

ButtonとLink、どう実装する？

2023/08/30 Encraft #6

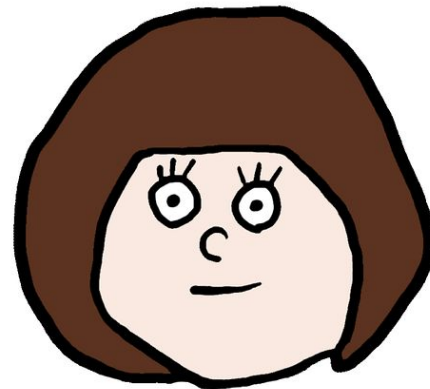
@yoshiko

よしこ [@yoshiko_pg](#)

株式会社ナレッジワークのフロントエンドエンジニア
GUIをSPAとして作るのが好き。

自社での技術スタックや設計を Zenn で公開しています。

<https://zenn.dev/yoshiko>



TECH

2020年に立ち上げたWebフロントエンド構成の振り返り

2021/09/27 ♡ 653

TECH

React+TSプロジェクトで便利だったLint/Format設定紹介

2021/09/30 ♡ 396

TECH

SPA Componentの押しディレクトリ構成について語る

2021/10/29 ♡ 821

TECH

「3種類」で管理するReactのState戦略

2021/12/31 ♡ 741



TECH

フロントエンドアーキテクチャの話: Resource Setの紹介

16日前 ♡ 320

要件

デザインデータから導き出された必要そうな要件はこんな感じ

-  onClick によるアクション
-  href によるページ遷移
- Link Text[†] onClick によるアクション
- Link Text[†] href によるページ遷移

※ 便宜上クリック/onClickと記しますがキーボードその他の操作も対象としています

見た目 \ 挙動	onClick	href
ボタン		
リンク	<u>リンクみたいなボタン</u>	<u>リンク</u>

見た目 \ 挙動	onClick	href
ボタン		
リンク		

ちょっとまって

見た目 \ 挙動	onClick	href
ボタン		
リンク		

これ本当に要るの？

見た目 \ 挙動	onClick	href
ボタン		
リンク	<u>リンクみたいなボタン</u>	<u>リンク</u>

これ本当に要るの？
(って思った人手あげて～)

当初は存在したが、途中から
アクションにリンクの見た目を
使うのをやめる整理をした。

今は全て Transparent Button に
置き換わっている。

→「リンクみたいなボタン」は不要

ナレッジワーク

プロダクトガイドラインより抜粋 ▶

ボタン

ユーザーが行うアクションには、3種類のボタンを使い分ける。

Components - Button

Primary



- 主要なアクション
- 原則、1画面に1回のみ

Secondary



- 汎用的なアクション
- Primaryに対する副次的なボタン
- 複数のアクションが並ぶようなレイアウトに最適

Transparent



- 優先度が低いアクション
- 複数のアクションが並ぶようなレイアウトに最適

ページ遷移がメインアクションになる文脈が存在する。
たとえば「ユーザー追加」でユーザー追加ページに
遷移する要件の場合、必要な振る舞いはリンクである。

だが「ユーザー追加」の見た目がリンクだと、
それが現在期待するメインのアクションであっても、
そのことが直感的にユーザーに伝わらなくなる。

→「ボタンみたいなリンク」は必要な場面がある



見た目 \ 挙動	onClick	href
ボタン		
リンク		<u>リンク</u>

見た目 \ 挙動	onClick	href
ボタン		
リンク	まだある	<u>リンク</u>

見た目 \ 挙動	onClick	href
ボタン		
リンク		
任意の内容	<p>スペースで絞り込む</p> <p>営業資料</p> <div> <div>  会社紹介 3 </div> <div>  詳細別掲載 4 </div> <div>  ヒアリング 2 </div> </div> <p>リソースに応じたカスタム選択UI</p>	 <p>リソースに応じた詳細への遷移UI</p>

見た目 \ 挙動	onClick	href
ボタン		
リンク		
任意の内容	<p>スペースで絞り込む</p> <p>営業資料</p> <div>  会社紹介 3 </div> <div>  課題別提案 4 </div> <div>  ヒアリング 2 </div> <p>リソースに応じたカスタム選択UI</p>	 <p>リソースに応じた詳細への遷移UI</p>

これらをうまく、いい感じに実装したい

実装

まずは見た目と振る舞いを分離して考える

振る舞い

クリック
アクション

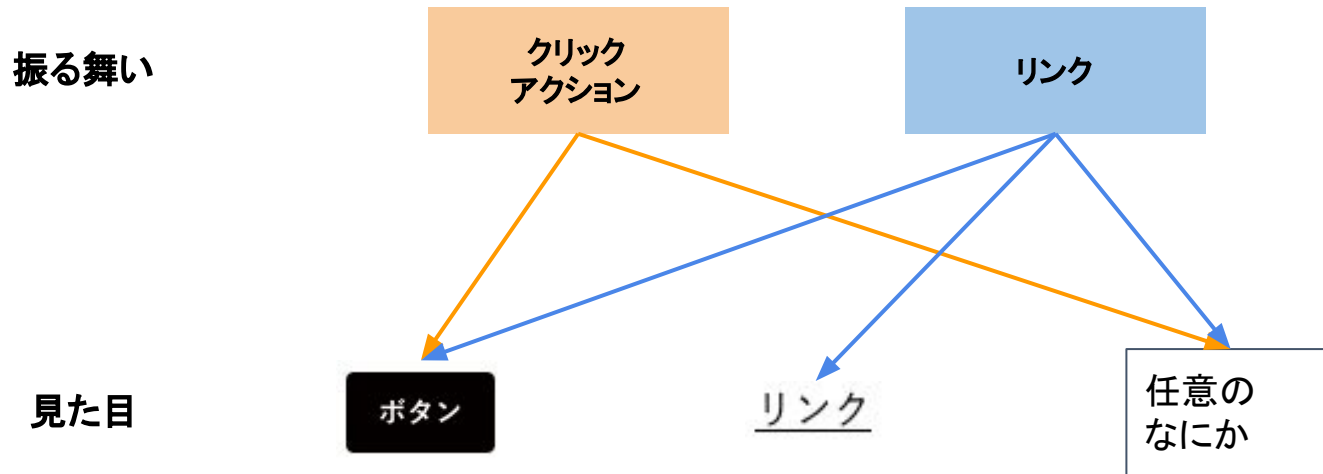
リンク

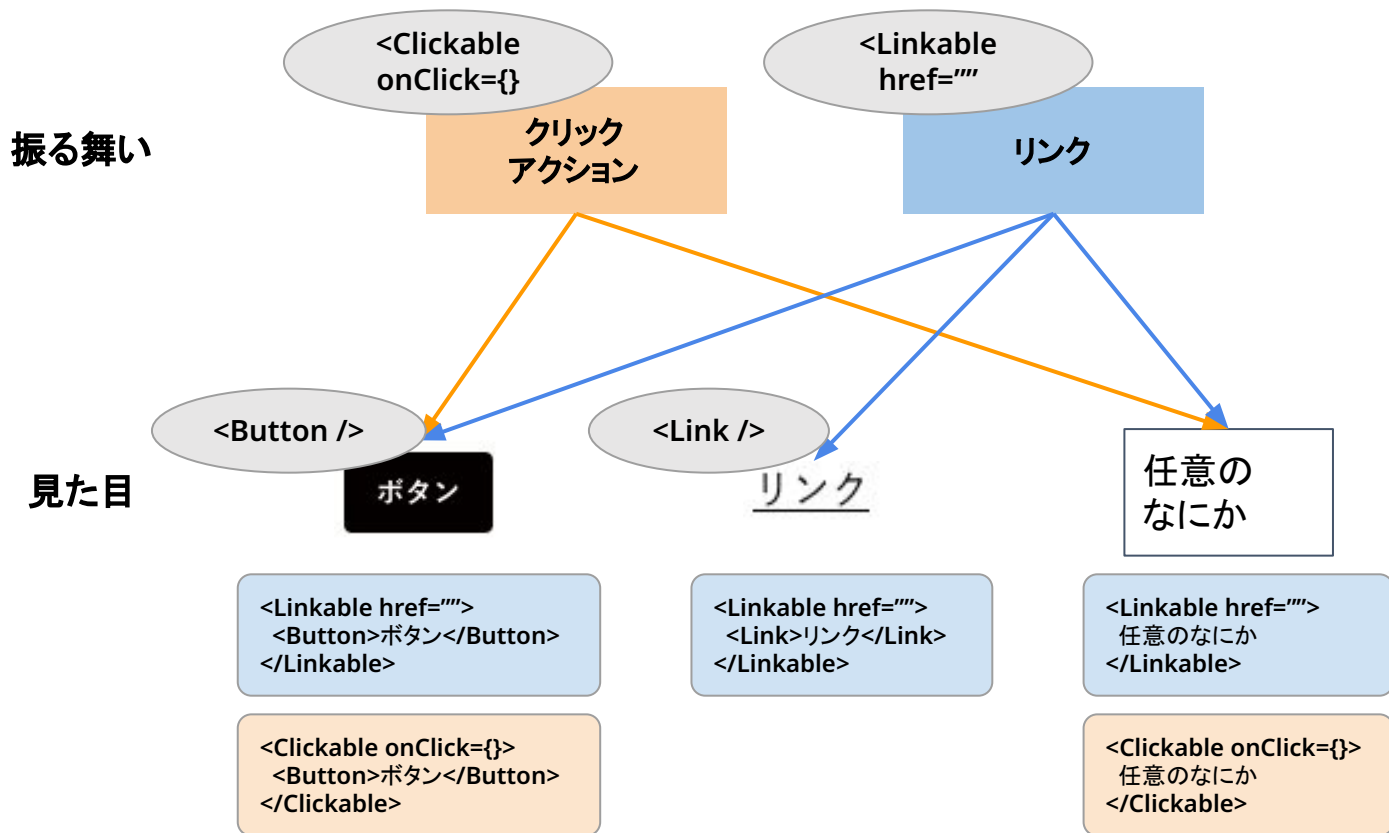
見た目

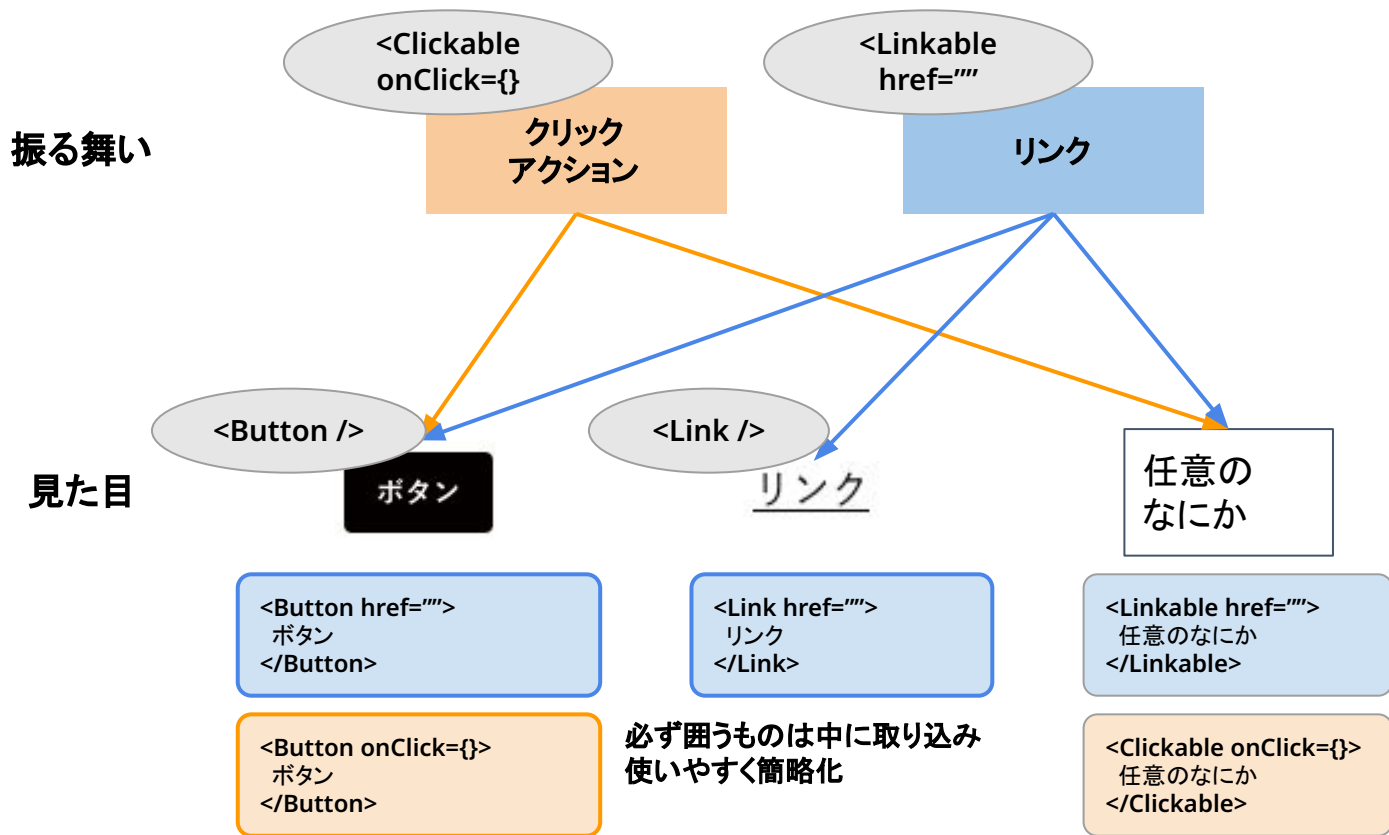
ボタン

リンク

任意の
なにか







見た目を持たず、ボタンの振る舞いだけを持つComponent

- 必須Props: `onClick` , `children`
- 任意Props: button要素の持つProps + div要素の持つProps
- 実体: `div`
 - HTMLのInteractive Content (button) を使いたいが、buttonの下に許可されているのはPhrasing Contentのみなのでdivなどを入れられない。Componentとして任意のchildrenを持たせる際にその制約は大変なので、要素はdivにしてボタンの振る舞いは自前で実装している
 - roleをつける、tabIndexをつける、keyboardによる操作の対応、など

ほんとにbutton使えたら一番楽なんですけどね。

中身のマークアップが複雑かつクリッカブルな要素、みんなどうしてますか？

見た目を持たず、リンクの振る舞いだけを持つComponent

- 必須Props: `href` , `children`
- 任意Props: a要素の持つProps + `disabled` , `replace`
- 実体: `a`
 - Propsに応じてnext/linkのLinkとHTMLのa要素を使い分けている
 - リンクがここに集約されるので、next/linkのインターフェイスが変わったときの対応も楽だった
 - `referrerPolicy` 属性や `rel` 属性などの漏れがちな指定も集約できる
 - Clickableのようにdisabled属性が欲しいことがたびたびあったので独自に追加

ボタンの見た目と、ボタンもしくはリンクの振る舞いを持つComponent

- 必ずインタラクティブになるので、ClickableとLinkableの振る舞いは内部に取り込んだ
 - `href` が渡されたら内部的にLinkableを使い、そうでなければHTMLのbutton要素を使う
(ButtonではHTMLのbutton要素を使うのが適切と思ったのでClickableは使わないことにした)
 - button要素の場合、typeのデフォルトはsubmitではなくbutton
- 見た目のバリエーション
 - カラーバリエーション、サイズバリエーションを持つ
 - 任意のアイコンを入れられる
 - loading属性でloadingを表示できる

リンクの見た目と振る舞いを持つComponent

- 必ずリンクになるので、Linkableの振る舞いは内部に取り込んだ
 - 内部的には直接 Linkable を返している
 - このComponentがやっていることはリンクの見た目をCSSで実装し、LinkableのclassNameに渡すこと

振る舞いと見た目を分離して実装し、振る舞いだけでも利用できるようにすることで、
複雑なインタラクションの実装を一箇所に集約しています。

カスタムコンポーネントを実装中にクリックアクション / リンクの振る舞いが必要になったときにも、
<Clickable> <Linkable> で囲うだけで配慮込みの振る舞いを実現できるので、効率的に機能実装ができます。

うち是这样しているよという紹介なので、うちは他のやりかたでうまくやれているよという話があれば知りたいです！

できる喜びが巡る日々を届ける

Deliver the joy of enablement