Where do you build a Design System?

About Design System Kit

Design Tools Meetup Hamburg, precious, 15.6.2018
We are,

**Marius Wilms**
CTO at Atelier Disko

**Christoph Labacher**
Interaction Designer at Atelier Disko
List

A List-component displays items in an ordered or unordered list.

Example

```html
<list>
  <listItem>Item 1</listItem>
  <listItem>Item 2</listItem>
  <listItem>Item 3</listItem>
</list>
```

Properties

<table>
<thead>
<tr>
<th>COMPONENT</th>
<th>PROPERTY</th>
<th>DESCRIPTION</th>
<th>DEFAULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>ListItem</td>
<td>Indicator</td>
<td>Which indicator to be displayed for the item</td>
<td>bullet</td>
</tr>
</tbody>
</table>
a workbench for collaboratively creating Design Systems
DSK
is short for
Design System Kit
- Open Source
- 4 weeks to 0.5
- 7 months to 1.0
- young
- in production
Our Talk

Not a sales pitch
Our Talk

— What are Design Systems?
— Where do you build a Design System?
— Core Aspects
— Technical Implementation
— The Design Definitions Tree
— Deployment Options
— Architecture & API
What are Design Systems?
What are Design Systems?

Design Systems are repositories of organizational knowledge on design.
Colors

Shopify uses colors purposefully to communicate how things function in the interface. This helps us create visual patterns that can make interacting with our product easier and more predictable for merchants.

Guidelines

These guidelines are the framework upon which we have built our system for how colors are used in Shopify.

Communication over Decoration

Although we value an aesthetically pleasing use of color, we place a higher value on clear communication. Our use of color should be purposeful, rational, and should serve to support the purpose of the content.

Color Should be Accessible

Our color choices should be made with consideration of merchants who are color blind or have a low visual acuity.
Components

IBM Carbon
Design Culture

Design Spotlight: Google highlights our Design Language System.

Building a Visual Language: Behind the scenes of our new design system.

What's in a Word?: Using location context to build better search tools for travel.

The Way We Build: How rethinking the Airbnb app changed the way we approach design.
Design Systems...

... help designers work faster

... improve consistency

... create a shared language for designers and developers
Where do you build a Design System?
Where do you build a Design System?

Blank Canvas  ?  Component Library
What we needed

A place for shared thinking
What we needed

A place that allows both structure and chaos
What we needed

A place for fragments and uncertainty
What we needed

Flexibility to suit a non-linear process
Core Aspects of DSK

— Direct Manipulation
— Trees
— Diverse Forms of Expression
— Open Development
List

A List-component displays items in an ordered or unordered list.

```xml
<List>
  <ListItem>Item 1</ListItem>
  <ListItem>Item 2</ListItem>
  <ListItem>Item 3</ListItem>
</List>
```

<table>
<thead>
<tr>
<th>Component</th>
<th>Property</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>ListItem</td>
<td>indicator</td>
<td>Which indicator is to be displayed for the item</td>
<td>bullet</td>
</tr>
</tbody>
</table>
Ziel: innerhalb der Gruppe Konsens über gemeinsame Themen und Elemente der für die Ziele des Designprojekts am wertvollsten sind.
Languages & Formats

- Backend
- Frontend
- Document
- Config

= <3
Languages & Formats

Go
+ JavaScript
+ Markdown
+ YAML
= <3
Languages & Formats

- Go
- ReasonML
- HTML
- JSON

= <3
Languages & Formats

- Backend: fixed
- Frontend: agnostic
- Document: agnostic
- Config: agnostic

= <3
Prioritize Ziel: innerhalb der Gruppe Konsens über gemeinsame Themen und Elemente der für die Ziele des Designprojekts am wertvollsten sind.

The Design Definitions Tree
The Design Definitions Tree

“Just a bunch of files and folders”
freedom in storage and editing

Natural usage of folders and files
means less to learn

Little/not DSK specific
makes sense on their own
Folders
Folders represent...

- components
- chapters
- design aspects
<table>
<thead>
<tr>
<th>Name</th>
<th>Änderungsdatum</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-About.md</td>
<td>08.05.2018, 11:11</td>
</tr>
<tr>
<td>02-API.md</td>
<td>08.05.2018, 11:11</td>
</tr>
<tr>
<td>03-Discussion.md</td>
<td>08.05.2018, 11:11</td>
</tr>
<tr>
<td>List Design (Proposal).sketch</td>
<td>08.05.2018, 11:11</td>
</tr>
<tr>
<td>List Design.sketch</td>
<td>08.05.2018, 11:11</td>
</tr>
<tr>
<td>List Styles.svg</td>
<td>08.05.2018, 11:11</td>
</tr>
<tr>
<td>List.png</td>
<td>08.05.2018, 11:11</td>
</tr>
<tr>
<td>meta.yaml</td>
<td>Heute, 17:04</td>
</tr>
<tr>
<td>Previous Design.png</td>
<td>08.05.2018, 11:11</td>
</tr>
</tbody>
</table>
# Visual Design

The following visual design has been agreed upon by our team:

![detail screen](List.png)

# Usage Rules

We use lists when we have to convey large amounts of information, but still keep it well structured. As they can easily grow very large, we try to keep the number of lists in our interfaces down to a minimum.

# Not to be confused with ...

We use both lists and [tables](/Components/Displaying Content/Table) display content in an orderly fashion, list are to be used for listed data, while table must only be used for tabular data. Duh.
...to explain

List

A List-component displays items in an ordered or unordered list.

Visual Design

The following visual design has been agreed upon by our team:

- UNORDERED LIST
  - List Item 1
  - List Item 2
  - List Item 3
  - List Item 4

- ORDERED LIST
  - List Item 1
  - List Item 2
  - List Item 3
  - List Item 4
# Example

```
<List>
  <ListItem>Item 1</ListItem>
  <ListItem>Item 2</ListItem>
  <ListItem>Item 3</ListItem>
</List>
```

# Properties

<table>
<thead>
<tr>
<th>Component</th>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ListItem</td>
<td><code>indicator</code></td>
<td>Which indicator is to be displayed for the item.</td>
</tr>
</tbody>
</table>
List

A List-component displays items in an ordered or unordered list.

Example

```html
<List>
  <ListItem>Item 1</ListItem>
  <ListItem>Item 2</ListItem>
  <ListItem>Item 3</ListItem>
</List>```

Properties
<table>
<thead>
<tr>
<th>Name</th>
<th>Änderungsdatum</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-About.md</td>
<td>08.05.2018, 11:00</td>
</tr>
<tr>
<td>02-API.md</td>
<td>08.05.2018, 11:00</td>
</tr>
<tr>
<td>03-Discussion.md</td>
<td>08.05.2018, 11:00</td>
</tr>
<tr>
<td>List Design (Proposal).sketch</td>
<td>08.05.2018, 11:00</td>
</tr>
<tr>
<td>List Design.sketch</td>
<td>08.05.2018, 11:00</td>
</tr>
<tr>
<td>List Styles.svg</td>
<td>08.05.2018, 11:00</td>
</tr>
<tr>
<td>List.png</td>
<td>08.05.2018, 11:00</td>
</tr>
<tr>
<td>meta.yaml</td>
<td>Heute, 17:04</td>
</tr>
<tr>
<td>Previous Design.png</td>
<td>08.05.2018, 11:00</td>
</tr>
</tbody>
</table>
A List-component displays items in an ordered or unordered list.

tags:
- Progress/Draft
- Release/0.6

version: 1
authors:
  - barb@hooli.com

description: >
  A List-component displays items in an ordered or unordered list.

tags:
  - Progress/Draft
  - Release/0.6

version: 1
# AUTHORS.txt

Barbara M. Grayson <barb@hooli.com>
Melisa Mosher <melisa@acme.io>
Randall Hyman <randall@evilcorp.org>
Assets
Nesting
Starting DSK

Last login: Mon Jun 11 18:28:50 on ttys000
/Users/christophlabacher/go/src/github.com/atelierdisko/dsk/Desire
~/ /Users/christophlabacher/go/src/github.com/atelierdisko/dsk/Design

DSK
Version head-549927b

Starting message broker...
Detecting tree root...
Tree root found: /Users/christophlabacher/go/src/github.com/atelierdisko/dsk/Desire
Begin watching tree for changes...
Opening tree...
Synced tree with 27 total node/s in 20.416925ms
Mounting APIv1...
Mounting frontend...
Starting web interface on 127.0.0.1:8080....

Please visit: http://127.0.0.1:8080
Hit Ctrl+C to quit
List

A List-component displays items in an ordered or unordered list.

Visual Design

The following visual design has been agreed upon by our team:

**UNORDERED LIST**
- List item 1
- List item 2
- List item 3
- List item 4

**ORDERED LIST**
- List item 1
- List item 2
- List item 3
Deployments
Getting it out there
Getting it to the user

App = Dropbox
Content = Dropbox
Getting it to the user

App = Dropbox
Content = Git
Getting it to the user

App = Server
Content = Git
Architecture
Architecture

- Custom Frontend
- Internal Tools
- Bundled Frontend

API

DSK

Design Definitions Tree

Git
Prioritize Ziel: innerhalb der Gruppe Konsens über gemeinsame Themen und Elemente der für die Ziele des Designprojekts am wertvollsten sind.
API Endpoints

/hello
/tree
/tree/{path}
/tree/{path}/{asset}
/search?q={query}
/messages
API Endpoints

/hello (version and greeting)
/tree (full tree)
/tree/{path} (single aspect)
/tree/{path}/{asset} (aspect asset)
/search?q={query} (fuzzy search)
/messages (websocket)
API Endpoints (versioned)

/api/v1/hello
/api/v1/tree
/api/v1/tree/{path}
/api/v1/tree/{path}/{asset}
/api/v1/search?q={query}
/api/v1/messages
{
  "hash": "d3f81…",
  "url": "Components/Displaying-Content/List",
  "title": "List",
  "description": "A List-component displays…",
  "authors": [
    {
      "email": "barb@hooli.com",
      "name": "Barbara M. Grayson"
    }
  ],
  "modified": 1528721196,
  "version": "1",
  "tags": [
    "Progress/Draft",
    "Release/0.6"
  ]
}
"docs": [
  {
    "title": "About",
    "html": "<h1>Visual Design</h1>…",
    "raw": "# Visual Design…"
  },
  {
    "title": "API",
    "html": "<h1>Example</h1>…",
    "raw": "# Example…"
  }
],
"downloads": [
  {
    "url": "Components/Displaying-Content/List/List Design.sketch",
    "name": "List Design.sketch"
  },
  ...
],
...
"parent": {
    "url": "Components/Displaying-Content",
    "title": "Displaying Content"
},
"children": [...],
"crumbs": [
    {
      "url": "Components",
      "title": "Components"
    },
    ...
],
"related": [...],
"prev": { "url" ... },
"next": {
    "url": "Components/Input",
    "title": "Input"
}
Custom Frontend

Easy to build your own:
— Language Agnostic Frontend
— Decoupled Architecture
— Versioned API
Prioritize

Ziel: innerhalb der Gruppe Konsens über gemeinsame Themen und Elemente für die Ziele des Designprojekts am wertvollsten sind.

Custom Frontend

From workbench to storefront
Future

— More Governance
— Classic Component Library
— Git Live Sync
— Native (Wrapper) App
— Fulltext search
Ziel: innerhalb der Gruppe Konsens über gemeinsame Themen und Elemente definieren, die für die Ziele des Designprojekts am wertvollsten sind.
Prioritize

Ziel: innerhalb der Gruppe
Konsens über gemeinsame Themen und Elemente de
finieren, die für die Ziele des Designprojekts am
wertvollsten sind.

Merci.

Project page:
github.com/atelierdisko/dsk

Download these slides:
share.atelierdisko.de/where-dt2018.pdf