Style Guides with Catalog

Zürich ReactJS Meetup

April 27, 2016

Jeremy Stucki, Interactive Things
@herrstucki

Interactive Things

Design Studio

Digital Product Design

Est. 2010

interactivethings.com

@ixt

Catalog is a tool for creating style guides

What is Catalog?

- A tool for creating style guides
- A JavaScript library
- Standalone with no dependencies (works with any system!)
- Document anything: design, components, concepts
- React component playground
- Markdown-based

Style guide > Component playground

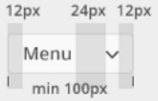
Style guides are for communication

A Style Guide Contains …

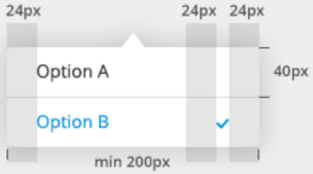
Design Specifications

Components
Menus

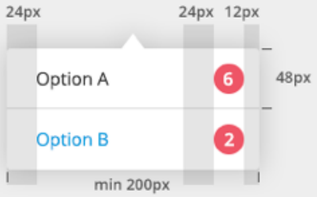
Dimensions



Drop Down Button
*Overruled when the width is based on the grid system

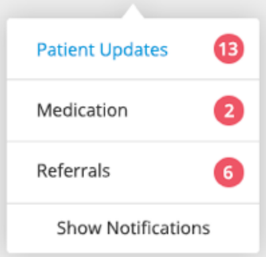


Drop Down Menu with Tick

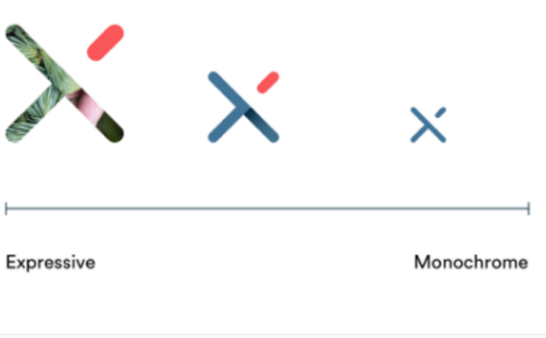


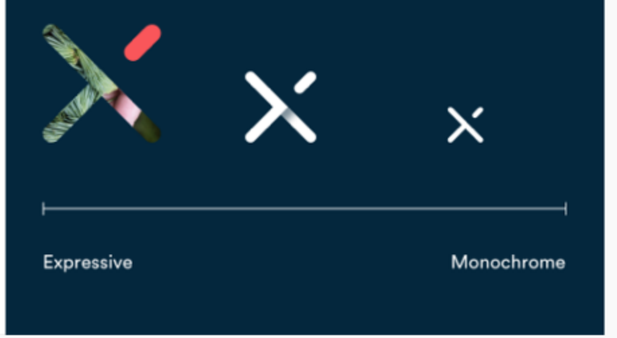
Drop Down Menu with Icon

Notification Menu



Instructions





Application

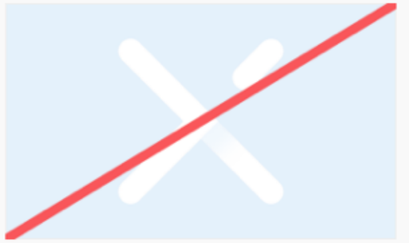
Do Expressive brandmark on plain background

Do Regular brandmark on a white background

Do Inverted brandmark on a dark background

Don't Don't recolor the brandmark

Don't Don't rotate, strech or distort the brandmark



Don't Never use the inverted brandmark on a bright background

Visual Variations

Open in new window Download

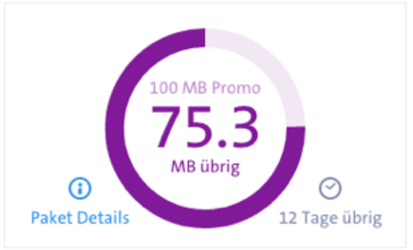
Data Package — Alert

Where proportional value displayed is lower than a certain predefined treshold. Relies on `.cp-large-package__core--alert` class to carry red colour of the donut chart.

Open in new window Download

Data Package — Promotion

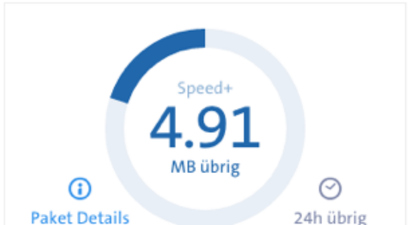
Relies on `.cp-large-package__core--promo` class to carry purple colour of the donut chart.



Open in new window Download

Data Speed Package

For data speed packages. Relies on `.cp-large-package__core--graded` class to carry animated navy/blue colour-coding of the donut chart.

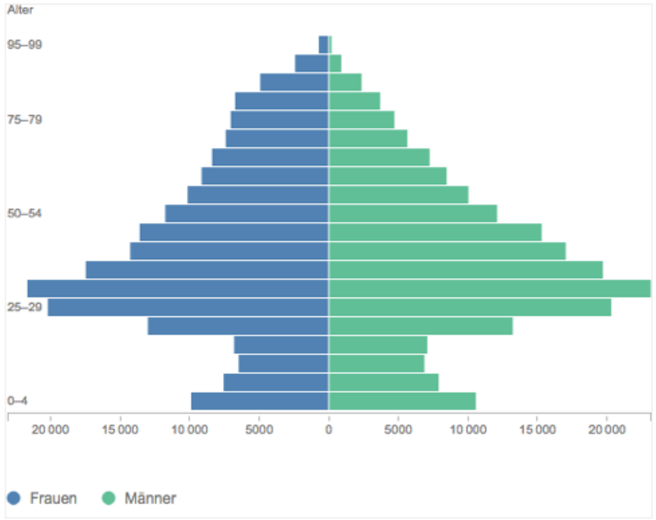


Interactive Documentation

Population Pyramid – Parametric Configuration

Configuration:

- `ageColumn` – Data column to use for the x-axis (required)
- `valueColumn` – Data column to use for the y-axis (required)
- `categoryColumn` – Data column to use for the categories (required)
- `leftCategory` – The category to use for the left side of the chart (required)
- `rightCategory` – The category to use for the right side of the chart (required)
- `groupSize` – The number of age categories to group into one bar (required)
- `xAxisLabel` – Label for the x-axis (optional)
- `yAxisLabel` – Label for the y-axis, shows in tooltip (required)
- `xTicks` – Number of ticks on the xAxis (optional)
- `yTicks` – Number of ticks on the yAxis (optional)
- `legendPadding` – Padding between legend and x-axis in pixels (required)



Open in new window Download index.html data.csv

Scatterplot – Parametric Configuration

- `xColumn` – Data column to use for the x-axis (required)
- `yColumn` – Data column to use for the y-axis (required)
- `cColumn` – Data column to use for the categories(required)
- `xAxisLabel` – Label for the x-axis, shows in tooltip (required)
- `yAxisLabel` – Label for the y-axis, shows in tooltip (required)

Thoughts

Notebook
Index

Catalyst 2.0 Lab Notes

This page is a record of research done, decisions made, and best practices that have arisen during Catalyst development.

Documentation Style

The component folder should contain a 'Docs.js' file which can be loaded by the Catalog documentation and displays the component in action. This file should also include info about where to use the component, and how to use its props. It should also include a download link to the `.sketch` file that contains the component's design template.

Stability Index

Mark's proposal

None of these categories comes with a backwards-compatibility support guarantee, since we reserve the ability to make changes to the components for future projects. In extreme cases this may involve changing the tech stack with which the components are built.

Solid: We know this component is good, it works in the wild, it's extensible and flexible. It has tests, it's accessible, the interaction works fast and intuitively on desktop and mobile. Essentially like "stable", within the context of our design and development goals.

Ok: This component is pretty good, it's going in the right direction, but it could be improved. Might be missing tests, might need documentation, might have an outstanding bug, might be a feature request which isn't satisfied yet.

Fresh: Just written, or freshly ported from an ongoing project. May be missing features, may not be ready for a desktop browser. (Remember, mobile-first!)

Testing

See also: Notes on testing frameworks and the Testing Survey.

List of requirements that a test solution must fulfill, in order of importance. It should:

- Be written in ES6 Javascript
- Be able to load and test javascript files written in ES6
- Be able to test React components are rendering and behaving correctly
- Enable component render visual testing

Component Playgrounds

Small Style

```
<Button small>Small Style</Button>
```

Large Style

```
<Button large>Large Style</Button>
```

Primary Style, for special buttons

Primary Style

```
<Button primary>Primary Style</Button>
```

Primary Hover Style

```
<Button primary hover>Primary Hover Style</Button>
```

Primary Focus Style

```
<Button primary focus>Primary Focus Style</Button>
```

Primary Active Style

```
<Button primary active>Primary Active Style</Button>
```

Primary Disabled Style

```
<Button primary disabled>Primary Disabled Style</But
```

Primary Small Style

```
<Button primary small>Primary Small Style</Button>
```

Primary Large Style

```
<Button primary large>Primary Large Style</Button>
```

Include Interesting or Iconic Children

+ Button With Icon

Examples

Real-World Style Guides

Demo

With Live Coding 😅

interactivethings.github.io/catalog

Further Reading

- Anna Debenham et al, Living Style Guides
- Bruce Hauman, Literate interactive coding: Devcards (2015)
- Brad Frost, Atomic Design (2013)
- Jina Bolton, Writing an Interface Style Guide (2008)

Thanks!

Jeremy Stucki
jeremy@interactivethings.com
github.com/herrstucki
@herrstucki