

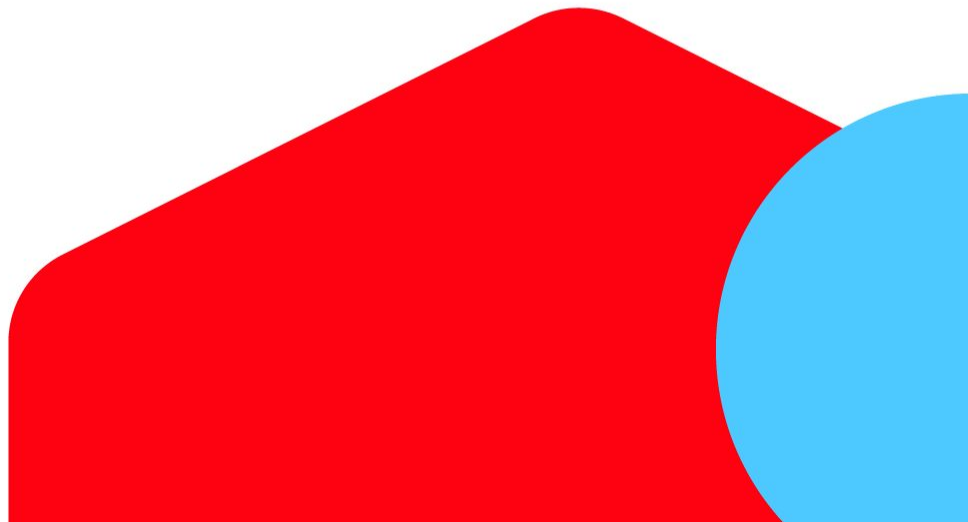
# メルカリ Design System Web の実装と責務

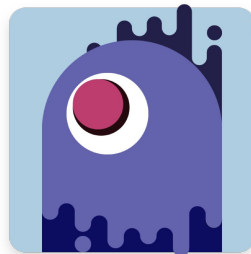
2022-07-20

**PWA Night vol.41**

@nekobato

mercari





HN

@nekobato

職業

Web Frontend Engineer

所属

株式会社メルカリ

# | Agenda

**01** メルカリ Design System Webの紹介

**02** Design System とは

**03** なぜ Design System なのか

**04** Design Systemの責務

**05** まとめ

# 今日の目標

---

Design Systemの実装紹介と運用の感想を一般化しつつお話する

# メルカリ Design System Web 実装の紹介

## Design Token

Color Palette  
Size, Spacing  
Shadow, Motion, etc...  
Alias token

## Styles

Reset CSS  
Theme CSS  
Util CSS

## Components

66 Components  
100+ Icons  
Common Resources

# Design Token

Color

Shadow

Typography

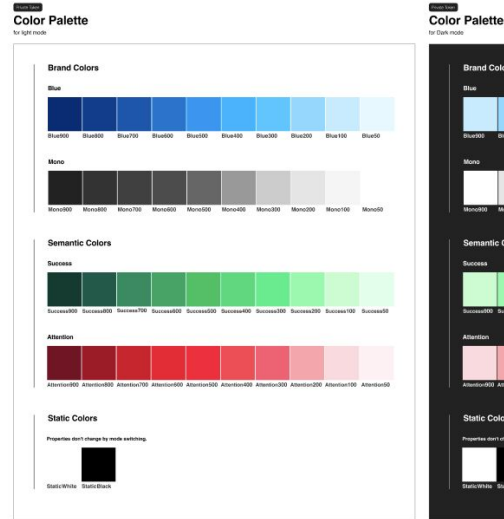
Sizing

Radius

Border

Opacity

Motion



JS Object

CSS postcss-simple-vars

# Design Token - Global TokenとAlias Token

## Global Token

Design System 内部用

変数名は基本的に数値

mono 100

fontSize 230

## Alias Token

外部から使用できる

変数名は意味で付ける

border primary highlight

typography x4large fontSize

# Design Token - 実際に運用してみた

- 変数に最初から意味を付けるのは良くない
  - Global Token → Alias Tokenのクッションを作ることで、変更を容易にできる
- 外部から使える変数にmediumとかlargeとか付けるのはやめたほうがいい
  - 絶対に後でmidLargeが追加される
  - medium, largeはまだAlias Tokenとしての抽象度が足りない



# Styles (CSS)

- Reset CSS
- Alias Tokenのオブジェクトを全てflattenして並べた変数
  - e.g. alias-color-light-text-secondary-default
- Utils
  - spacing, z-index
- その他Web Components のためのStyles

# Styles (CSS)

- Reset CSS は modern-css-reset をカスタムして利用
- typography, spacing, shadow 諸々 Design System で独自に定義している  
と tailwind などのツールは利用候補にならない
  - ここはリソースとの相談

# Components

The screenshot displays the Storybook interface for a component named 'ボタン' (Button). The interface is divided into three main sections: the left sidebar, the central Canvas, and the right-hand Controls panel.

**Left Sidebar:** A search bar labeled 'Find components' is at the top. Below it is a list of component categories, with 'mer-button' selected and highlighted in blue. Under 'mer-button', the 'Default' story is selected.

**Canvas:** A red button with the Japanese text 'ボタン' is displayed in the center.

**Right-hand Controls Panel:** This panel is titled 'Controls (6)' and contains a table of component properties and controls. The table has four columns: Name, Description, Default, and Control.

Name	Description	Default	Control
text		-	ボタン
<b>PROPERTIES</b>			
size	"small" "medium" "large"	"medium"	medium
intent	"primary" "attention" "secondary" "secondary-black" "flat"	"primary"	primary
disabled	boolean	false	false
fluid	boolean	false	false
loading	boolean	false	false
<b>SLOTS</b>			
*default*	put a or button here to receive style	-	-

# Components

Web Componentsで実装

テストの充実

- CI
  - インタラクション
  - a11y
  - 主要なプロダクトでの動作確認
- キーボード操作・Screen Readerによる実際の操作確認(マニュアル)
- canary releaseでcommit毎の確認を可能にする
- Visual regression testing by @storybook/addon-storyshots

## Web Components



- Browser native feature
- Reusability of Custom Elements
- Encapsulation with Shadow DOM
- Combination with various frameworks
  - Vue.js
  - Angular.js
  - React.js (with some issues)

**※ Web Componentsの細かい技術の話は今回しない**

# Use case

## Mercari Web

The screenshot shows a Mercari product listing for a decorative lantern. The lantern is white with horizontal stripes and purple floral patterns, hanging from a black frame with gold tassels. A large red banner with the word "SOLD" in white is overlaid on the top left of the product image. The listing details on the right include the seller name "test", size "XXS以下", price "¥300 送料込み", and a "取引画面を表示する" button. The product description and information sections are also visible.

mercari なにをお探しですか? お知らせ やることリスト マイページ 出品

<

**SOLD**

**test**  
XXS以下  
¥300 送料込み

いいね! コメント

取引画面を表示する

**商品の説明**  
test  
先月

**商品の情報**

カテゴリー レディース >  
トップス >  
Tシャツ/カットソー(半袖/袖なし)

商品のサイズ XXS以下

商品の状態 新品、未使用

フィードバック

# Design System とは

---

# どこから Design System なのか

定義	Documents
デザイン	Figmaでのデザイン設計
見た目の実装	CSS または コンポーネント実装
振る舞いの実装	コンポーネント実装

どこまで提供するかは組織とチームによって最適な選択がある



# Why does it matter for Mercari?

---

なぜDesign Systemなのか

# なぜ Design System なのか

- **一貫性**
  - 一元化されたデザインリソースによってプロダクトの品質を保つ
  - 同じブランド内にあるプロダクト群でのデザインを統一する
- **効率性**
  - モダンなユーザー体験
  - スケーラブルな実装

## 一貫性:実際に運用してみて

独自のコンポーネント・独自のルールがもたらす開発体験は当然デザイナー  
・エンジニアにとって習熟に時間がかかる

難解なコンポーネントを作ればそれだけ意図しない利用が増える  
誤解のない定義と実装には検討の時間が非常に重要

# 効率性:実際に運用してみて

本当にDesign System実装は開発効率を上げているか

テストは開発効率を上げるか?という問いに似ている

Design Systemはプロジェクトのデザイン品質を保証する

サービスが複雑化するに従って品質が劣化するのを防ぐ

とはいえデザインの問題はバグとして明確に発見できるものではない

# Design System の責務

---

# ライブラリとしての責務

## Documentation

型によるパラメータの案内

コードドキュメントによるアンチパターンの防止

## Accessibility

- Color contrast
- Keyboard operation
- Text-to-Speech operation
- ...and other disabilities

## Mode & Theming

- Light Mode
- Dark Mode
- Custom Themes

# ライブラリとしての責務

## Devices

レスポンスな動作と境界数値の提供

- Desktop
- Tablet
- Smartphone

## Internationalization

全ての文字列においてのカスタム機能の提供

# Documentation

## Documentation for the Web Developers

- Storybook
- Code documents
- Type definition
- README
  - Code documentsとType definitionのまとめ
- Figma
  - デザインの確認

説明に不備があれば質問コーナーで一日の半分が終わることも



# | Accessibility

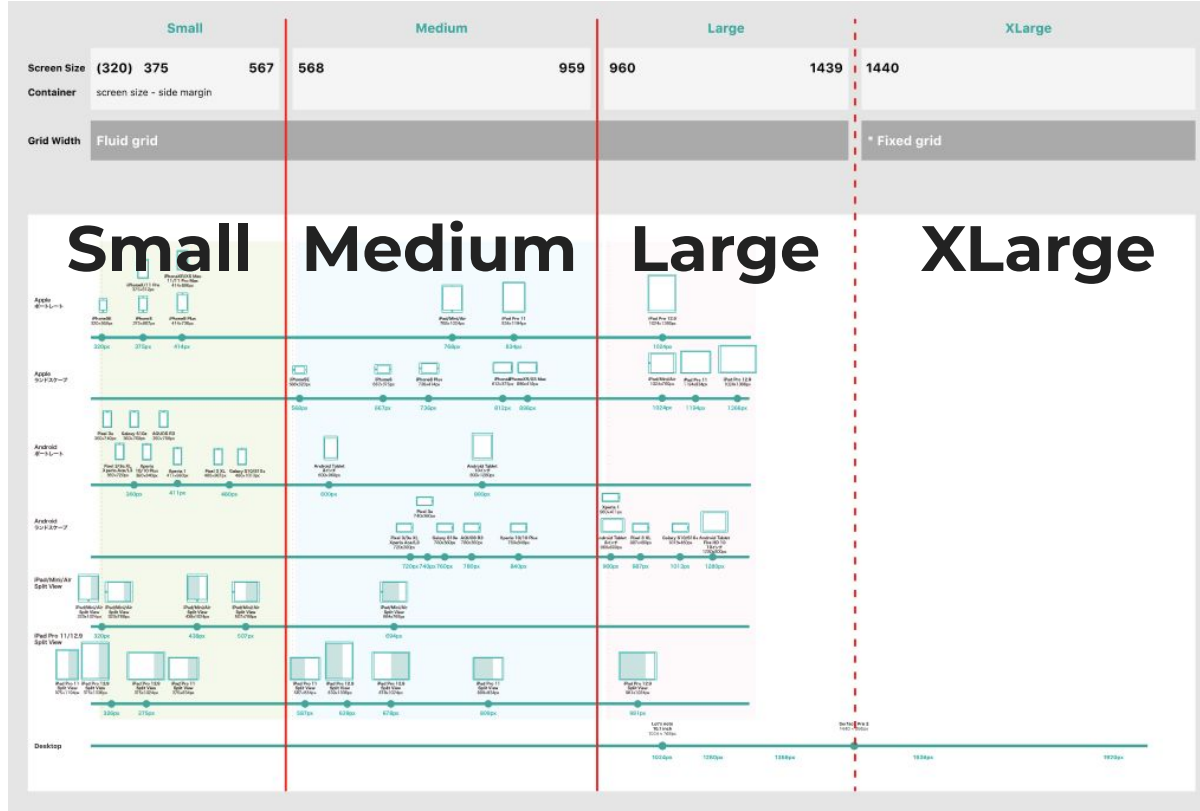
- [WCAG 2.1 AA](#) の達成
  - 基本的なDOM構造
    - axe-coreによるCI testing
  - 色やインタラクション時のDOM構造
    - @storybook/addon-ally
- キーボード操作とスクリーンリーダーのマニュアルテスト

# Mode and Theming

```
ⓧ ⓧ ⓧ  
  
@media (prefers-color-scheme: light) {  
  :root {  
    --alias-color-text-primary-default: $alias-color-dark-text-primary-default;  
  }  
}
```

```
ⓧ ⓧ ⓧ  
  
.dark-theme {  
  --alias-color-text-primary-default: $alias-color-dark-text-primary-default;  
}
```

# Device



# Internationalization (i18n)

- Design System 側では全てのテキスト情報を持たない
  - 全てをプロダクト側で指定するパラメータ化
    - + 型による実装の強制
  - 言語によって順番が違うテキスト個別で対応
    - 値段・日付など

# プロジェクトとしての責務

提案と仕様策定:新しいコンポーネントはプロダクト側から? DS側から?

デザインレビュー:全てのプロダクトのデザインはDSがチェックするか否か

実装:必要なコンポーネントはDSが実装するのか、必要なプロダクト

導入:プロダクトへの導入

バグ修正:コンポーネント周辺でバグがあった場合の調査、修正

バージョン更新:更新が発生した時、マイグレーションを誰が行うのか

# プロジェクトとしての責務

責務を握らないと Design System のプロジェクトは機能不全に

- 必要のないコンポーネント
- 守られないルール
- 進まない実装
- 更新されないライブラリ

リソースと組織の状況によって最適な選択があるはず

# まとめ

- メルカリ Design Systemの全体構造とその責務
- どこまで提供するのが最適かは状況次第
- どこまでサポートするかも状況次第
- 使ってもらうためにはドキュメントが大事