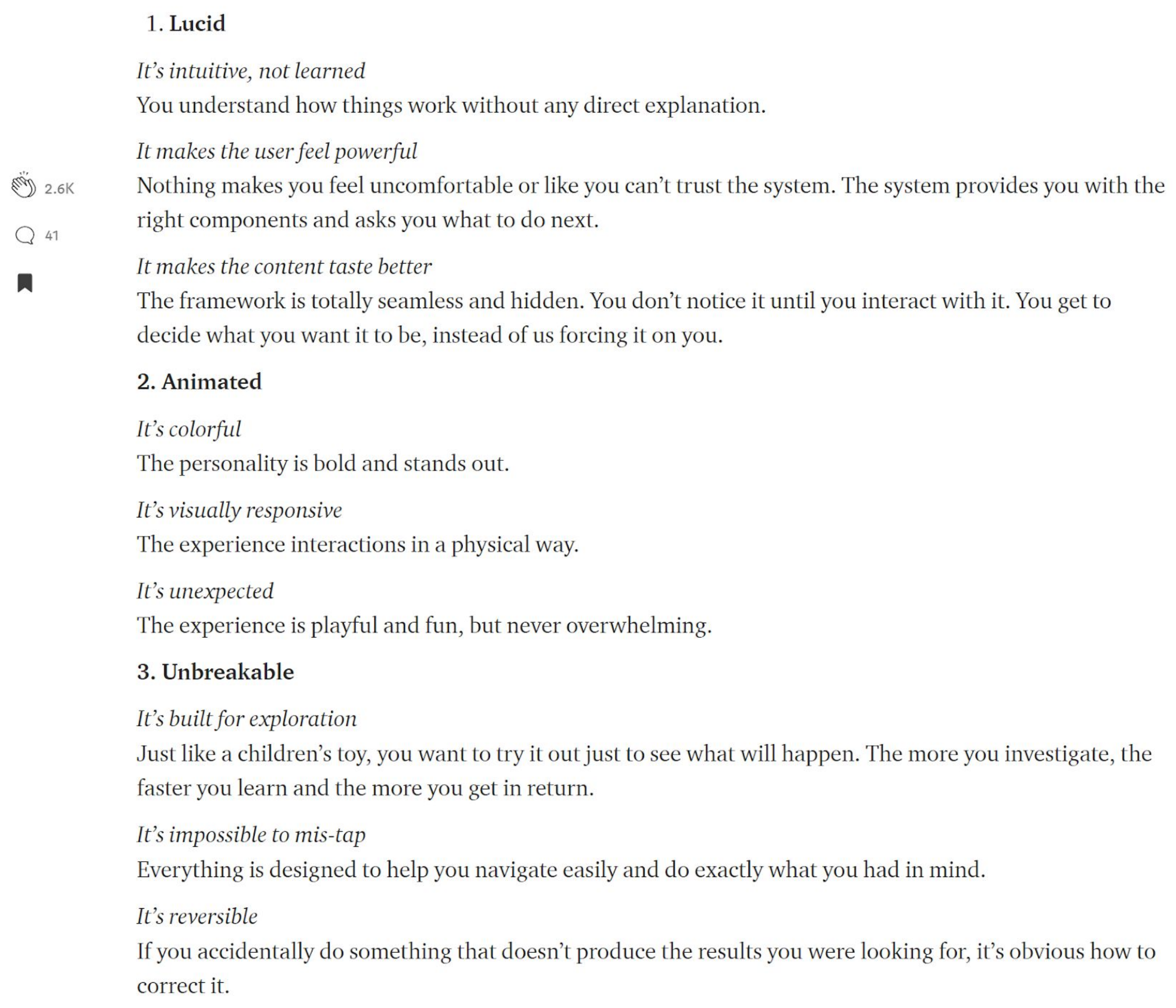
$$\underbrace{\text{Behaviors \& Functional Patterns} \quad \text{Aesthetics \& Perceptual Patterns}}_{\text{Design Principles}} \xrightarrow{\hspace{3em}} \text{Shared language}$$

Design principles

Ask the Expert

Grounding **principles** and **values**
emanating from the **brand ethos**,
shared by everybody around the **product**.

Example of Design Principles: Pinterest



1. **Lucid**

It's intuitive, not learned
You understand how things work without any direct explanation.

It makes the user feel powerful
Nothing makes you feel uncomfortable or like you can't trust the system. The system provides you with the right components and asks you what to do next.

It makes the content taste better
The framework is totally seamless and hidden. You don't notice it until you interact with it. You get to decide what you want it to be, instead of us forcing it on you.

2. **Animated**

It's colorful
The personality is bold and stands out.

It's visually responsive
The experience interactions in a physical way.

It's unexpected
The experience is playful and fun, but never overwhelming.

3. **Unbreakable**

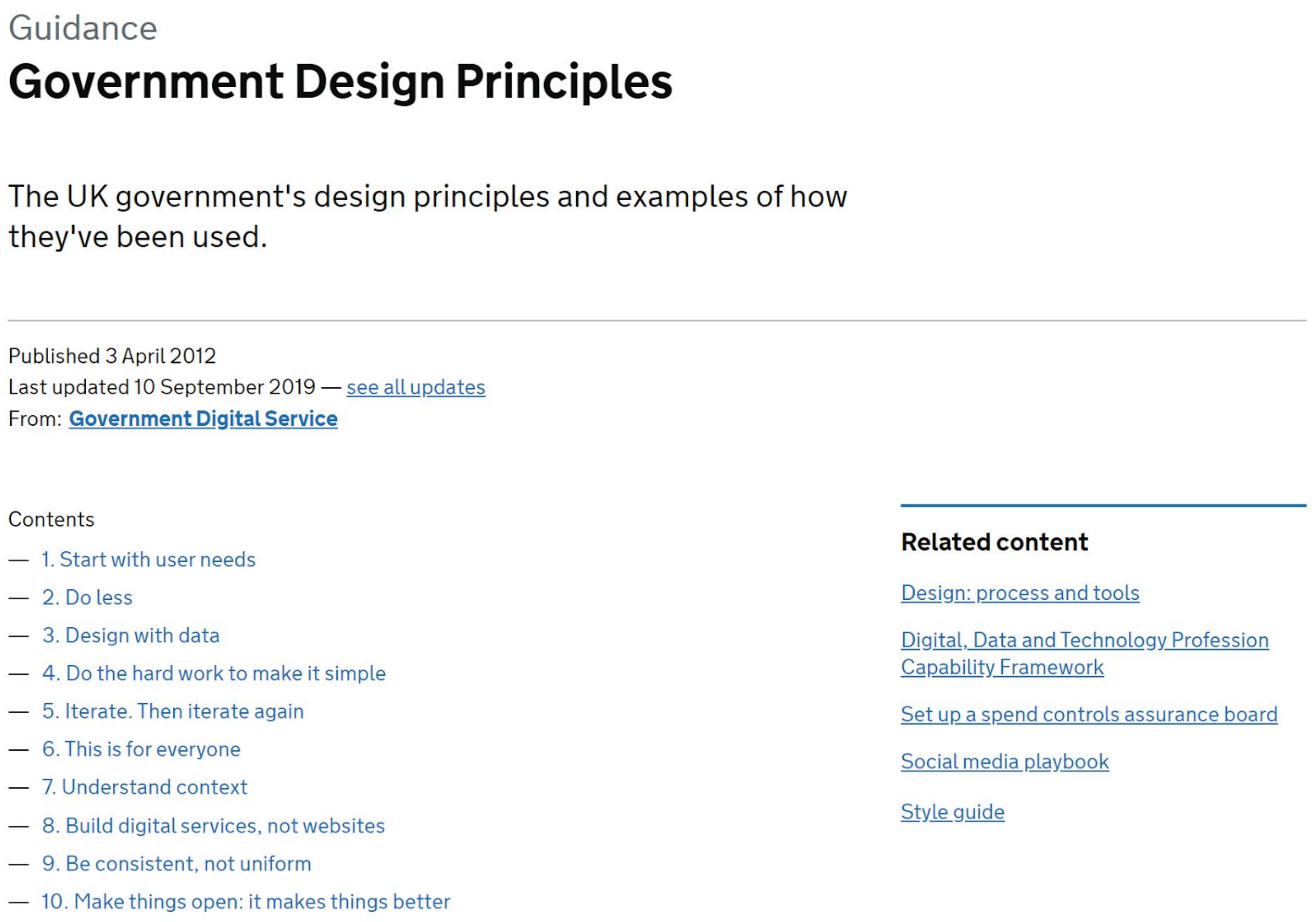
It's built for exploration
Just like a children's toy, you want to try it out just to see what will happen. The more you investigate, the faster you learn and the more you get in return.

It's impossible to mis-tap
Everything is designed to help you navigate easily and do exactly what you had in mind.

It's reversible
If you accidentally do something that doesn't produce the results you were looking for, it's obvious how to correct it.

https://medium.com/@suprb/redesigning-pinterest-block-by-block-6040a00d80a3

Example of Design Principles: gov.uk



Guidance

Government Design Principles

The UK government's design principles and examples of how they've been used.

Published 3 April 2012
Last updated 10 September 2019 — see all updates
From: **Government Digital Service**

Contents

- 1. Start with user needs
- 2. Do less
- 3. Design with data
- 4. Do the hard work to make it simple
- 5. Iterate. Then iterate again
- 6. This is for everyone
- 7. Understand context
- 8. Build digital services, not websites
- 9. Be consistent, not uniform
- 10. Make things open: it makes things better

Related content

Design: process and tools

Digital, Data and Technology Profession Capability Framework

Set up a spend controls assurance board

Social media playbook

Style guide

https://www.gov.uk/guidance/government-design-principles

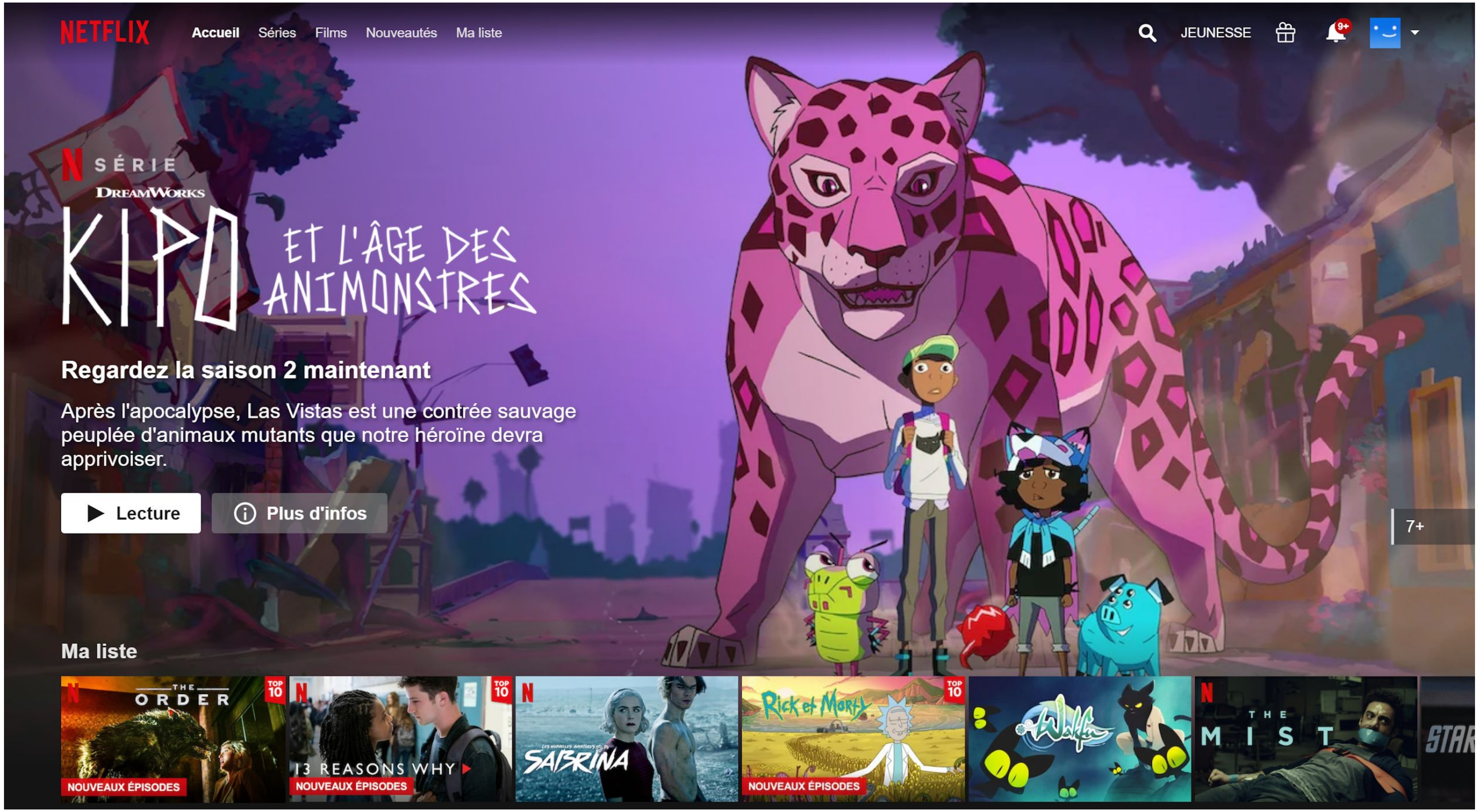
Functional Patterns

Ask the Expert

Functional patterns are the **building blocks** of the **user interface**.

They must **guide** and **facilitate** user's **behavior**.

Functional Patterns Example: Netflix



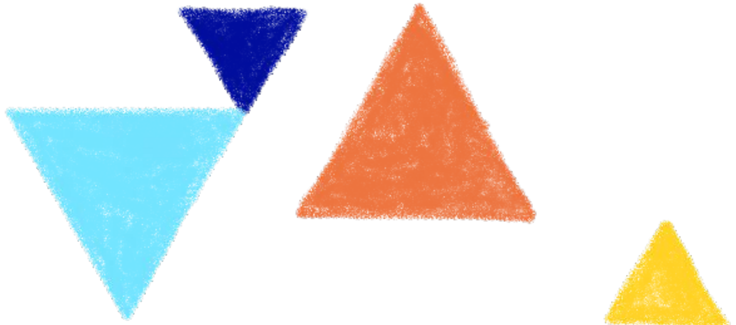
Regardez la saison 2 maintenant

Après l'apocalypse, Las Vistas est une contrée sauvage peuplée d'animaux mutants que notre héroïne devra apprivoiser.

Perceptual Patterns

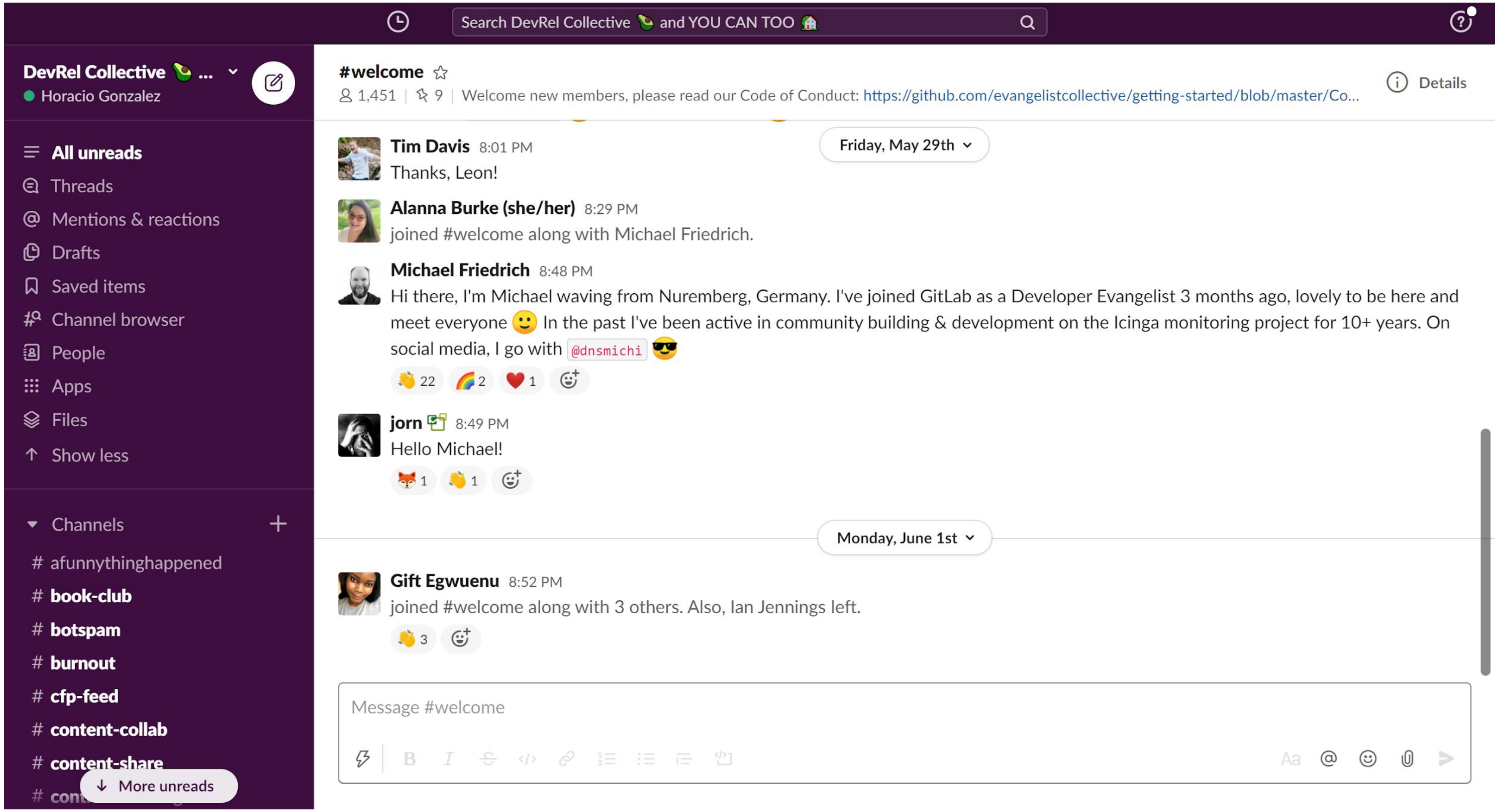
Perceptual patterns are **visual elements** defining the **look** of the product: typography, color palette, illustration styles, layout shapes, textures…

They should help to **express** the **brand image**.



Perceptual Patterns Example: Slack

Ask the Expert



Search DevRel Collective and YOU CAN TOO

DevRel Collective ...
Horacio Gonzalez

- All unreads
- Threads
- Mentions & reactions
- Drafts
- Saved items
- Channel browser
- People
- Apps
- Files
- Show less

Channels

- # afunnythinghappened
- # book-club
- # botspam
- # burnout
- # cfp-feed
- # content-collab
- # content-share
- # cont

More unreads

#welcome

1,451 | 9 | Welcome new members, please read our Code of Conduct: https://github.com/evangelistcollective/getting-started/blob/master/Co...

Details

Friday, May 29th

Tim Davis 8:01 PM
Thanks, Leon!

Alanna Burke (she/her) 8:29 PM
joined #welcome along with Michael Friedrich.

Michael Friedrich 8:48 PM
Hi there, I'm Michael waving from Nuremberg, Germany. I've joined GitLab as a Developer Evangelist 3 months ago, lovely to be here and meet everyone 🙂 In the past I've been active in community building & development on the Icinga monitoring project for 10+ years. On social media, I go with `@dnsmichi` 😎
👋 22 🌈 2 ❤️ 1

jorn 8:49 PM
Hello Michael!
🦊 1 👋 1

Monday, June 1st

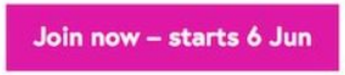
Gift Egwuenu 8:52 PM
joined #welcome along with 3 others. Also, Ian Jennings left.
👋 3

Message #welcome

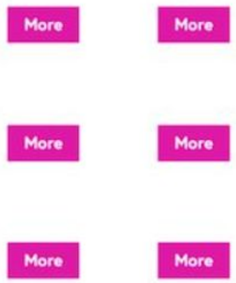
Shared Languages

A product team need to **share a pattern language** based on **design principles**, to create a **coherent** set of **functional** and **perceptual patterns**.

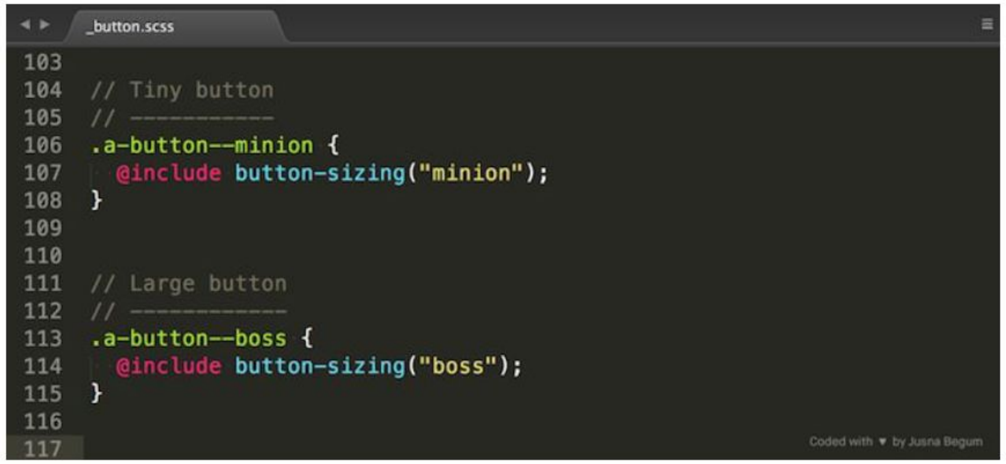
Shared Languages Example: Future Learn



Boss button.



Minion buttons.



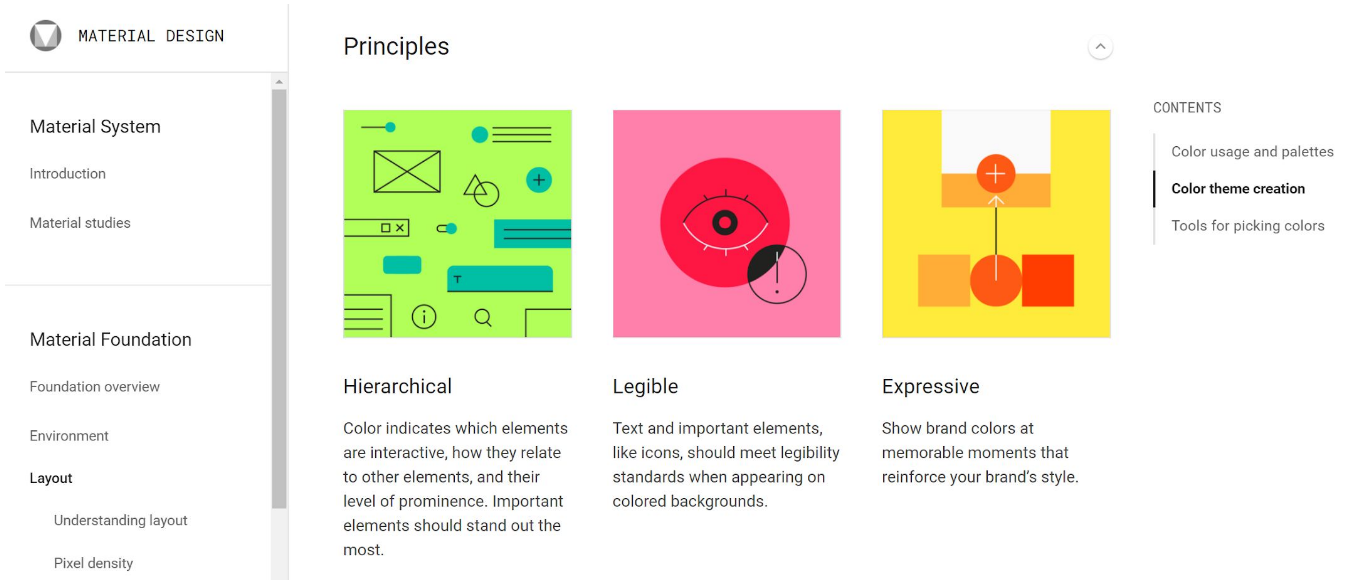
```scss

// Tiny button
// -----------
.a-button--minion {
  @include button-sizing("minion");
}


// Large button
// ------------
.a-button--boss {
  @include button-sizing("boss");
}

```

.minion and .boss class names in CSS.

Ask the Expert

So, what are Design Systems?

And why should I look at them?



MATERIAL DESIGN

Material System

Introduction

Material studies

Material Foundation

Foundation overview

Environment

Layout

Understanding layout

Pixel density

Principles

Hierarchical

Color indicates which elements are interactive, how they relate to other elements, and their level of prominence. Important elements should stand out the most.

Legible

Text and important elements, like icons, should meet legibility standards when appearing on colored backgrounds.

Expressive

Show brand colors at memorable moments that reinforce your brand's style.

CONTENTS

Color usage and palettes

Color theme creation

Tools for picking colors

The same or different?

Ask the Expert

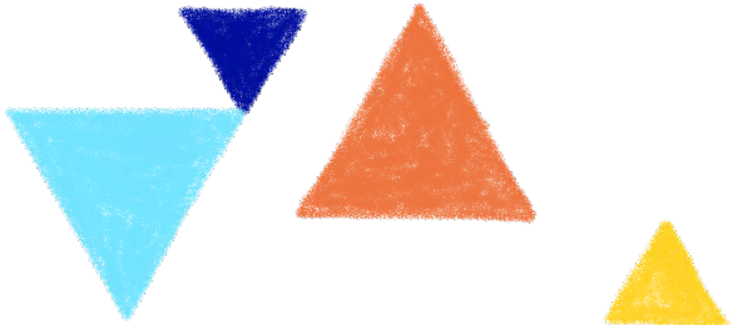
Design System

Component Catalog

Style Guide

Style Guides

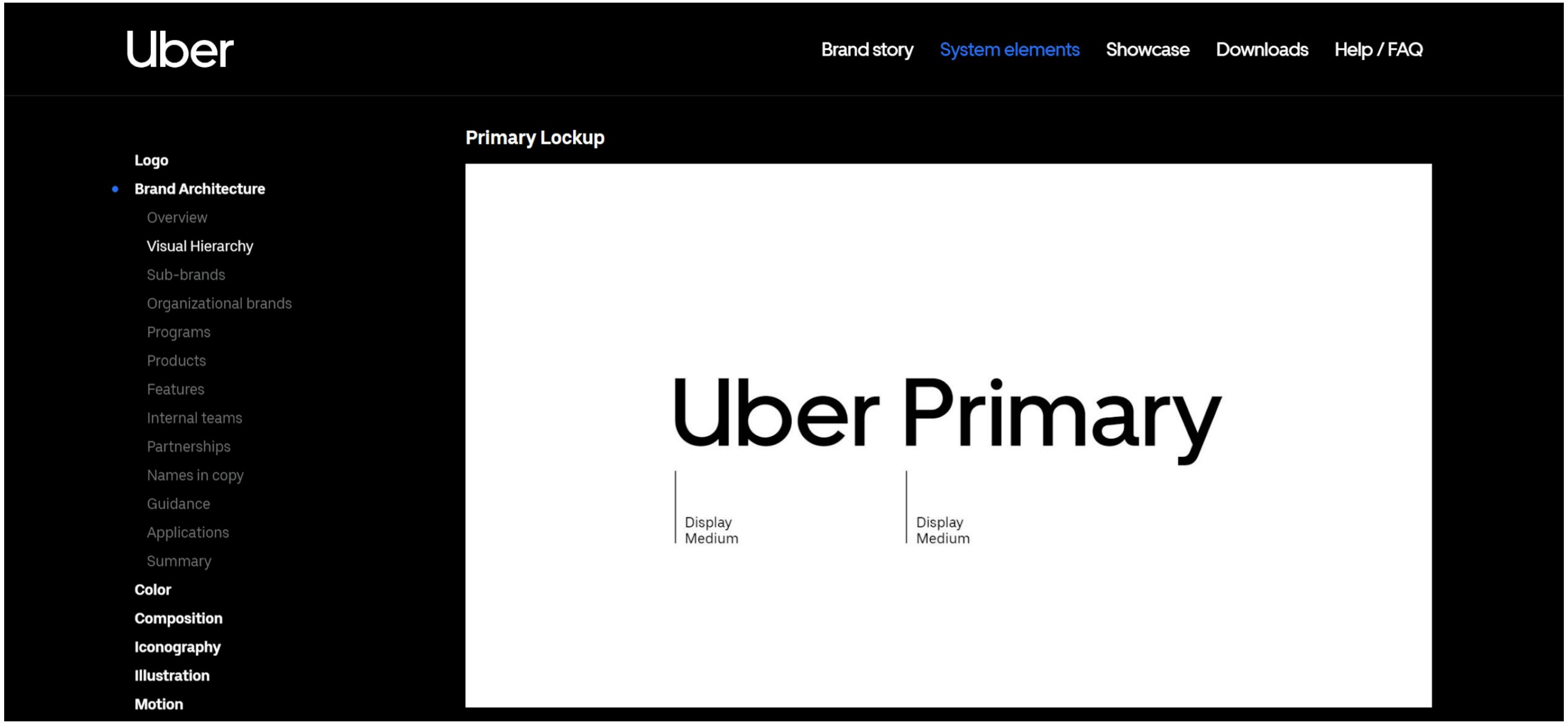
A **document** listing the **styles**, **patterns**, **practices**, and **principles** of a brand **design standards**



Style Guides

Style guides define the **application's look and feel**

Style Guide Example: Uber


Uber

Brand story · System elements · Showcase · Downloads · Help / FAQ

- Logo
- Brand Architecture
  - Overview
  - Visual Hierarchy
  - Sub-brands
  - Organizational brands
  - Programs
  - Products
  - Features
  - Internal teams
  - Partnerships
  - Names in copy
  - Guidance
  - Applications
  - Summary
- Color
- Composition
- Iconography
- Illustration
- Motion

Primary Lockup

Uber Primary

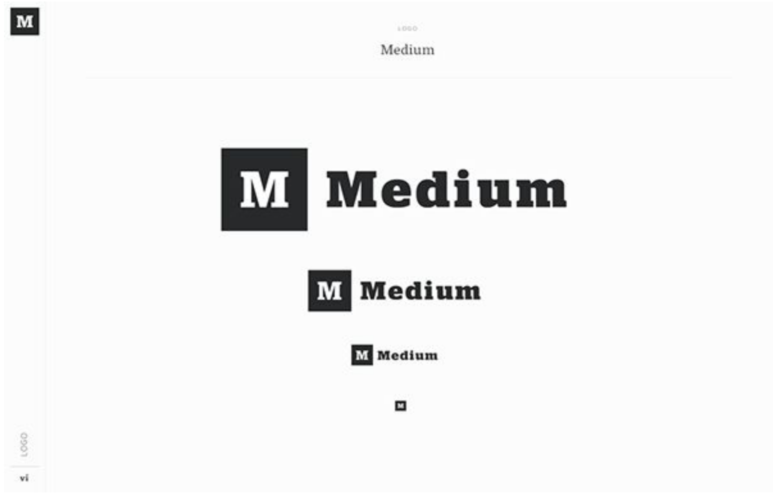
Display Medium

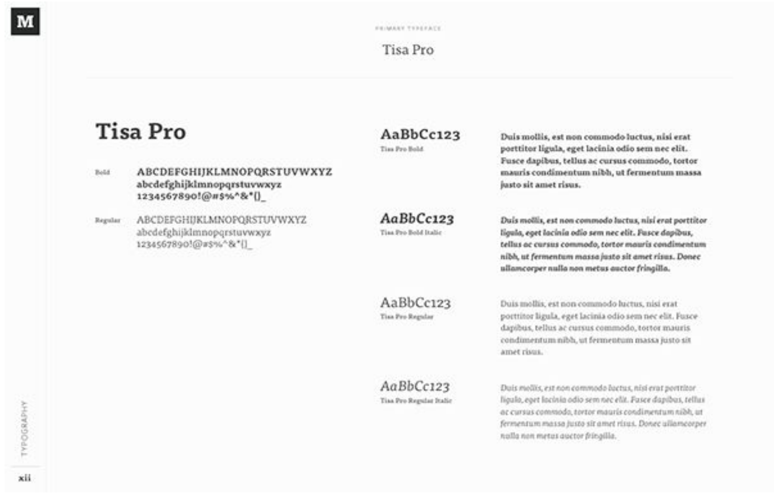
Display Medium

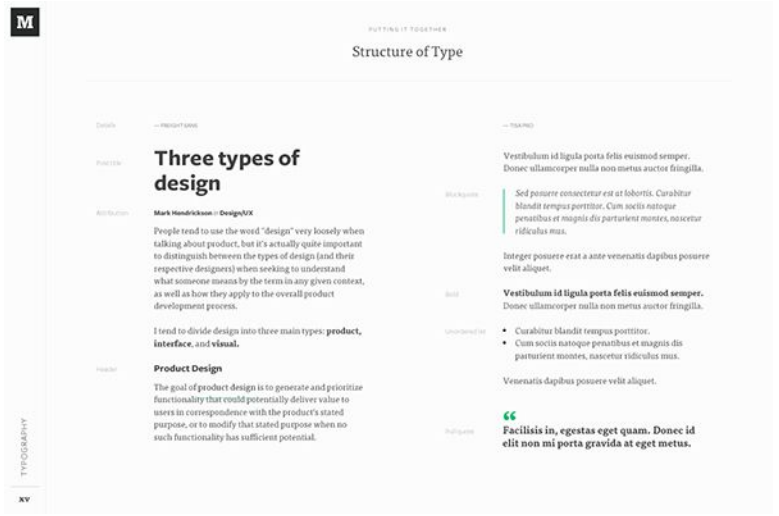
https://brand.uber.com/

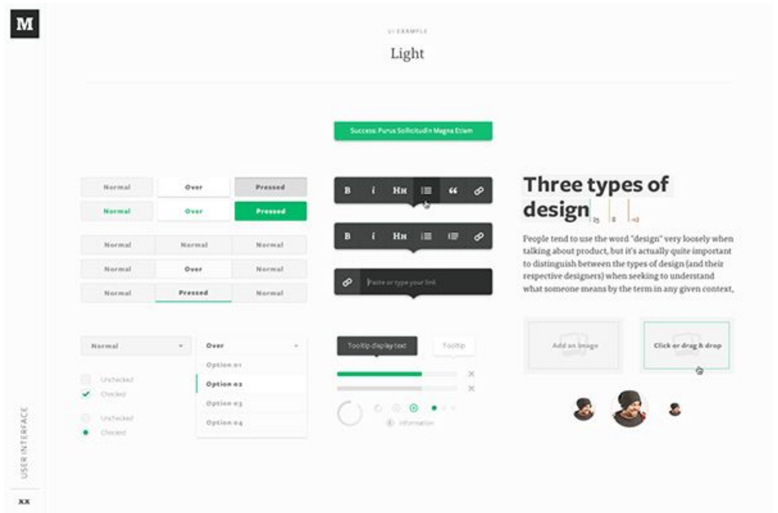
Style Guide Example: Medium

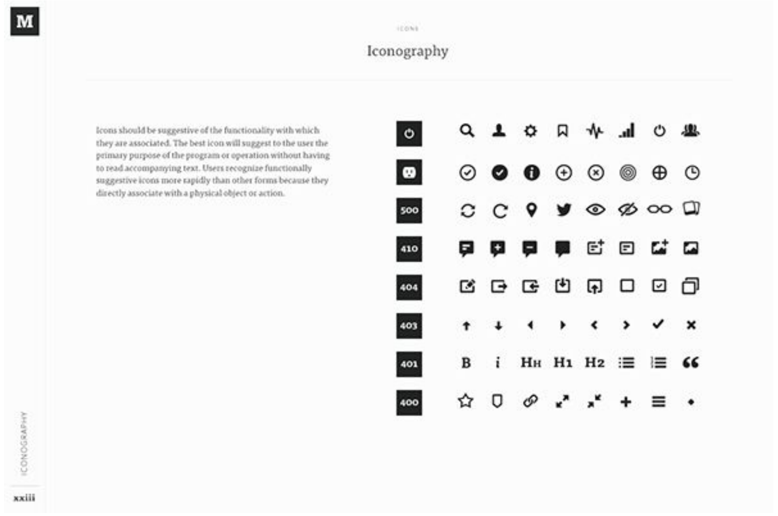
Ask the Expert





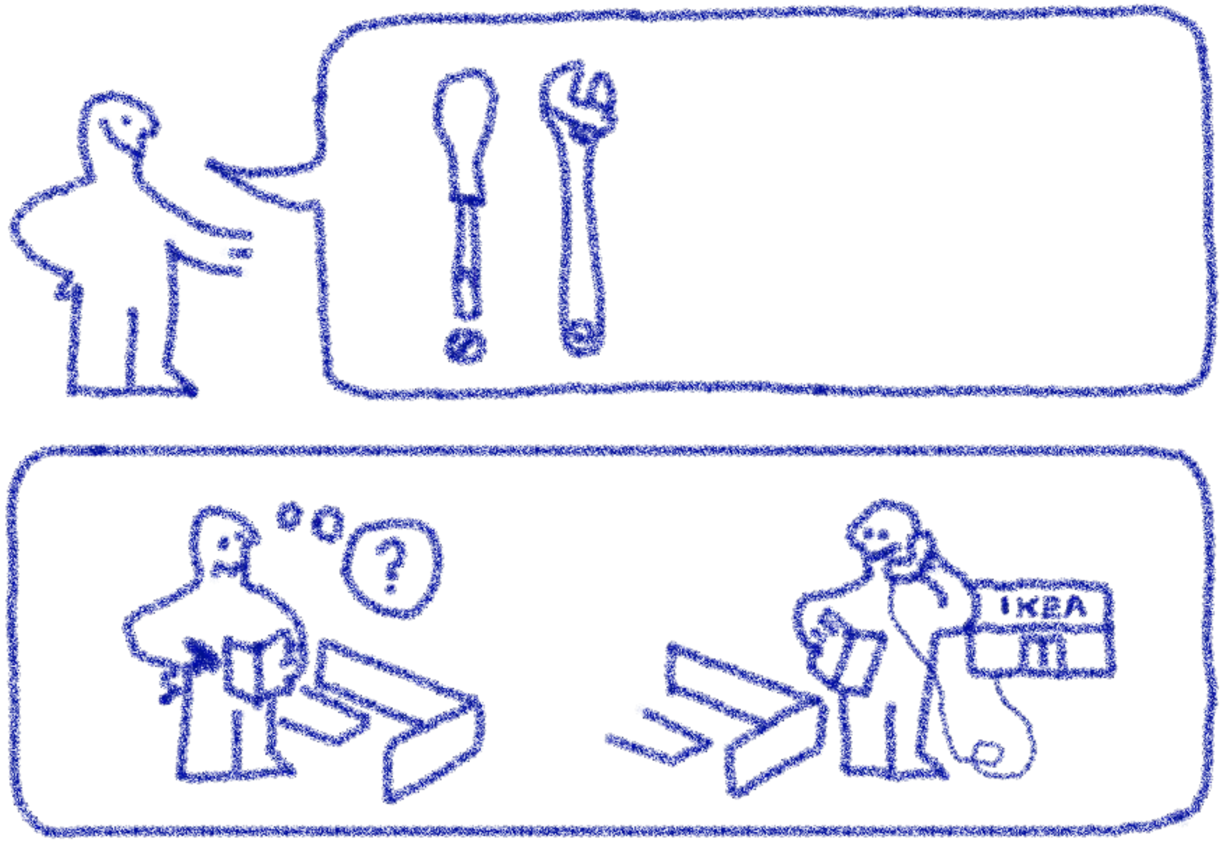






https://www.behance.net/gallery/7226653/Medium-Brand-Development

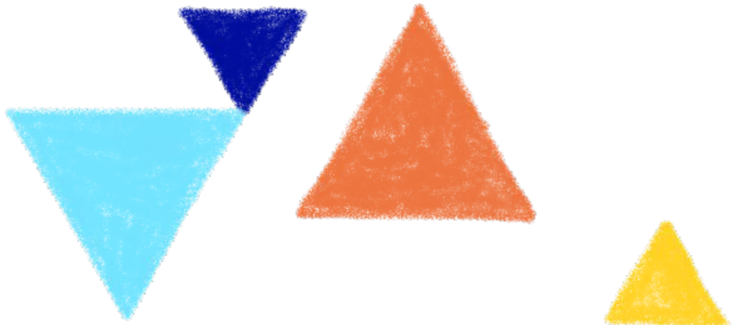
Style Guides alone are ambiguous



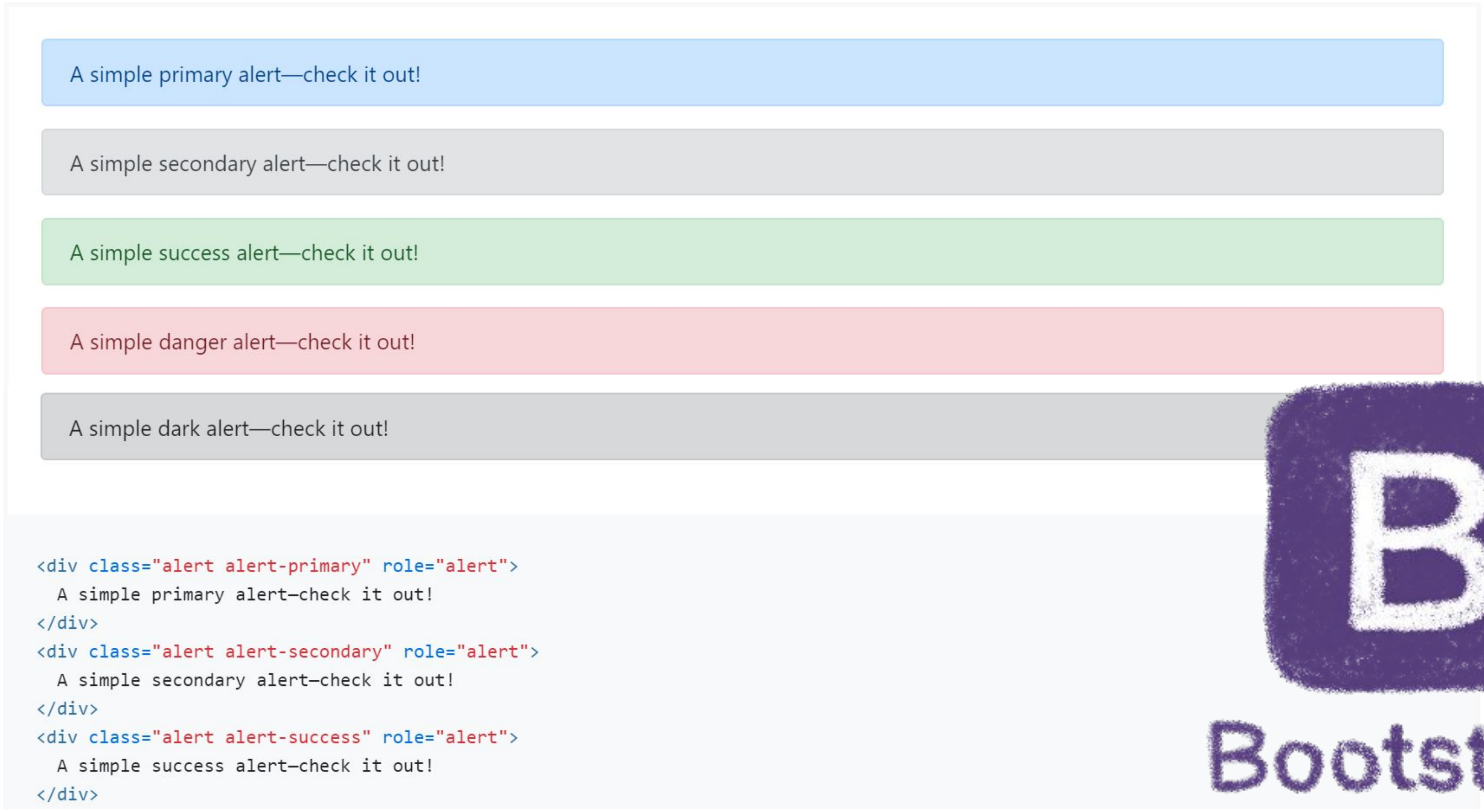
Interpretation needed to adapt the preconisation to the use case

Component Catalogs

A **component catalog** is a **repository** of components, with one or several **implementations**, code **examples** and **technical documentation**



Component Catalog example: Bootstrap



A simple primary alert—check it out!

A simple secondary alert—check it out!

A simple success alert—check it out!

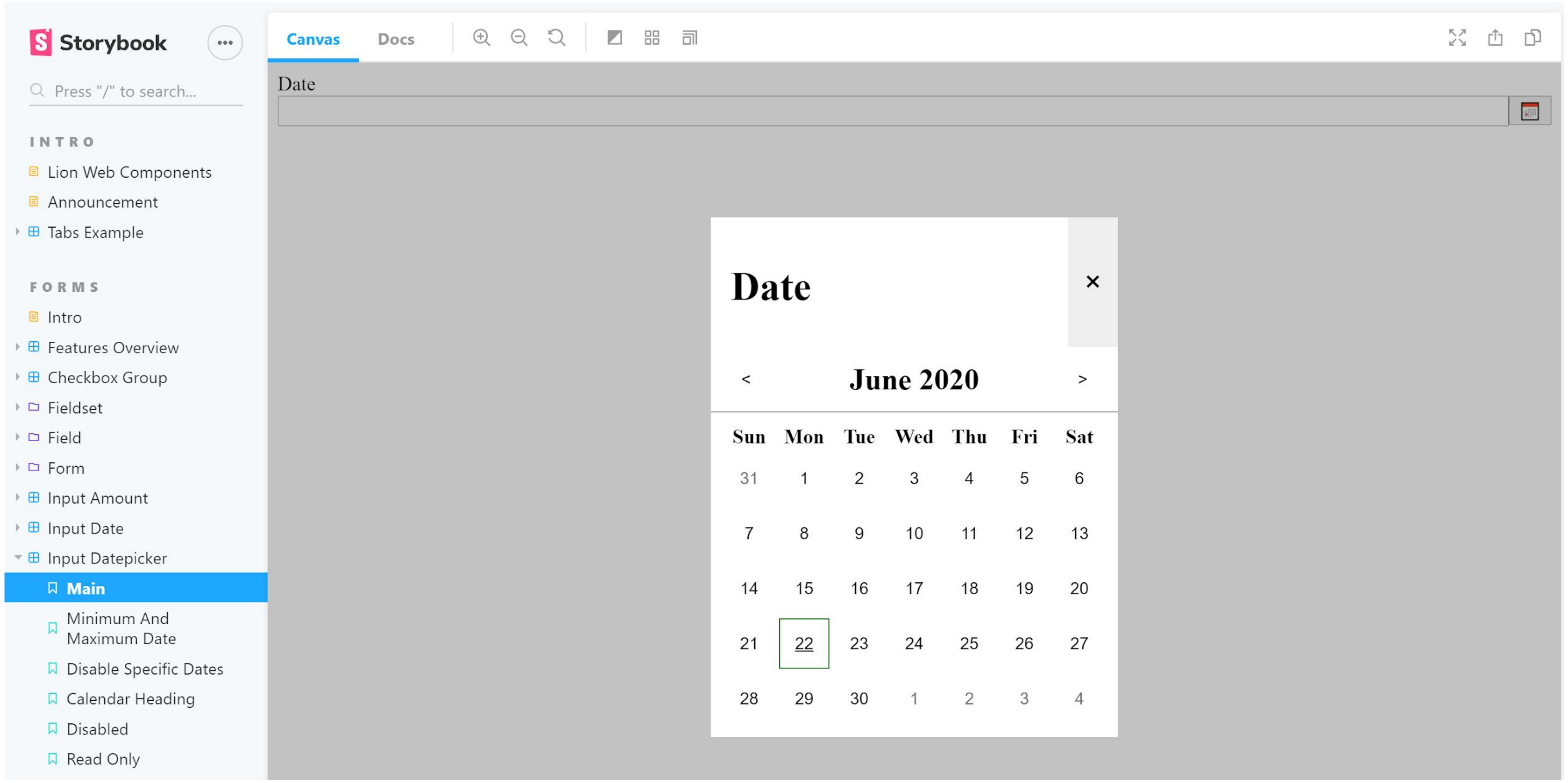
A simple danger alert—check it out!

A simple dark alert—check it out!

```
<div class="alert alert-primary" role="alert">
  A simple primary alert—check it out!
</div>
<div class="alert alert-secondary" role="alert">
  A simple secondary alert—check it out!
</div>
<div class="alert alert-success" role="alert">
  A simple success alert—check it out!
</div>
```

https://getbootstrap.com/

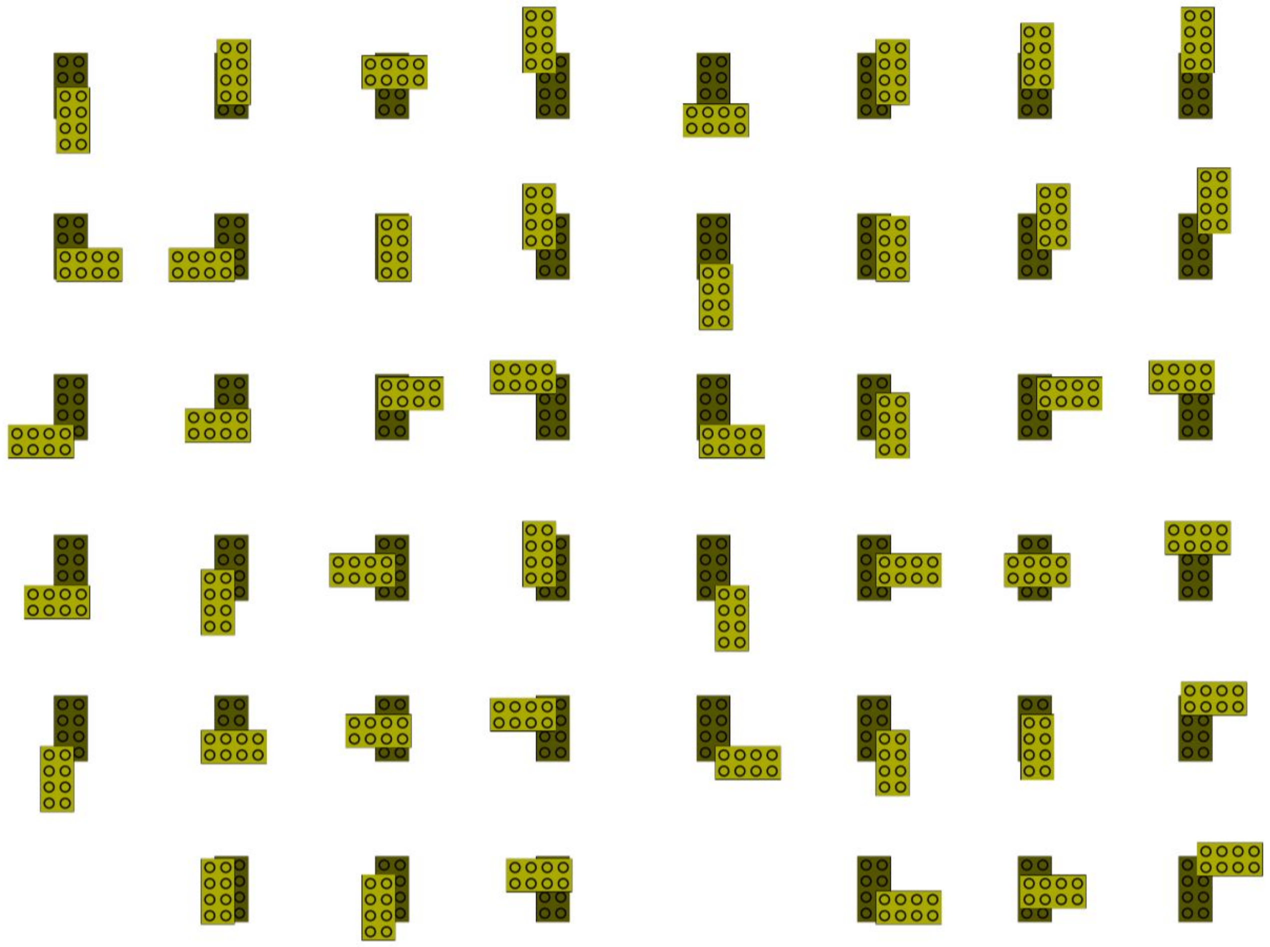
Component Catalog Example: ING's Lion



Sun	Mon	Tue	Wed	Thu	Fri	Sat
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4

https://lion-web-components.netlify.app/

Component Catalogs alone create inconsistency



Like using the same LEGO bricks to create very different objects

Design Systems

A Design System is like a **common visual language** for **product teams**

Design systems

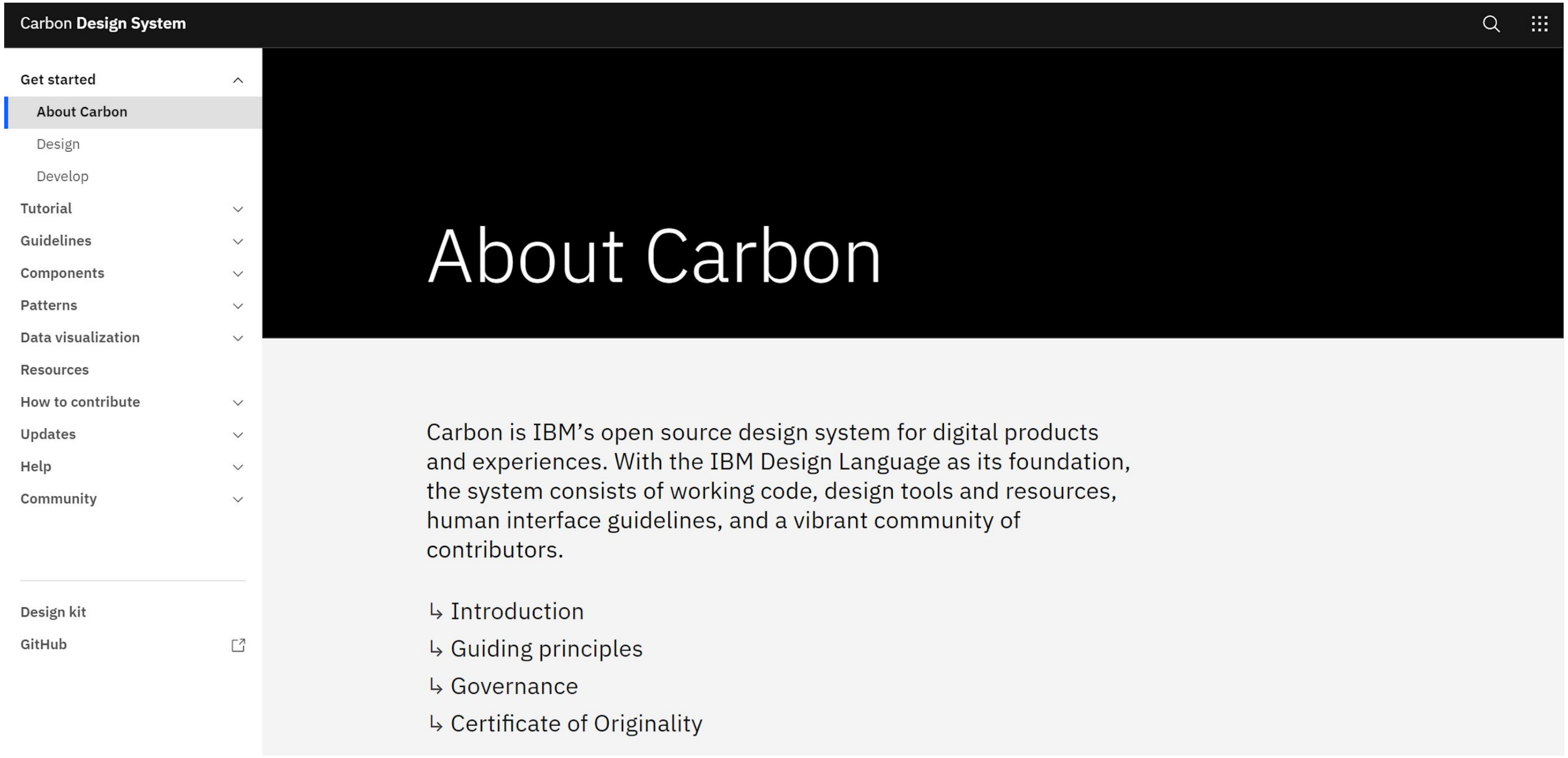
A Design System is a set of **design standards**, **documentations**, and **principles**, alongside with the toolkit (**UI patterns** and **code components**) to achieve those standards

Design systems

Design System ≈

Style Guide + Component Catalog

Example: Carbon Design System



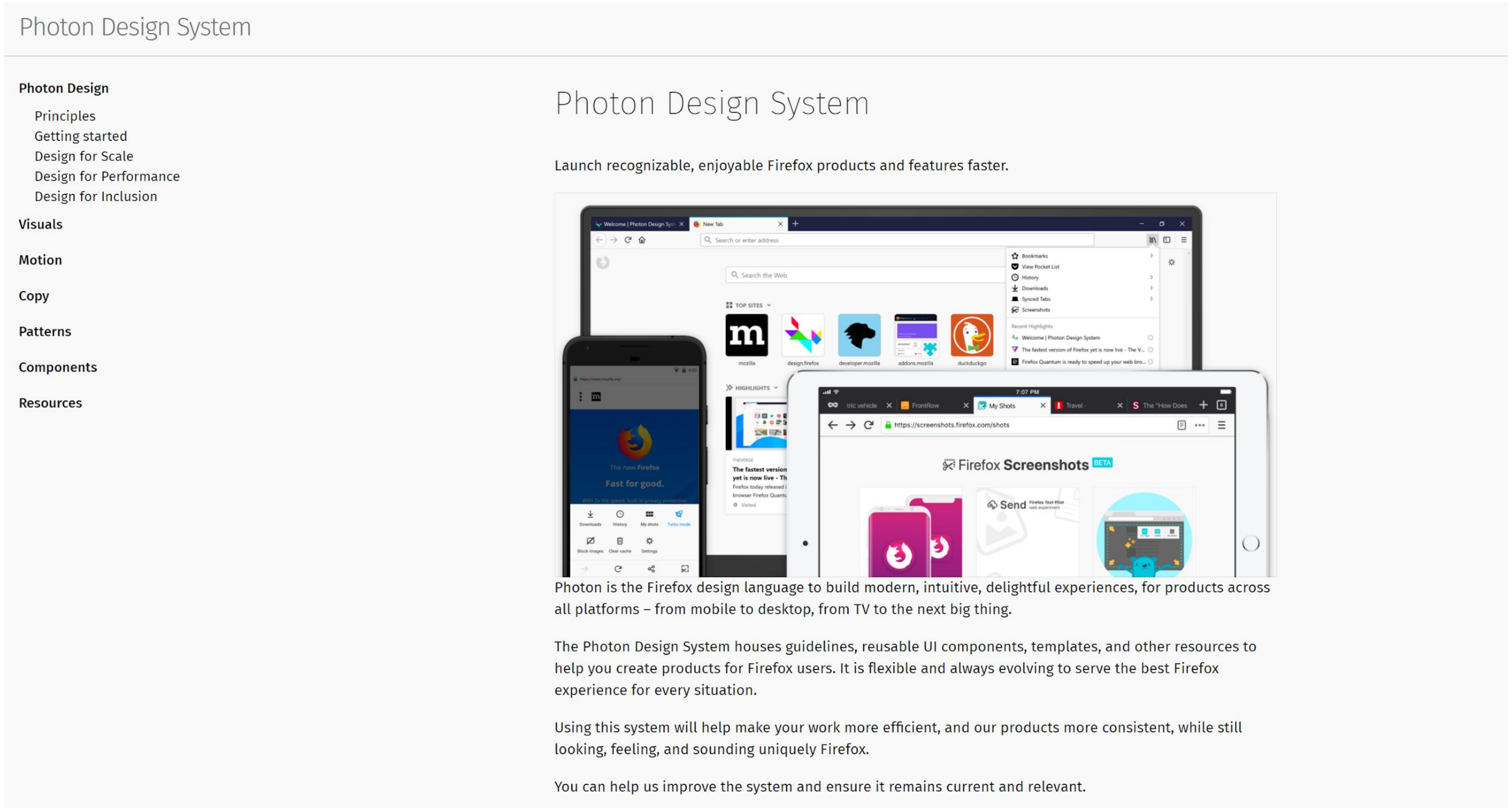
About Carbon

Carbon is IBM’s open source design system for digital products and experiences. With the IBM Design Language as its foundation, the system consists of working code, design tools and resources, human interface guidelines, and a vibrant community of contributors.

↳ Introduction
↳ Guiding principles
↳ Governance
↳ Certificate of Originality

https://www.carbondesignsystem.com/

Example: Firefox's Photon Design System



Photon Design System

Photon Design

- Principles
- Getting started
- Design for Scale
- Design for Performance
- Design for Inclusion

Visuals

Motion

Copy

Patterns

Components

Resources

Photon Design System

Launch recognizable, enjoyable Firefox products and features faster.

Photon is the Firefox design language to build modern, intuitive, delightful experiences, for products across all platforms – from mobile to desktop, from TV to the next big thing.

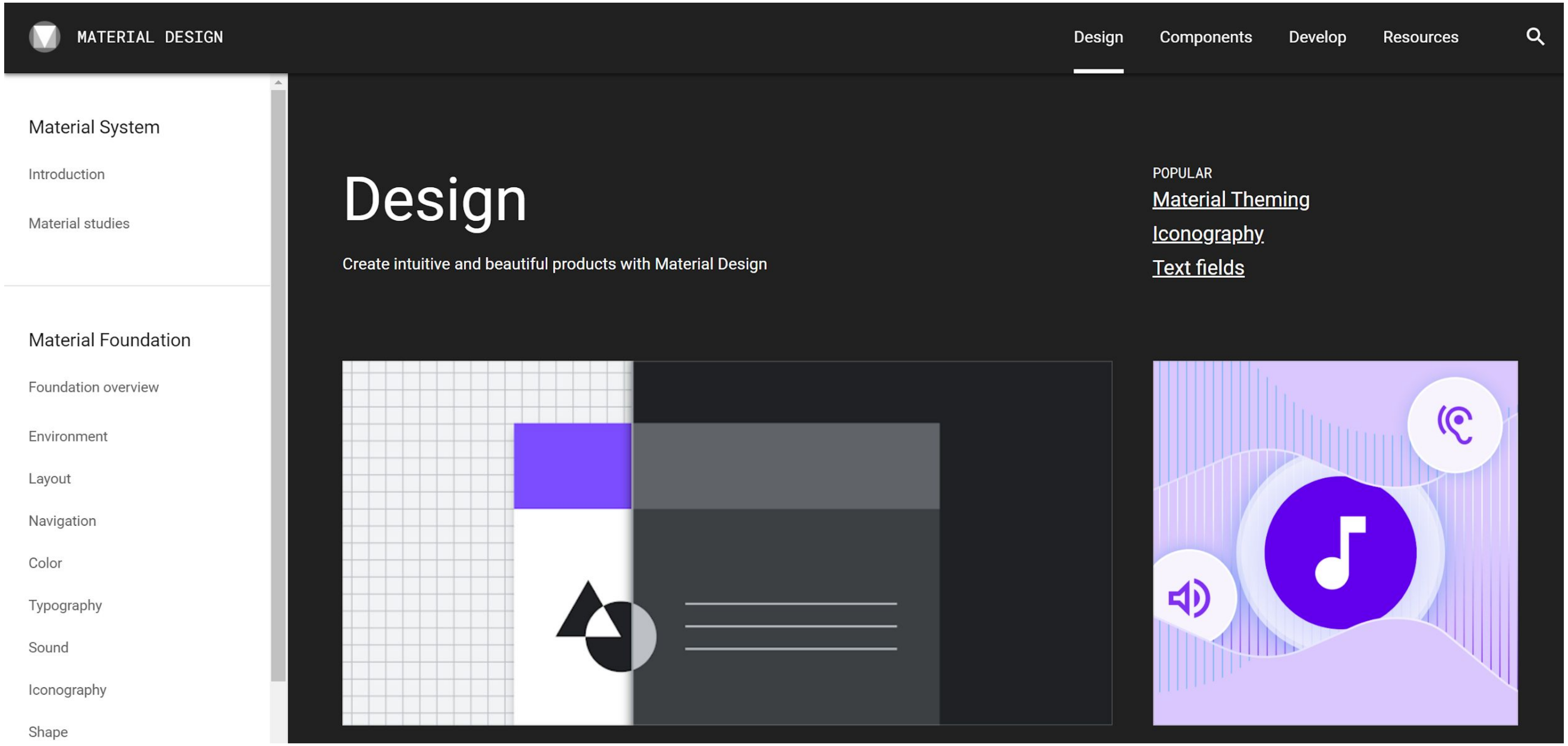
The Photon Design System houses guidelines, reusable UI components, templates, and other resources to help you create products for Firefox users. It is flexible and always evolving to serve the best Firefox experience for every situation.

Using this system will help make your work more efficient, and our products more consistent, while still looking, feeling, and sounding uniquely Firefox.

You can help us improve the system and ensure it remains current and relevant.

https://design.firefox.com/photon/

Example: Material Design



MATERIAL DESIGN

Material System

Introduction

Material studies

Material Foundation

Foundation overview

Environment

Layout

Navigation

Color

Typography

Sound

Iconography

Shape

Design

Create intuitive and beautiful products with Material Design

POPULAR

Material Theming

Iconography

Text fields

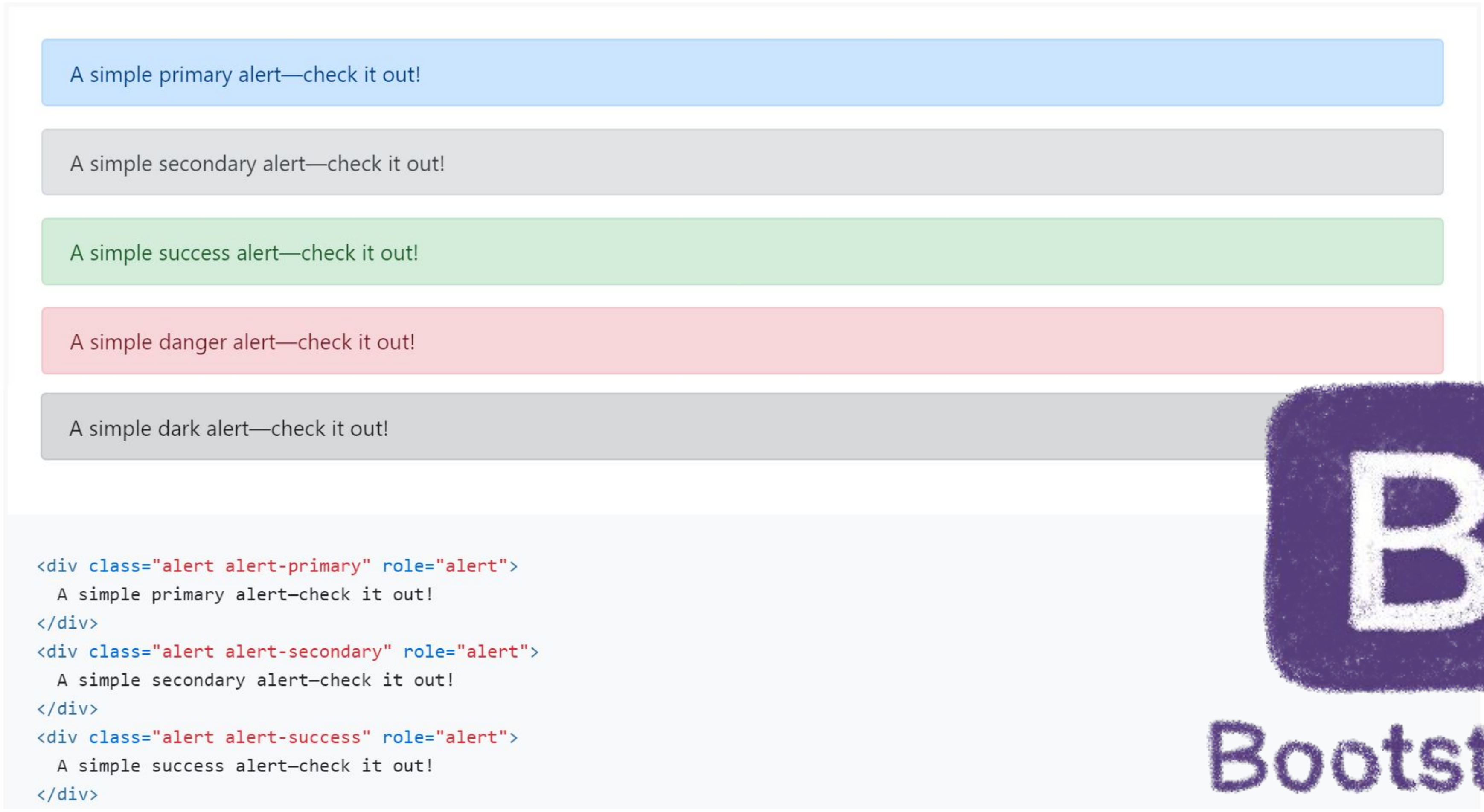
https://material.io/

Ask the Expert

The component catalog

The poor relative of the Design System family

Let's choose a simple example



A simple primary alert—check it out!

A simple secondary alert—check it out!

A simple success alert—check it out!

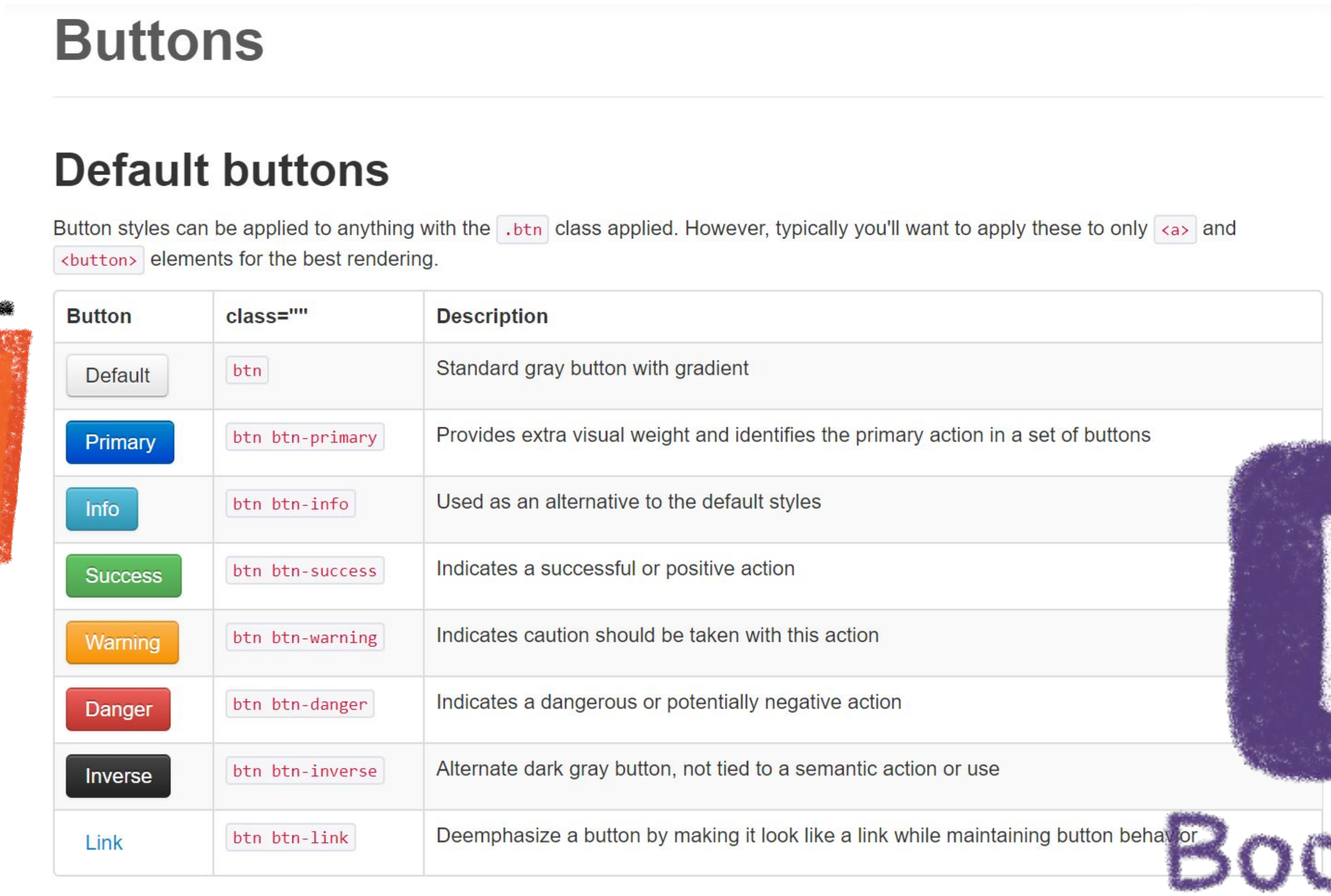
A simple danger alert—check it out!

A simple dark alert—check it out!

```
<div class="alert alert-primary" role="alert">
  A simple primary alert—check it out!
</div>
<div class="alert alert-secondary" role="alert">
  A simple secondary alert—check it out!
</div>
<div class="alert alert-success" role="alert">
  A simple success alert—check it out!
</div>
```

Bootstrap based component catalogs

A long time ago



Buttons

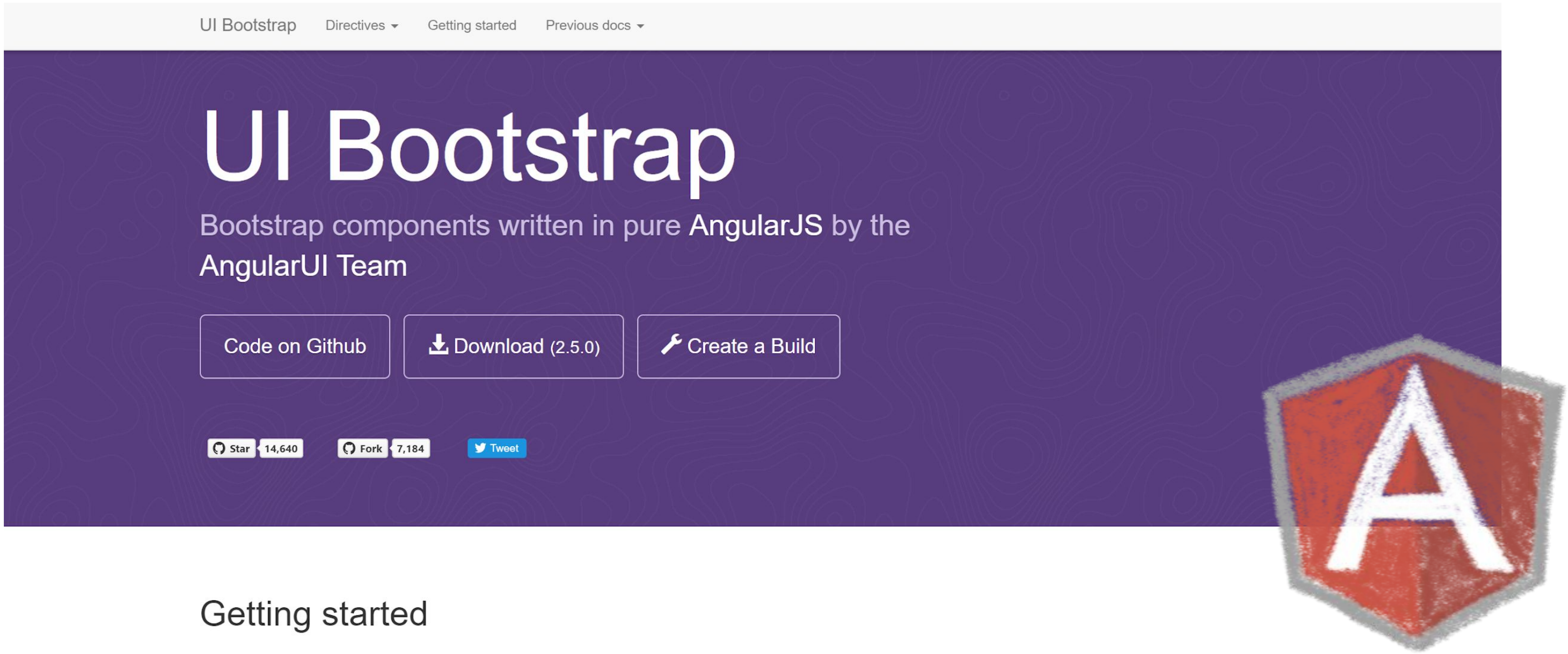
Default buttons

Button styles can be applied to anything with the `.btn` class applied. However, typically you'll want to apply these to only `<a>` and `<button>` elements for the best rendering.

Button	class=""	Description
Default	`btn`	Standard gray button with gradient
Primary	`btn btn-primary`	Provides extra visual weight and identifies the primary action in a set of buttons
Info	`btn btn-info`	Used as an alternative to the default styles
Success	`btn btn-success`	Indicates a successful or positive action
Warning	`btn btn-warning`	Indicates caution should be taken with this action
Danger	`btn btn-danger`	Indicates a dangerous or potentially negative action
Inverse	`btn btn-inverse`	Alternate dark gray button, not tied to a semantic action or use
Link	`btn btn-link`	Deemphasize a button by making it look like a link while maintaining button behavior

Components were defined in HTML, CSS and some jQuery

Then it was AngularJS time...


UI Bootstrap

Directives ▾ Getting started Previous docs ▾

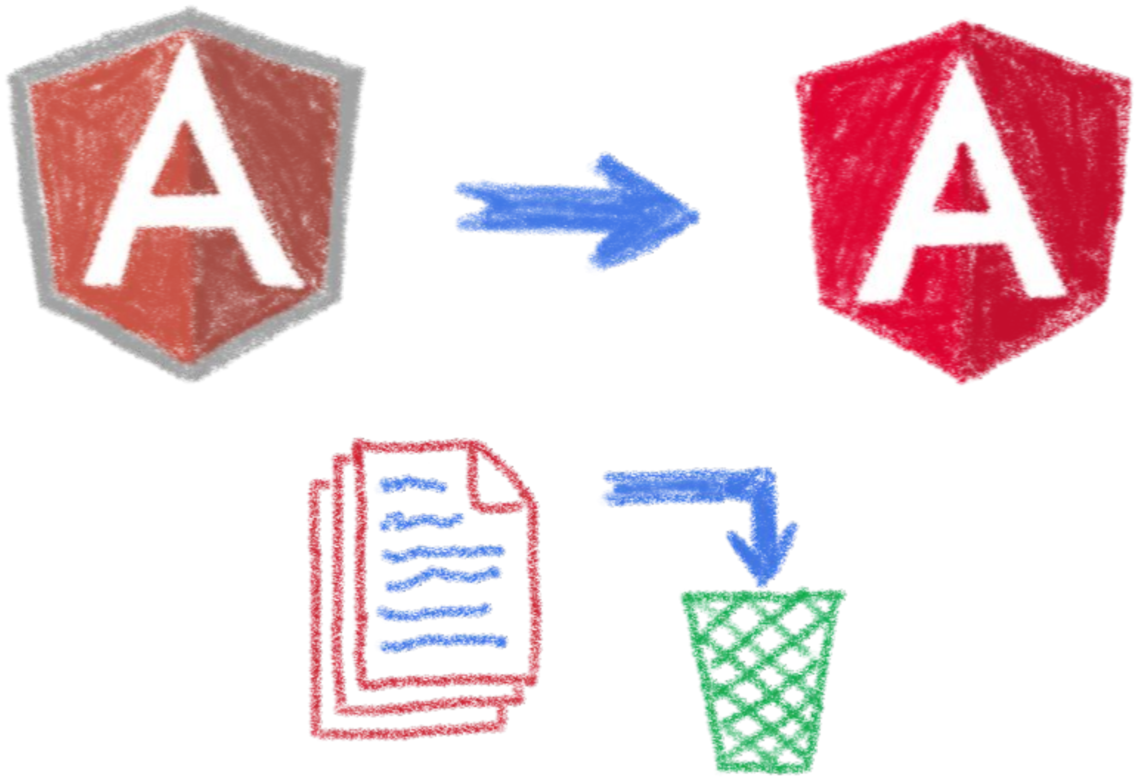
UI Bootstrap

Bootstrap components written in pure AngularJS by the AngularUI Team

Getting started

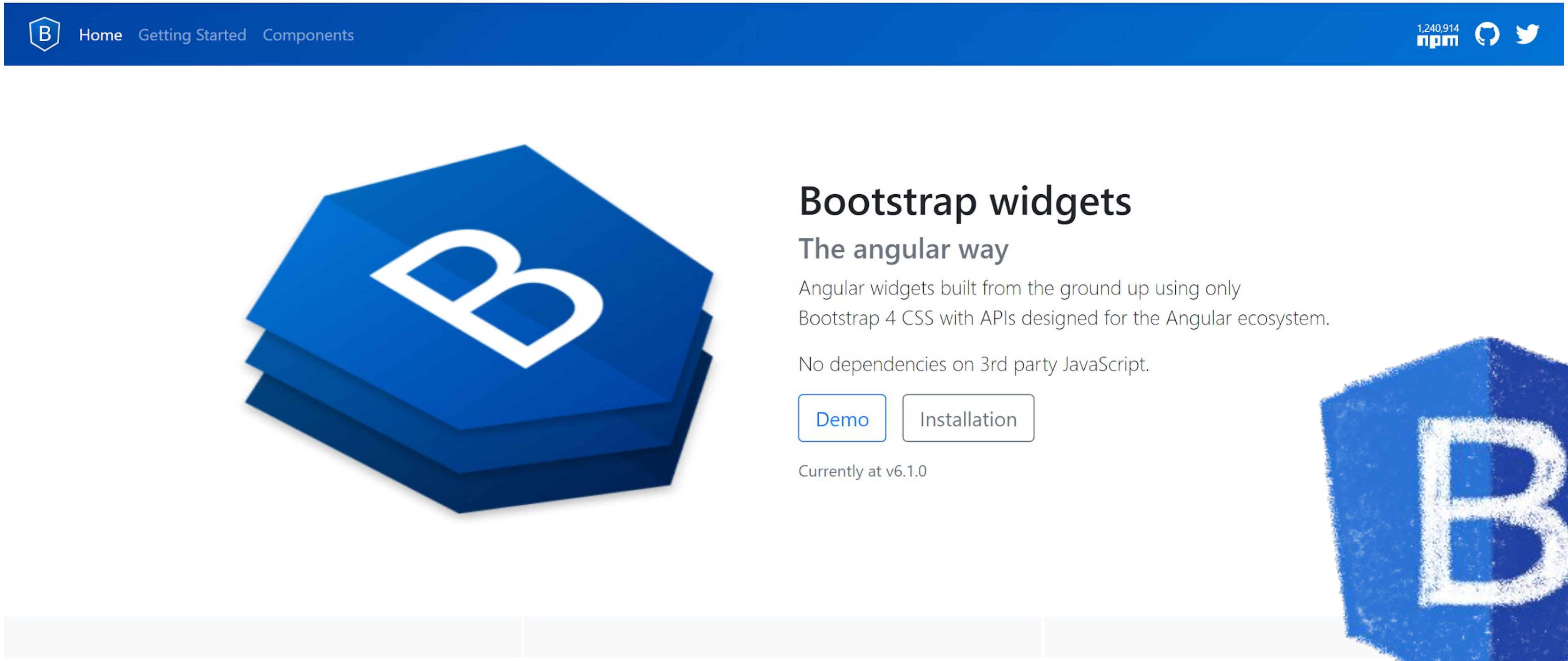
And new reference implementations were needed

But you know the sad story...



All UI Bootstrap based catalogs woke up with an obsolete implementation

Worry no more, let's do Angular!


Home Getting Started Components

Bootstrap widgets

The angular way

Angular widgets built from the ground up using only Bootstrap 4 CSS with APIs designed for the Angular ecosystem.

No dependencies on 3rd party JavaScript.

Demo Installation

Currently at v6.1.0

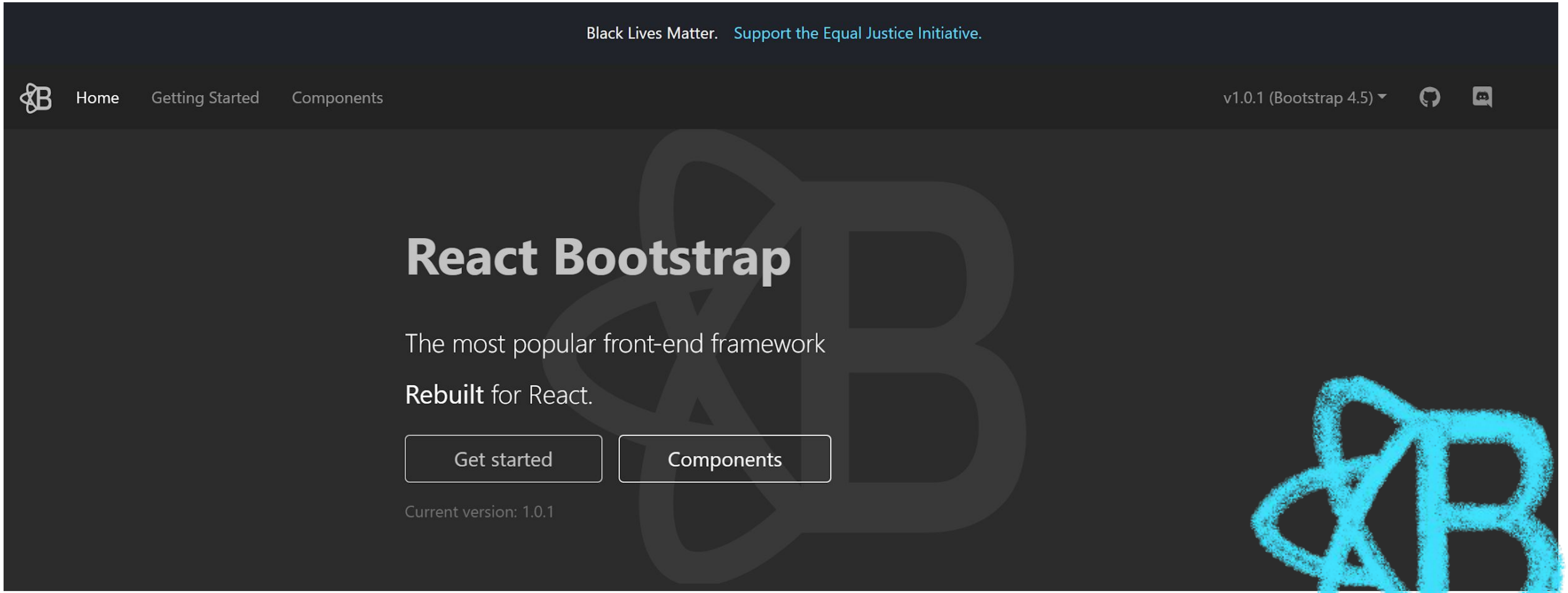
ng-bootstrap to the rescue

But times had changed...

In 2017 Angular is only one more in the clique

React is the new Big Thing™

Ask the Expert


Black Lives Matter. Support the Equal Justice Initiative.

React Bootstrap

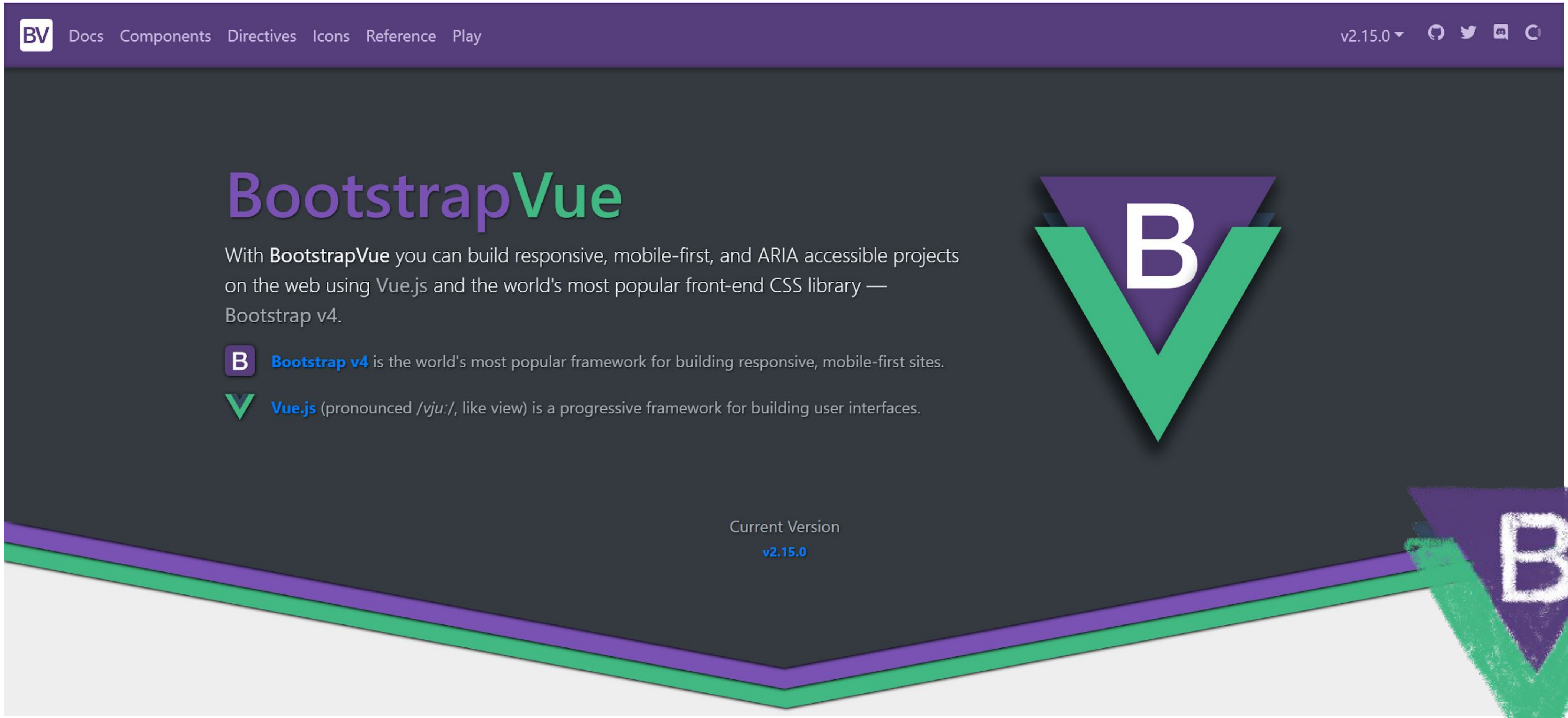
The most popular front-end framework

Rebuilt for React.

Current version: 1.0.1

So let's build React Bootstrap...

Wait, what about Vue?



BootstrapVue

With BootstrapVue you can build responsive, mobile-first, and ARIA accessible projects on the web using Vue.js and the world's most popular front-end CSS library — Bootstrap v4.

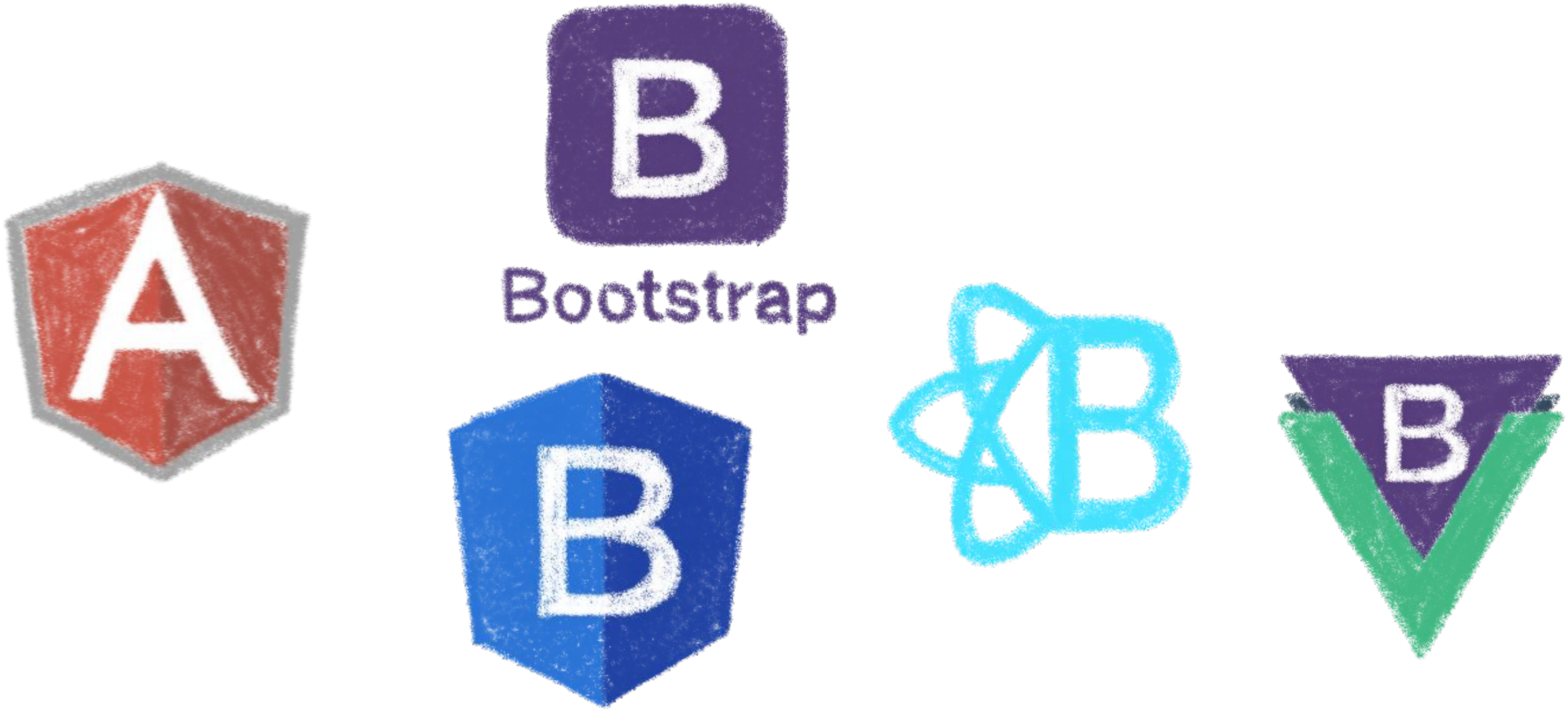
Bootstrap v4 is the world's most popular framework for building responsive, mobile-first sites.

Vue.js (pronounced /vjuː/, like view) is a progressive framework for building user interfaces.

Current Version
v2.15.0

We also need BootstrapVue

OK, I think you see my point...



Most Design System do a choice

Either they choose a by canonical implementation
or they ship and maintain several implementations

Both choices are problematic

Ask the Expert

Shipping only one implementation:

Web dev ecosystem changes quickly and almost nobody keeps the same framework for years...

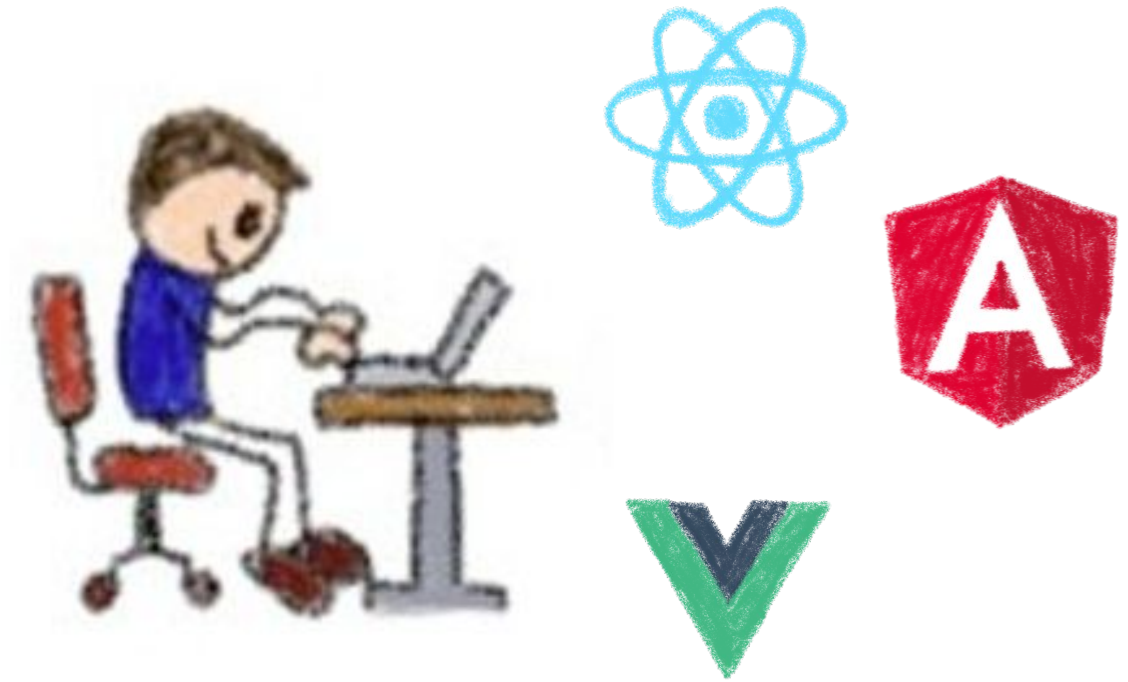
Both choices are problematic

Ask the Expert

Shipping several implementations:

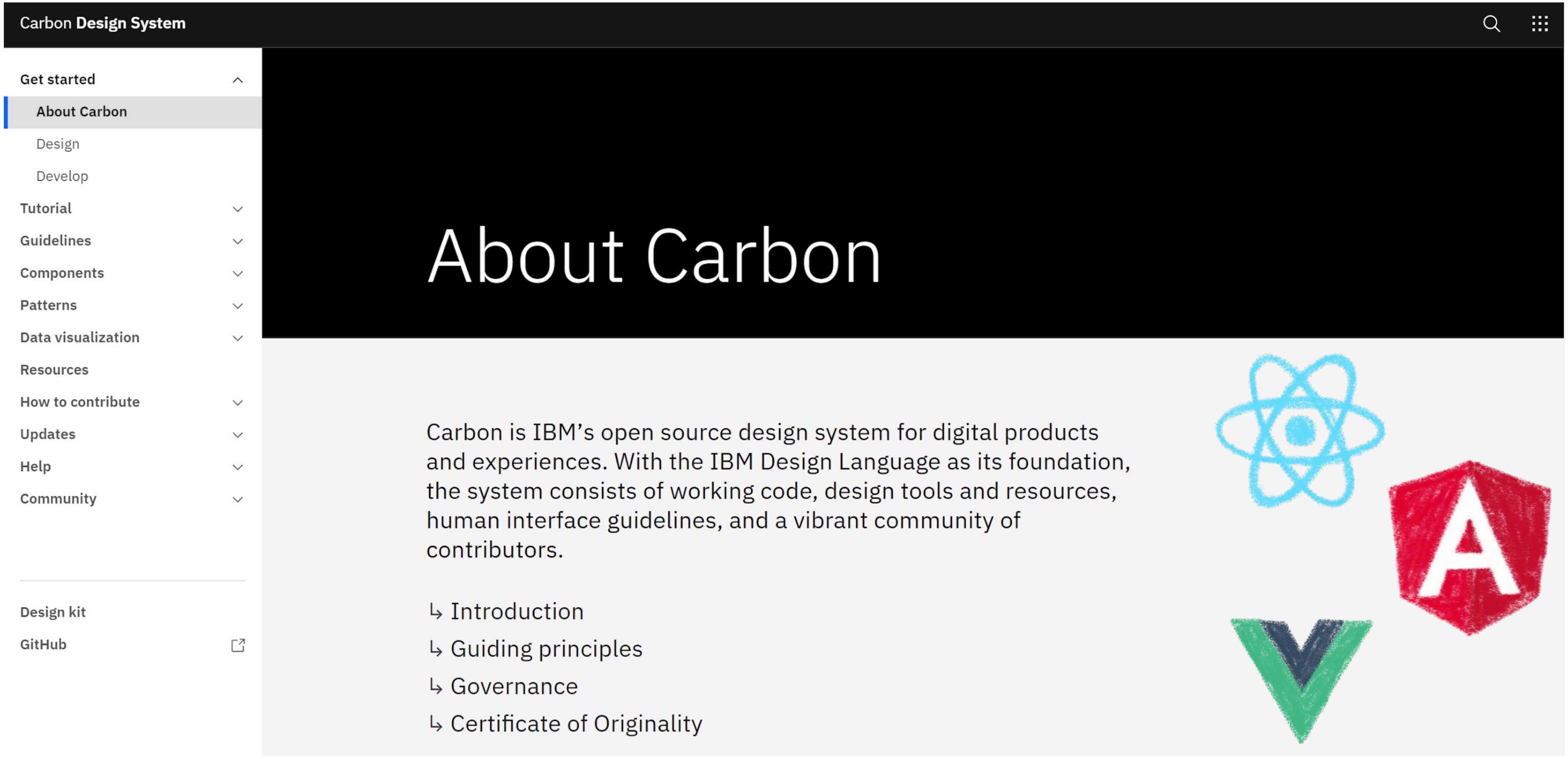
You need to maintain all the implementation…
and you still miss some others

Incomplete catalogs are problematic



People will need to recode the components in their chosen framework…
Coherence is not guaranteed!!!

Example: Carbon Design System



Carbon Design System

Get started

About Carbon

Design

Develop

Tutorial

Guidelines

Components

Patterns

Data visualization

Resources

How to contribute

Updates

Help

Community

Design kit

GitHub

About Carbon

Carbon is IBM's open source design system for digital products and experiences. With the IBM Design Language as its foundation, the system consists of working code, design tools and resources, human interface guidelines, and a vibrant community of contributors.

↳ Introduction

↳ Guiding principles

↳ Governance

↳ Certificate of Originality

Ask the Expert

The 3 minutes context

What the heck are web component?

Web Components

Web standard W3C

Web Components

Available in all modern browsers:

Firefox, Safari, Chrome

Web Components

Create your own HTML tags

Encapsulating look and behavior

Web Components

Fully interoperable

With other web components, with any framework

Web Components

CUSTOM ELEMENTS

SHADOW DOM

TEMPLATES

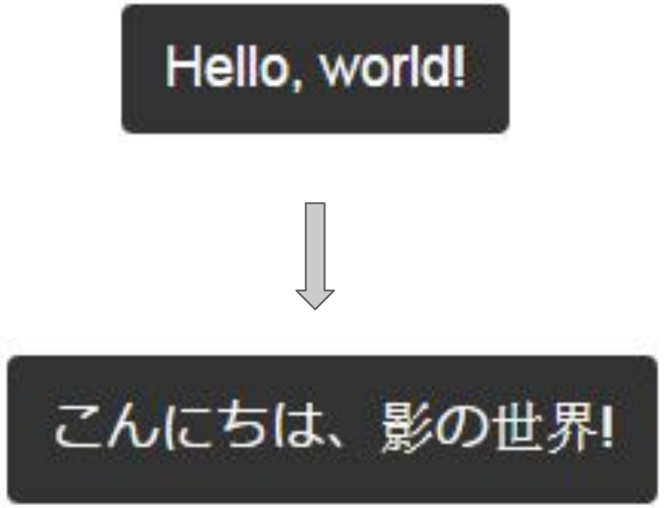
Custom Element

</> To define your own HTML tag

```
<body>
  ...
  <script>
    window.customElements.define('my-element',
      class extends HTMLElement {...});
  </script>
  <my-element></my-element>
</body>
```

Shadow DOM

To encapsulate subtree and style in an element



```
<button>Hello, world!</button>
<script>
var host = document.querySelector('button');
const shadowRoot = host.attachShadow({mode:'open'});
shadowRoot.textContent = 'こんにちは、影の世界!';
</script>
```

Template

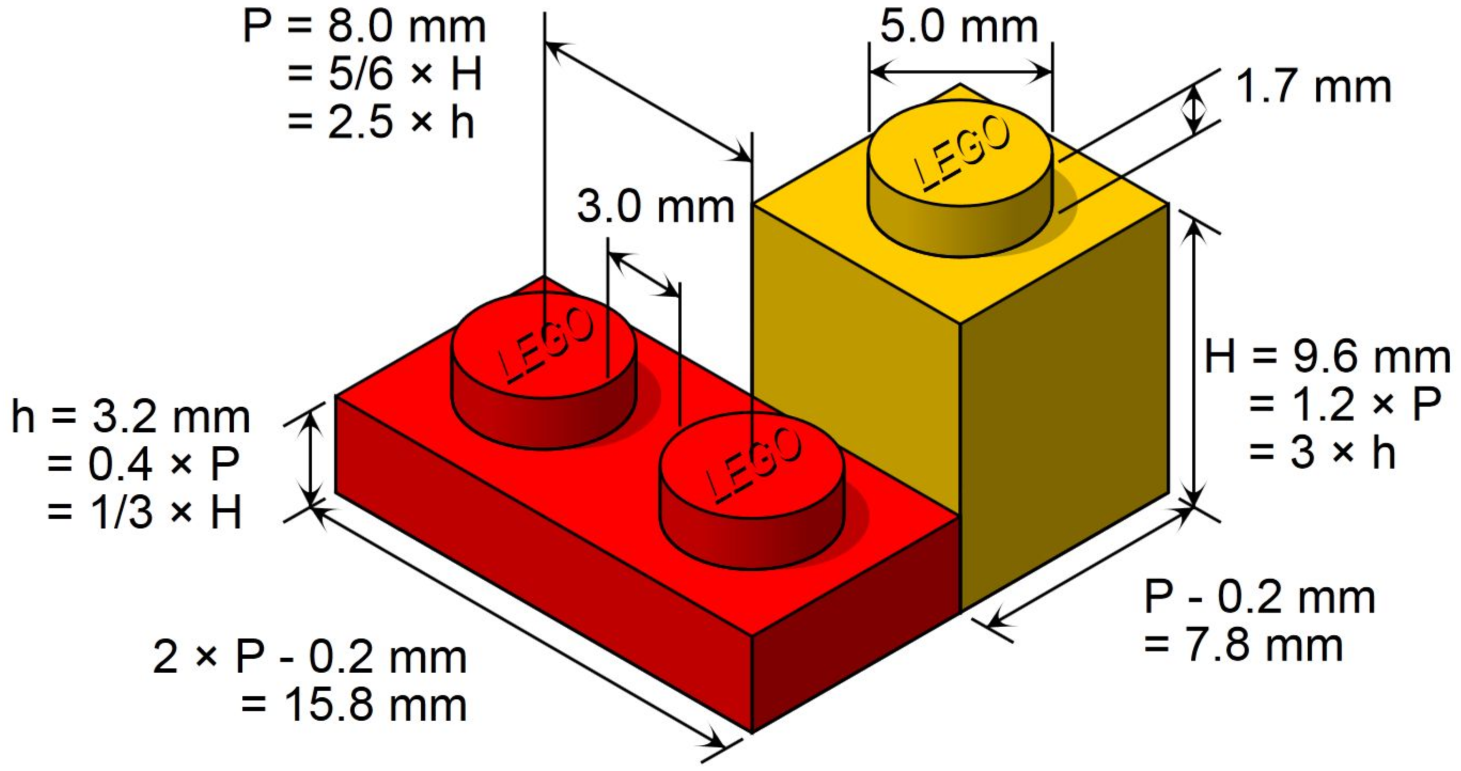
To have clonable document template

```
<template id="mytemplate">
  <img src="" alt="great image">
  <div class="comment"></div>
</template>

var t = document.querySelector('#mytemplate');
// Populate the src at runtime.
t.content.querySelector('img').src = 'logo.png';
var clone = document.importNode(t.content, true);
document.body.appendChild(clone);
```

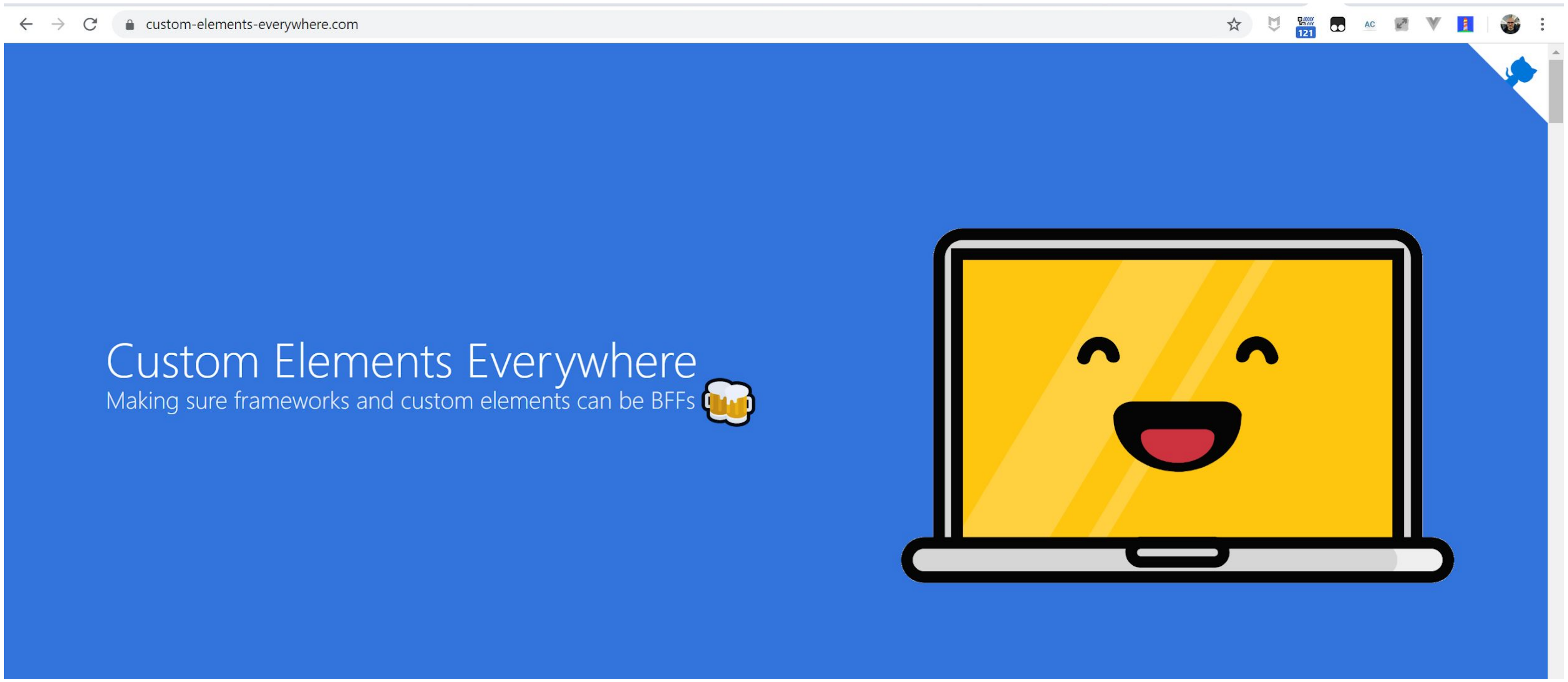
But in fact, it’s just an element…

Ask the Expert

- Attributes
- Properties
- Methods
- Events



Web Components are a web standard

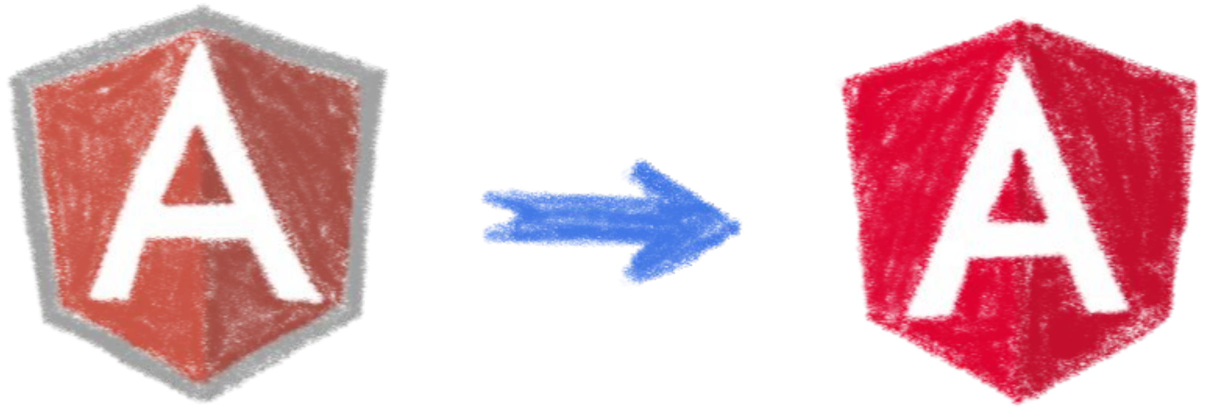

custom-elements-everywhere.com

Custom Elements Everywhere

Making sure frameworks and custom elements can be BFFs 🍻

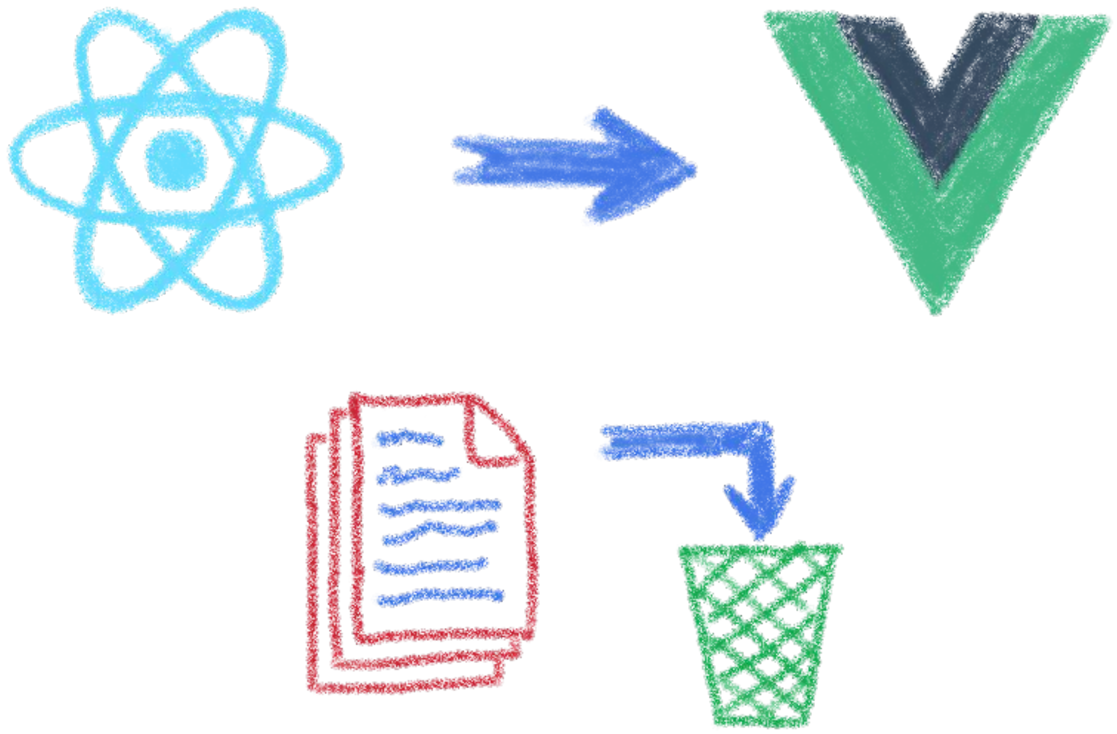
Web Components everywhere, baby!

Do you remember AngularJS?



And all the code put in the trash bin when Angular arrived...

The pain of switching frameworks?

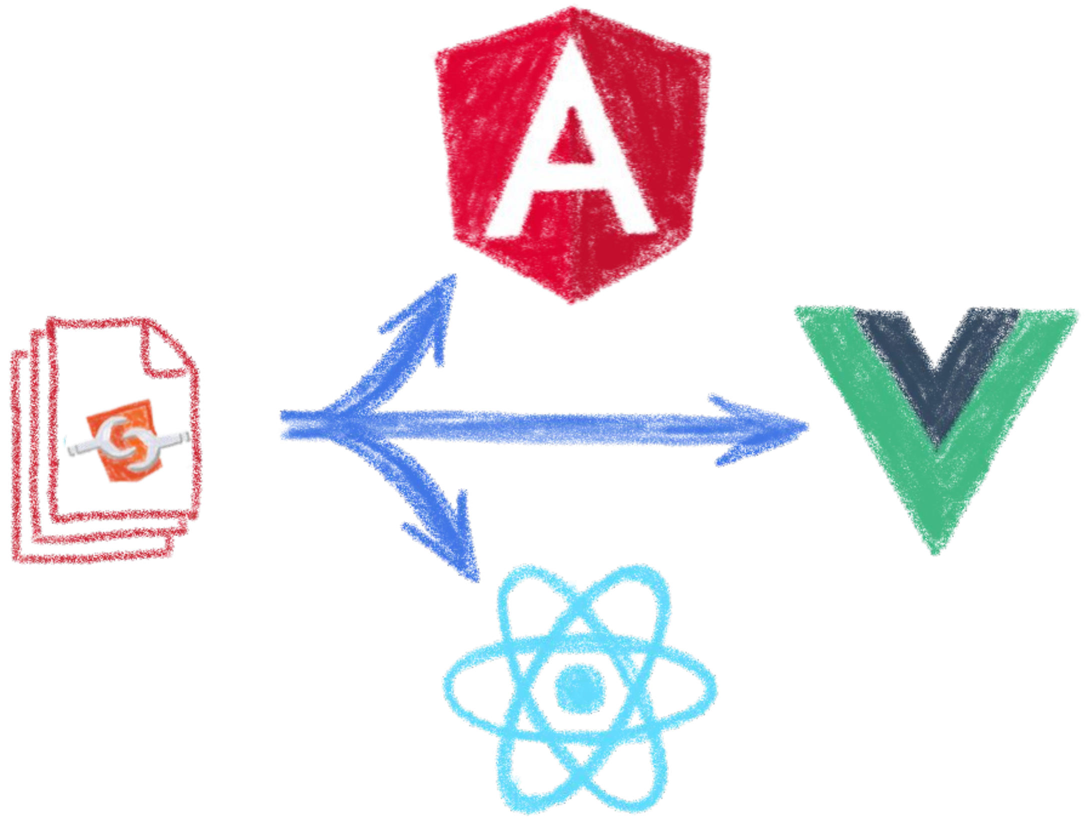


Rewriting once again your code...

The impossibility of sharing UI code?

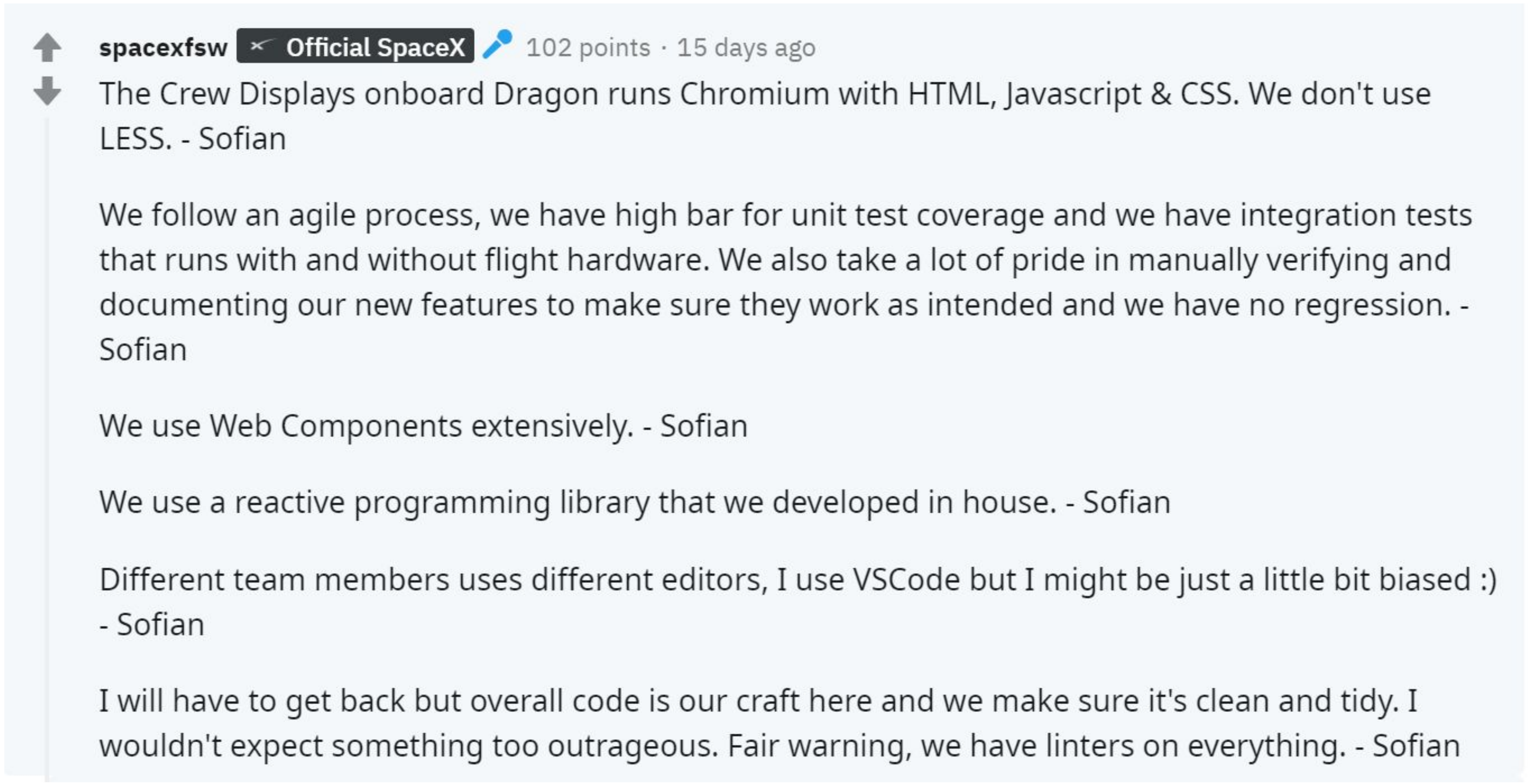
Between apps written with different frameworks

Web Components change that



In a clean and standard way

They are truly everywhere 🚀



spacexfsw Official SpaceX 102 points · 15 days ago

The Crew Displays onboard Dragon runs Chromium with HTML, Javascript & CSS. We don't use LESS. - Sofian

We follow an agile process, we have high bar for unit test coverage and we have integration tests that runs with and without flight hardware. We also take a lot of pride in manually verifying and documenting our new features to make sure they work as intended and we have no regression. - Sofian

We use Web Components extensively. - Sofian

We use a reactive programming library that we developed in house. - Sofian

Different team members uses different editors, I use VSCode but I might be just a little bit biased :) - Sofian

I will have to get back but overall code is our craft here and we make sure it's clean and tidy. I wouldn't expect something too outrageous. Fair warning, we have linters on everything. - Sofian

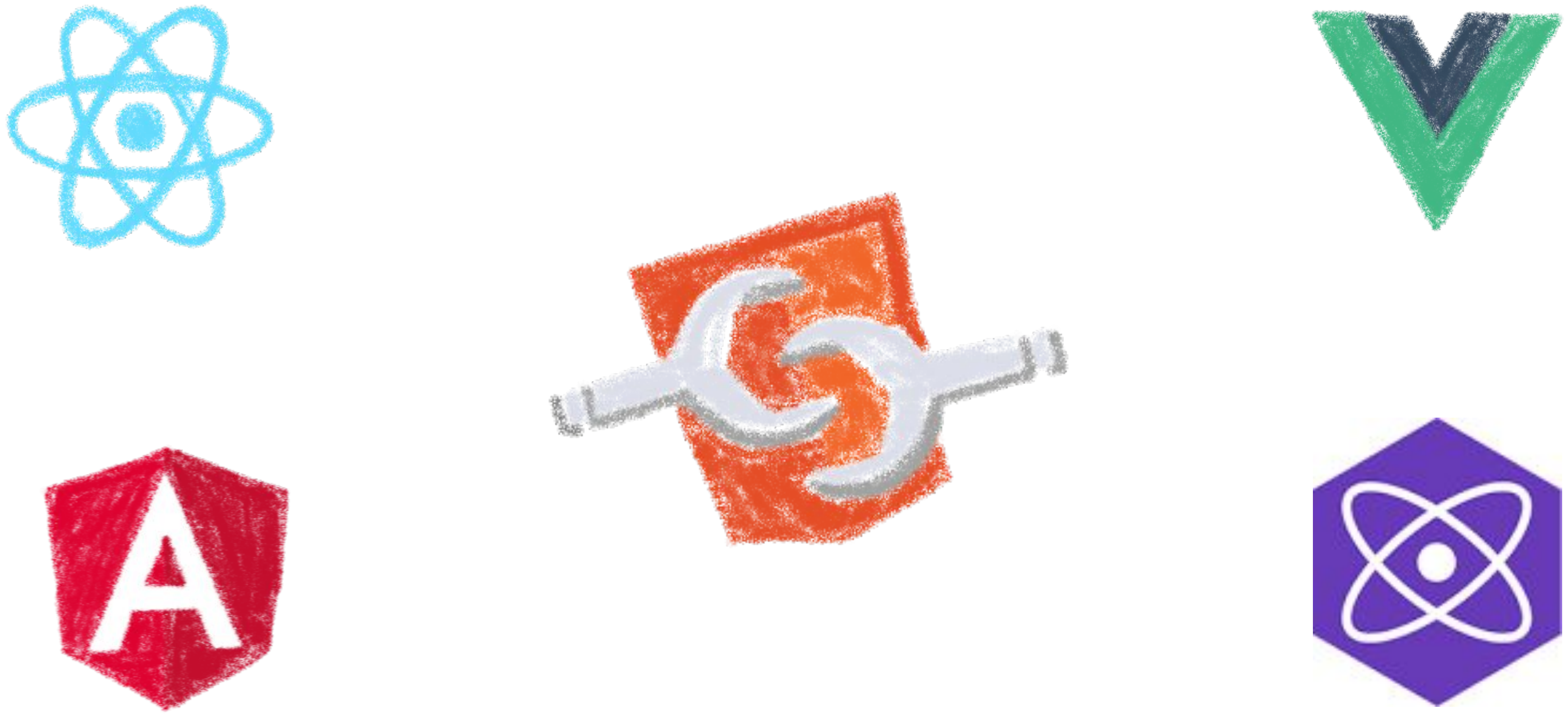
🚀 Even in the spaaaaaaaace 🚀

Ask the Expert

Web Components & Design Systems

A match made in heaven

You can have a single implementation



And it simply works everywhere

When you need interoperability

Nothing beats the standard

Ask the Expert

One more thing...*

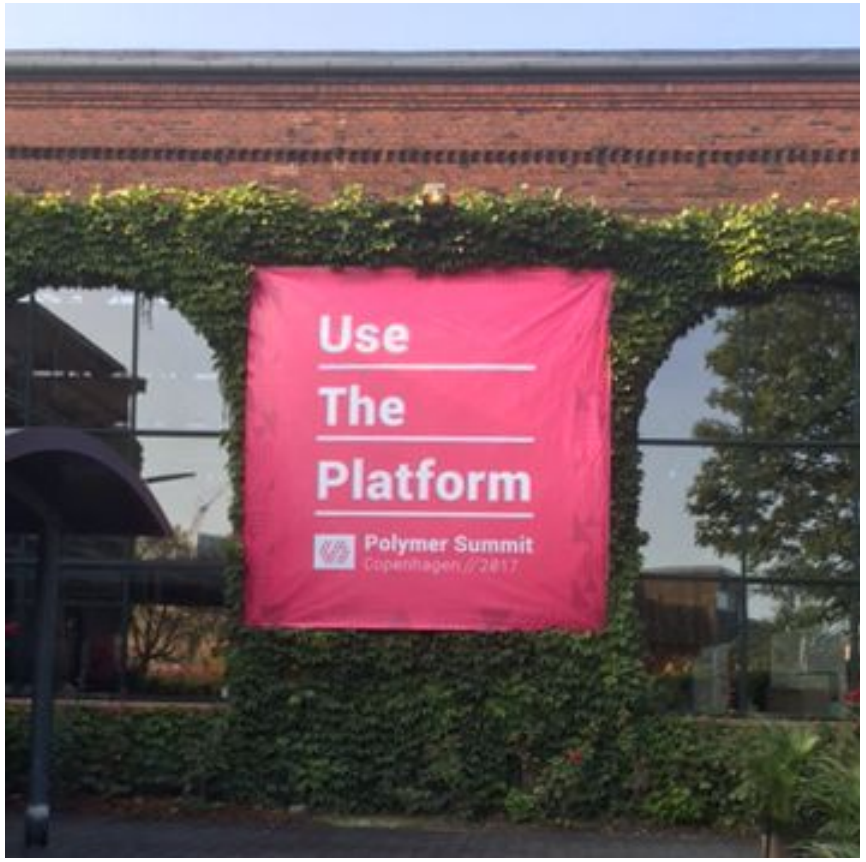
Let's copy from the master

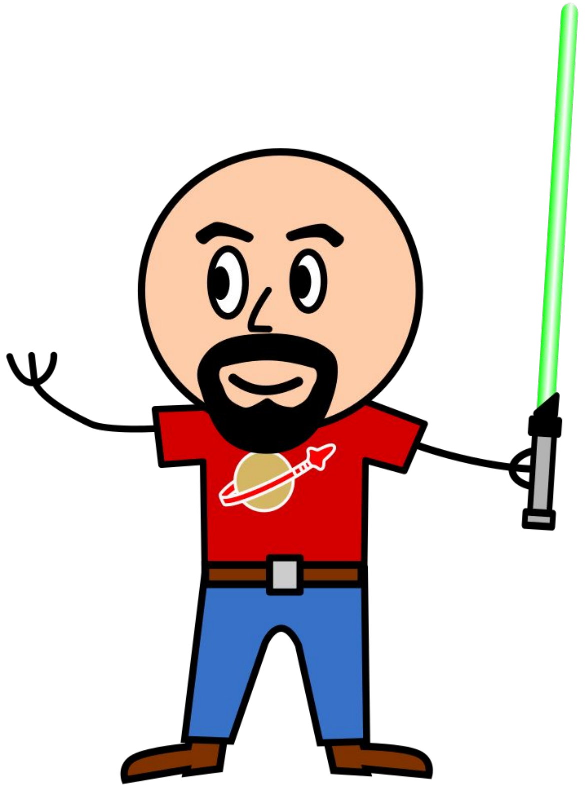
Stencil is not so important

WHEN THE SAGE POINTS AT THE MOON, THE FOOL LOOKS AT THE FINGER.
- BUDDHIST PROVERB
@AmericnDreaming
AmericanDreaming.deviantart.com

WebComponents ARE

Use the Platform, Luke...





WebComponents ARE native

Do you love your framework?

Oh yeah, we all do

Would you marry your framework?

Ask the Expert

Like until death…

How much does cost the divorce?

Ask the Expert

Do you remember when you dropped AngularJS for Angular?

Why recode everything again?

Ask the Expert

Reuse the bricks in your new framework

Lots of web components libraries

slim.js

LitElement

hybrids

snuggsi ツ

SkateJS

smart HTML ELEMENTS

stencil

And some good news

Angular Elements

Vue Web Component Wrapper

Frameworks begin to understand it

So for your next app

Choose a framework, no problem…

But please, help your future self

Use Web Components!

Ask the Expert

Conclusion

That's all, folks!