

# Coder's High 2017

2017.11.4

# 多様なユーザーニーズに応える フロントエンドデザインパターン ～ベーシック編～

書籍「インクルーシブHTML+CSS & JavaScript」より

伊原 力也 / 太田 良典

2017.11.04

CSS Nite LP54「Coder's High 2017」

# 自己紹介



## 伊原 力也

free株式会社 IA/UX。アクセシブルなインタラクションデザインの実践を標榜し、Webサービスやスマートフォンアプリの設計業務に従事。ウェブアクセシビリティ基盤委員会(WAIC)理解と普及作業部会委員としても活動。HCD-Net認定 人間中心設計専門家および評議委員。クリエイティブユニットmokuva所属。



## 太田 良典

HTML4.01のW3C仕様書を翻訳した「HTML4仕様書邦訳計画補完委員会」の委員を務めた後、2001年にBAに参加。Web技術の分野で幅広い専門性を持ち、セキュリティ分野においては「第二回IPA賞(情報セキュリティ部門)」を受賞。アクセシビリティ分野では、ウェブアクセシビリティ基盤委員会(WAIC)の委員として活動している。



## ウェブアクセシビリティ 基盤委員会(WAIC)

Web  
Accessibility  
Infrastructure  
Committee

ウェブアクセシビリティ基盤委員会(WAIC: Web Accessibility Infrastructure Committee)は、日本におけるウェブアクセシビリティの公的規格であるJIS X 8341-3の理解と普及を促進するとともに、JIS X 8341-3を利用してウェブアクセシビリティを高めていくために必要な基盤を構築するために、さまざまな活動を行っています。

JIS X 8341-3の規格票は、以下のいずれかの方法でご覧いただくことができます。どちらも規格番号「X8341-3」で検索してください。

- [閲覧：「日本工業標準調査会 \(JISC\)」Webサイトの「JIS検索」](#)
- [購入：「JSA Web Store \(日本規格協会\)」Webサイトの「JIS検索」](#)

### ▶ [組織について](#)

ウェブアクセシビリティ基盤委員会についてのご紹介です。

### ▶ [Q&A](#)

JIS X 8341-3について寄せられた質問への回答を掲載しています。

### ▶ [新着情報](#)

2017年9月15日

10月5日開催の[アクセシビリティセミナー2017](#)で予定されているセッション「わたしたち、こうしてつくっています～アクセシブルなサービス提供に向けた取り組み～」に登壇者が2名、新たに加わりました。

2017年9月1日

2017年10月3日から6日まで、幕張メッセで開催されるCEATEC JAPAN 2017の期間中、10月5日(木)に[アクセシビリティセミナー2017](#)を開催します。

2017年8月10日

平成29年7月24日付で施行した[規程](#)(旧「会則」)を掲載しました。

2017年7月17日

[お問い合わせのページ](#)に、当サイトで公開しているWCAG 2.0関連翻訳文書に関するお問い合わせ先を掲載しました。

# Amazonで「アクセシビリティ」と検索！



問題は設計にあり!?  
ガイドライン対策ではない、  
UX視点のデザインプロセス

戦略策定から要件定義・設計・ビジュアルデザイン・実装まで。  
ユーザーが陥りがちな問題を各フェーズで解決するノウハウ集

マルチデバイス時代の  
“使える”  
Webサイトとは!



HTML5+JavaScriptの  
モダンWebアプリを  
アクセシブルに!

日本初!  
WAI-ARIAの  
本格解説書

ドイツのWeb開発&デザインメディア SMASHING MAGAZINE刊  
「Apps For All - Coding Accessible Web Applications」日本語版



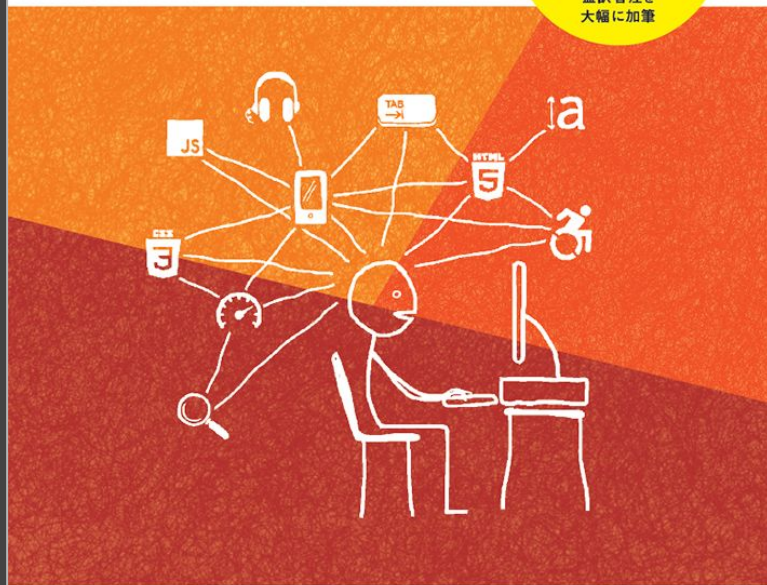
# インクルーシブHTML +CSS&JavaScript

多様なユーザーニーズに応える  
フロントエンドデザインパターン

著者 | ヘイドン・ピカリング  
監訳 | 太田典典、伊原力也

「ダメ」ではなく  
「こうしよう!」  
を学ぶ本

監訳者注を  
大幅に加筆



マークアップの最適解に近づくための、ロバストで理にかなった考え方



# このセッションの流れ

- インクルーシブ・デザインパターン
- ボタン vs ボタン
- ナビゲーション
- ブログ記事
- まとめ

# インクルーシブ・デザインパターン



原著名: インクルーシブ・デザインパターン

# **Inclusive Design Patterns**

## **Coding Accessibility Into Web Design**

# インクルーシブとは？

## Inclusive

- 「包括的」「全てを含む」など
- exclusive (排他的) の反対語

ここでは、「誰でも受け入れる」  
「特定の人を除外しない」のような意味

= どんな人、どんな環境でもアクセスできる  
「アクセシビリティ」とほぼ同義



# アクセシビリティ系の記事のよくあるパターン

- いろいろダメと言われる
- 特にJS系の処理がダメと言われる
- 結論が「やるな」「避ける」になってしまうことが多い
  - 一般に普及しているUIを、ただ「ダメ」と言われても困る
  - 結局、指摘を無視してやってしまうことになることが多い

# この本は？

普及しているUIについて、アクセシブルなデザインパターンを提供

- 「やるな」ではなく、「**こう実装するとアクセシブル**」という話  
(とはいえ、これはやってはダメ、という話もちろんある)
- ブログ記事、ナビゲーション、メニューボタン、  
商品リスト、フィルターウィジェット、登録フォームなど
- 今日は「ボタン」「ナビゲーション」「ブログ記事」の一部を紹介

# ボタン vs ボタン

スタート



賛成!



メニュー

# インクルーシブなボタン

インタラクティブ要素である「スタート」と書かれたボタンを、  
**3種類のデザイナーの視点**で見えます。

Webというメディアに対する少しの知識が、よりシンプルかつ  
インクルーシブな解決策を導き出すことに注目してください。



スタート

# 3種類のデザイナー

- グラフィックデザイナー
  - ビジュアルデザインの専門家、実装はできない
- コードを書くデザイナー
  - ビジュアルデザインがメインだが、ある程度実装もできる
- インクルーシブデザイナー

みなさんなら、どのように実装しますか？

**スタート**



# グラフィックデザイナー

彼らにとってボタンは視覚的な成果物であり、

Adobe IllustratorやSketchで作るものです。

本物のボタンのように見えるか、

全体のブランディングにフィットしているかを常に考えています。

そのボタンをどうやってWebページに実装するか、

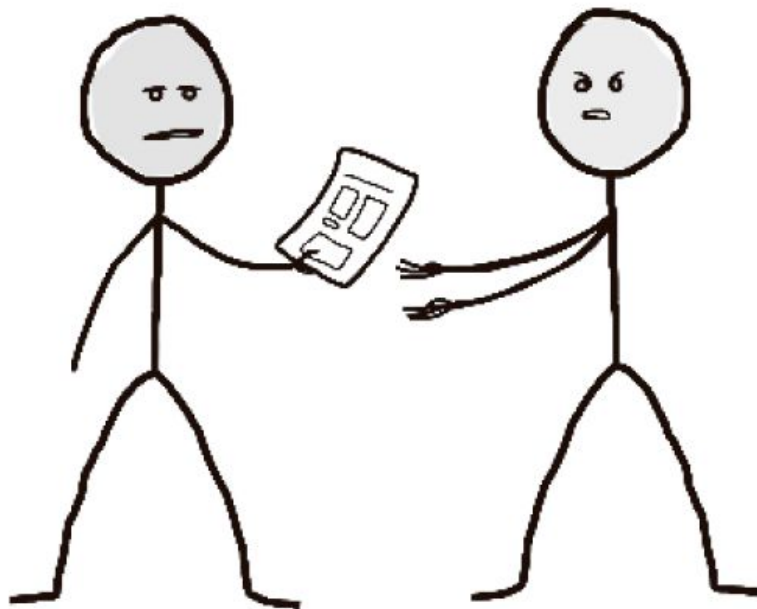
どうやって動作させるかは**まったく考えていません**。

# グラフィックデザイナーの「スタート」ボタン

はい、頼まれていた  
君のWebサイトだよ

ただの紙切れじゃないか

作ってくれじゃなくて、  
デザインしてくれ  
って言ったよね



# コードを書くデザイナー

2番めにとりあげるデザイナーは、最初のデザイナーとほとんど同じスキルをもっていますが、ひとつ大きな違いがあります。

彼らは、Webページ内にボタンを表示し、JavaScriptのイベントリスナーを指定できるだけの**HTML、CSS、JavaScriptの知識**を持っています。

# コードを書くデザイナーの「スタート」ボタン

```
<div class="button"></div>
```

```
.button {  
  width: 200px;  
  height: 70px;  
  background: url('../images/button.png');  
}
```

```
button.addEventListener('click', function() {  
  // クリックでイベント発火  
});
```

# インクルーシブデザイナー

インクルーシブデザイナーは、

**コードを書くデザイナーのボタンを、**

潜在ユーザーを想定した**さまざまな観点から検討**します。

そうすると、**この実装の問題点**が次々と見えてきます。

# コードを書くデザイナーのボタンにはこんな問題が

- ズームすると画像がボヤける
- ブラウザの設定で文字サイズを大きくしても文字が拡大しない
- 背景画像が表示されないことがある
- キーボードで操作できない
- スクリーンリーダーにボタンであると伝わらない
- ラベルが画像なので機械翻訳できない

<div class="button"

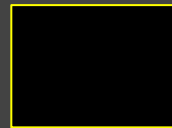
↑

id="123"



# インクルーシブデザイナーの「スタート」ボタン

```
<button>スタート</button>
```







```
<button type="button">  
  Use button,  
  not div  
</button>
```

<https://suzuri.jp/masuP9/815489/t-shirt/s/sumi>

「賛成」ボタンをどう修正する？



**賛成！**

# 「コードを書くデザイナー」によるマークアップ

```
<div class="upvote" data-action="upvote"></div>
```

## アクセシビリティチェックで登録されたチケット

- ▶ 問題のある代替テキスト、102ページ **バグ**
- ▶ 代替テキストの不足、66ページ **バグ**
- ▶ 不適切な代替テキスト、94ページ **バグ**
- ▶ 代替テキストの欠如、29ページ **バグ**
- ▶ 代替テキストの不具合、84ページ **バグ**

## 1.1.1「非テキストコンテンツ」に対応

```
<div class="upvote" data-action="upvote" aria-label="賛成！"></div>
```

## 2.1.1「キーボード」に対応

```
<div class="upvote" data-action="upvote" aria-label="賛成！"  
tabindex="0"></div>
```

## 4.1.2「名前(name)・役割(role)及び値(value)」に対応

```
<div class="upvote" data-action="upvote" aria-label="賛成！"  
tabindex="0" role="button"></div>
```



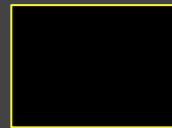
```
<button type="button">  
  Use button,  
  not div  
</button>
```

<https://suzuri.jp/masuP9/815489/t-shirt/s/sumi>



# 賛成ボタンもbutton要素で

```
<button data-action="upvote" aria-label="賛成！"></button>
```



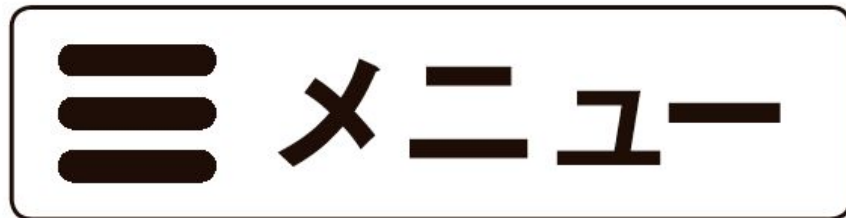
# 画像を背景から前景に変更

```
<button data-action="upvote">  
  <svg aria-label="賛成！">  
    <use xlink:href="#upvote"></use>  
  </svg>  
</button>
```

# よくあるメニューボタンの実装



ボタンらしい見た目 + ラベルがおすすぬ



# アイコンの実装方法あれこれ

- 背景画像
- 前景画像
- アイコンフォントのグリフ
- Unicode
- SVGスプライト

# 背景画像で実装した場合の問題点

Windowsでハイコントラストモードに切り替えると  
背景画像は表示されなくなります。

「メニュー」というテキストも提供していれば、

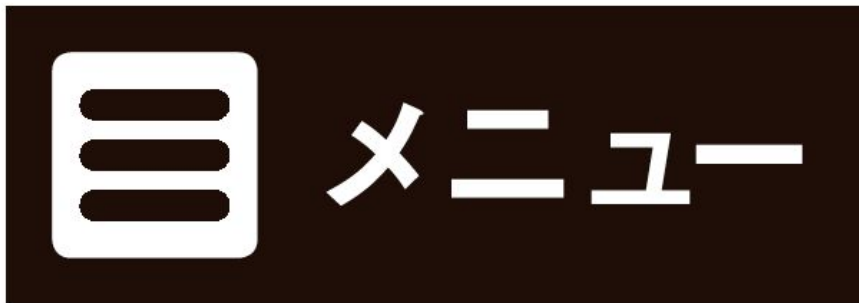
ぶち壊しにはなりません

(ハイコントラストモードにしたとき、背景色とともに

テキストの色も反転するため、引き続き読めるのです)。

# 前景画像にするとどうか？

☰ メニュー



ハイコントラストモードの黒背景では、  
アイコンは白枠で囲まれて異なる見え方になるうえに、  
「メニュー」というテキストとも分離して見えます。

# アイコンフォントによる実装

最近よく見られるが、いくつか問題がある

- Webページのフォントをユーザーが独自に変更している場合
- アイコンフォントの読み込みに失敗した場合



豆腐

メニュー



# Unicode標準の記号を使う



U+2630 'TRIGRAM FOR HEAVEN'

(「八卦」の「天」)

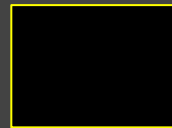
## U+2630(☹)を使った実装の問題

- すべてのデバイスがサポートしているわけではなく、  
表示できない可能性がある
- スクリーンリーダーがこの記号を  
「トリグラムフォーヘブン」と読み上げることがある



# 読み上げられないようにする

```
<button>  
  <span aria-hidden="true">☰</span>  
  メニュー  
</button>
```



# SVGスプライト

**SVGスプライト**は、急速にアイコン表示の事実上標準の解決策となりつつあります——それには**正当な理由**があります。

Googleによる305バイトのロゴ実装が証明したように、アセットを非常に**小さく**できます。劣化せずに**拡大縮小**でき、フォントカラーの変更に合わせて**色を変える**こともできます。

# SVGスプライトの準備

```
<svg style="display: none;">  
  <symbol id="navicon" viewBox="0 0 20 20">  
    <path  
      d="m0-0v4h20v-4h-20zm0 8v4h20v-4h-20zm0 8v4h20v-4h-20z"  
      fill="currentColor" />  
  </symbol>  
  <!-- 他にも<symbol>要素がたくさん -->  
</svg>
```

# SVGスプライトの使用

```
<button>  
  <svg><use xlink:href="#navicon"></use></svg>  
  メニュー  
</button>
```

```
button svg {  
  width: 1em;  
  height: 1em;  
}
```

# アイコン単体で使いたいときは？

前述したとおり、ボタンに「メニュー」というテキストを含めることには認知面のメリットがあります。

また、何らかの理由でアイコンが表示されなかった場合でも、`<button>`のわかりやすさが保たれます。

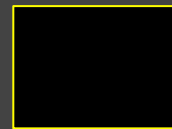
それでもやはり、**アイコン単体**で使いたい場合もあるでしょう。

その場合は、スクリーンリーダーユーザーにも

確実に「**メニューボタン**」と伝わるようにすることが重要です。

# SVGに加えて非表示の<span>でラベルを追加

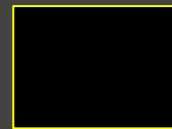
```
<button>  
  <svg><use xlink:href="#navicon"></use></svg>  
  <span class="visually-hidden">メニュー</span>  
</button>
```





# ボタン全体にaria-label属性でラベルを追加

```
<button aria-label="メニュー">  
  <svg><use xlink:href="#navicon"></use></svg>  
</button>
```

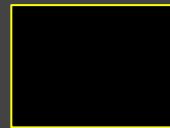


# 記号文字のボタンにaria-labelでラベルを追加

```
<button aria-label="メニュー">
```

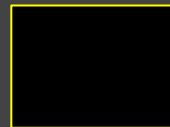


```
</button>
```



# 開閉するメニューを実装して完成

```
<nav aria-label="サイト">  
  <button aria-expanded="false">  
    <svg><use xlink:href="#navicon"></use></svg>  
    メニュー  
  </button>  
  <ul hidden>  
    <li><a href="#main">ホーム</a></li>  
    <li><a href="/about">企業情報</a></li>  
    <li><a href="/products">製品情報</a></li>  
    <li><a href="/contact">お問い合わせ</a></li>  
    <li><a href="/login">ログイン</a></li>  
  </ul>  
</nav>
```



# 開閉メニュー実装時のポイント

- JavaScriptで<ul>にhidden属性を追加して非表示にする  
→hiddenで隠せば、閉じているとき中身にフォーカスが当たらなくなる
- <button>は<nav>要素の中に配置する  
→外に置くと、ランドマークに飛んだとき中身が空になってしまう
- <button>の**すぐ後**にメニューを配置する  
→ボタンの次にメニューの最初の項目にフォーカスが当たる
  - すぐ後にメニューを置けないことも……?

# ボタンのまとめ

- 原則として、ボタンは<button>で実装する
  - <button>警察 (Tシャツの人)を思い出しましょう
- メニューボタンにはラベルを付けるのがおすすめ
  - アイコンのみの実装もアリだが、スクリーンリーダーにも伝わるように
- アイコンの実装方法はいろいろある
  - 記号を使ったりSVGを使ったり
- 開閉メニューを実装するときもいろいろ注意
  - キーボード操作時も考慮

# ナビゲーション領域



ホーム 企業情報 製品情報 お問い合わせ ログイン



# サイト内共通のナビゲーション

```
<nav>
```

```
<ul>
```

```
<li><a href="/">ホーム</a></li>
```

```
<li><a href="/about">企業情報</a></li>
```

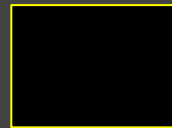
```
<li><a href="/products">製品情報</a></li>
```

```
<li><a href="/contact">お問い合わせ</a></li>
```

```
<li><a href="/login">ログイン</a></li>
```

```
</ul>
```

```
</nav>
```



## スクリーンリーダーでは？

最初に「ナビゲーション ランドマーク」と読み上げられた後、  
「リスト 5項目」と続き、  
最後に「home リンク」と読み上げられたとします。

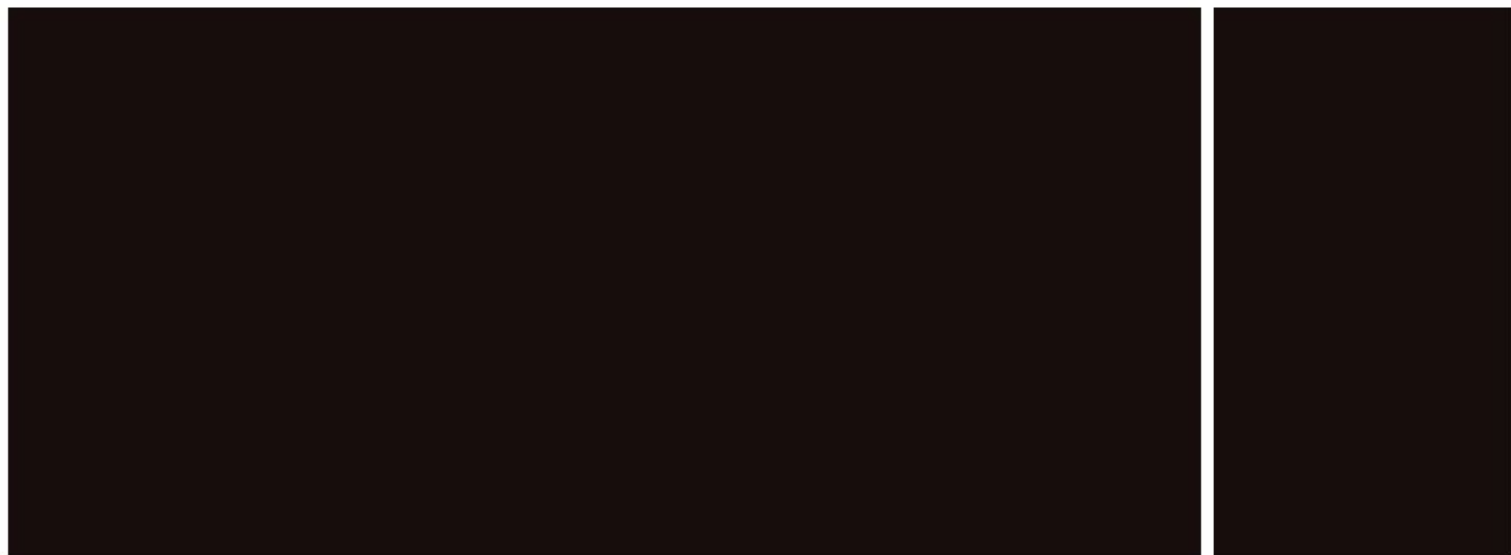
これにより、自分がいまnavigationランドマークにいること、  
そしてそこには全部で5つのリンクがあることがわかります。  
望むなら、すぐに最初のリンク先をたどることもできます。



# 外観と配置



ホーム 企業情報 製品情報 お問い合わせ ログイン



# CSSが読み込めなかったときも何なのかわかる

- ホーム
- 企業情報
- 製品情報
- お問い合わせ
- ログイン

現在位置は色以外でも区別する

ホーム

企業情報

製品情報

お問い合わせ

ログイン

さらに大胆にしても良い

ホーム

企業情報

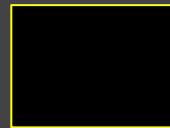
製品情報

お問い合わせ

ログイン

# スクリーンリーダーのサポート: 方法1

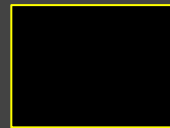
```
<nav>
  <ul>
    <li><a href="/">ホーム</a></li>
    <li><a href="/about"><span
class="visually-hidden">現在のページ</span> 企業情報</a></li>
    <li><a href="/products">製品情報</a></li>
    <li><a href="/contact">お問い合わせ</a></li>
    <li><a href="/login">ログイン</a></li>
  </ul>
</nav>
```



## スクリーンリーダーのサポート: 方法2

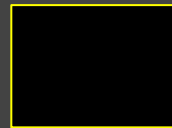
```
<nav>
  <ul>
    <li><a href="/">ホーム</a></li>
    <li><a href="/about"
      aria-describedby="current">企業情報</a></li>
    <li><a href="/products">製品情報</a></li>
    <li><a href="/contact">お問い合わせ</a></li>
    <li><a href="/login">ログイン</a></li>
  </ul>
  <div hidden id="current">現在のページ</div>
</nav>
```

※結果: iOS VoiceOverでは効果なし:



# スクリーンリーダーのサポート: 方法3

```
<nav>
  <ul>
    <li><a href="/">ホーム</a></li>
    <li><a href="/about" aria-current="page">企業情報</a></li>
    <li><a href="/products">製品情報</a></li>
    <li><a href="/contact">お問い合わせ</a></li>
    <li><a href="/login">ログイン</a></li>
  </ul>
</nav>
```



# 冗長なリンクは取り除くべき？

```
<nav>
  <ul>
    <li><a href="/">ホーム</a></li>

    <li><a href="/products">製品情報</a></li>
    <li><a href="/contact">お問い合わせ</a></li>
    <li><a href="/login">ログイン</a></li>
  </ul>
</nav>
```



# 冗長なリンクはスキップリンクに？

```
<nav>
  <ul>
    <li><a href="/">ホーム</a></li>
    <li><a href="#main">企業情報</a></li>
    <li><a href="/products">製品情報</a></li>
    <li><a href="/contact">お問い合わせ</a></li>
    <li><a href="/login">ログイン</a></li>
  </ul>
</nav>
```

# サイトロゴと「ホーム」のリンクかぶり問題

```
<header role="banner">  
  <a href="/home"></a>  
  <nav>  
    <ul>  
      <li><a href="#main">ホーム</a></li>  
      <li><a href="/about">企業情報</a></li>  
      <li><a href="/products">製品情報</a></li>  
      <li><a href="/contact">お問い合わせ</a></li>  
      <li><a href="/login">ログイン</a></li>  
    </ul>  
  </nav>  
</header>
```

# 重複をなくす？

```
<nav>
  <ul>
    <li><a href="#main"></a></li>
    <li><a href="/about">企業情報</a></li>
    <li><a href="/products">製品情報</a></li>
    <li><a href="/contact">お問い合わせ</a></li>
    <li><a href="/login">ログイン</a></li>
  </ul>
</nav>
```

# ナビゲーションのまとめ

- 現在位置の表示を考えることは重要
  - 色に依存しない、スクリーンリーダーでも伝わるようにする
- 現在位置の実装方法はいろいろある
  - 非表示のテキストを入れる、ページ内リンクにする、aria-currentなど
- 冗長なリンクをどうするか考える
  - さまざまな考え方があり、これが正解、というものはない

# ブログ記事

—— 公開日 2017/12/12 ——

## インクルーシブなブログ記事のマークアップ方法

私のすばらしいアイデアには、これを支える多くの裏づけがある。たとえば、John Thoughtleader氏の「ヘイドンが正しい理由」、Jane Unicornfield氏の「ヘイドンを支持する」、そしてHarry Surname氏の「ヘイドンが語る時、それはかぐわしい虹を吐くかのようだ」など

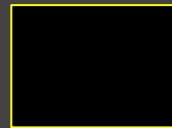
# 見出しレベルに注意

```
<h1>ブログ記事のマークアップ方法</h1>
```

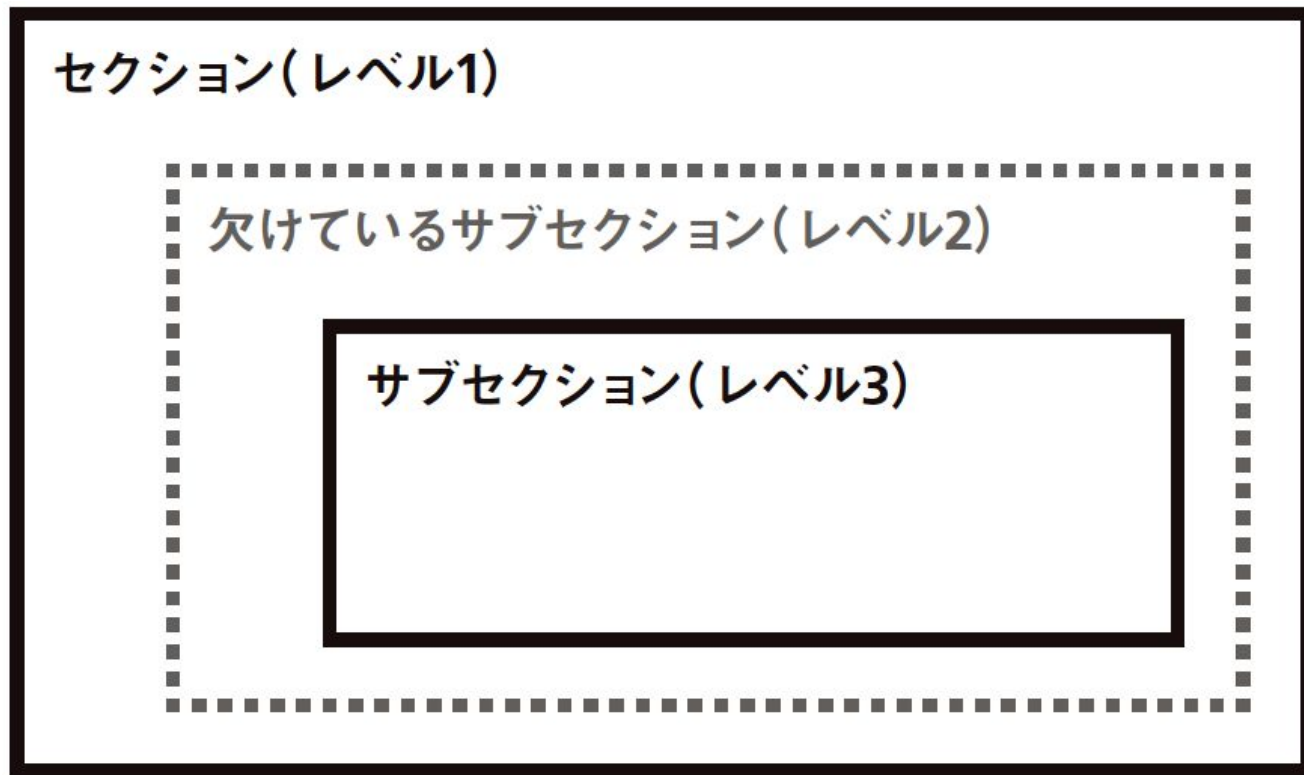
```
<!-- 序文コンテンツ-->
```

```
<h3>「セマンティック」という単語について</h3>
```

```
<!-- 待って！2番目のネストレベルはどこに？-->
```



# セクションの構造



# 見出しのレベルとスタイルを分離するのはどうか？

```
<h2 class="h3">私は何でしょう？</h2>
```



# インクルーシブ＝体験を等価なものにすること

コンテンツを視覚的に扱う一方で、非視覚的には別の方法でコンテンツを処理するということは、見えているユーザーと見えていないユーザーを**分離し始めている**ということです。

RSSリーダーなどを通じて表示される際には、異なるスタイルシートが適用されるため、**視覚的な構造が異なるものになる**こともあります。

# 見出しジャンプで読み飛ばされる情報に注意する

スクリーンリーダー見出しナビゲーション

—— 公開日 2017/12/12 ——

**インクルーシブなブログ記事のマークアップ方法**

# <article>要素は使うべきなのか？

```
<main id="main">
```

```
<article>
```

```
<h1>インクルーシブなブログ記事のマークアップ方法</h1>
```

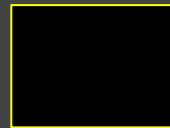
```
<div class="meta">公開日 <time
```

```
datetime="2017-12-12">2017/12/12</time></div>
```

```
<!-- ここに記事のコンテンツ -->
```

```
</article>
```

```
</main>
```



# JAWSとiOS VoiceOverは開始・終了を読み上げる

```
<article>
```

```
ここに最初の記事のコンテンツ
```

```
</article>
```

```
<article>
```

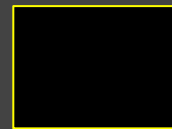
```
ここに2番目の記事のコンテンツ
```

```
</article>
```

```
<article>
```

```
ここに3番目の記事のコンテンツ
```

```
</article>
```



# 要素を選ぶときは、ユーザー体験の観点から

セマンティックな要素を選ぶときは、  
**ユーザー体験の観点**から考えるようにしましょう。

技術的に正しい要素を使用しているも、  
それが全く**サポート**されておらず、  
誰が見ても**何の効果も発揮**しないことがあります。

また、効果が**間違**いなく発揮される場合でも、混乱を招いたり、  
一貫性がなかったり、じゃまだったりする場合があります。

# 見出しのテキストにも注意(NGな例)

```
<h2>無料ですって？それならください！</h2>
```

# 何が無料なのでしょう？

ダイレクトで説明的な見出しは、あとに続く内容を明確にし、全体的な理解も助けます。言い換えれば、もったいぶったり、風変わりにしたりしても、**ろくなことがない**ということです！

さらに、スクリーンリーダーユーザーにとっては別の意味合いもあります。

スクリーンリーダーは、動的に**見出しをピックアップし**、**選択できるリストとして提供**します。

## 見出しのテキスト(OKな例)

```
<h2>フラップジャックが無料ですって?ください!</h2>
```



# ブログ記事のまとめ

- 見出しはとても重要
  - 見出しレベルに配慮する、見出しのテキストも重要
- 見た目とマークアップが食い違うのは要注意
  - ビジュアルブラウザとスクリーンリーダーの体験が異なるものになる
- ユーザー体験を重視する
  - 仕様で許されるかという観点だけでなく、  
実際にサポートされているか、ユーザーに使えるかを重視

まとめ

# このセッションのまとめ

- インクルーシブ = アクセシビリティとほぼ同義
- ボタンは `<button>` で
  - `<div>`で頑張るのはダサい、アイコンにはさまざまな実装方法あり
- ナビゲーションをインクルーシブに
  - スクリーンリーダーにも伝わる現在位置の表現
- ブログ記事では見出しが重要
  - 見出しの構造やテキストに配慮する

# Amazonで「インクルーシブHTML」と検索！

amazon.co.jp  
プライムを始める

本 ▾



プライム・ビデオ  
映画・TV番組見放題！

30日間  
無料体験を試す

カテゴリー ▾

Amazonポイント: 残高を確認

マイストア タイムセール ギフト券



こんにちは。サインイン  
アカウント&リスト ▾

注文履歴

今すぐ登録  
プライム ▾



本 詳細検索 ジャンル一覧 ▾ 新刊・予約 ▾ Amazonランキング ▾ コミック・ラノベ・BL ▾ 雑誌 ▾ 文庫・新書 ▾ Prime Student バーゲン ▾

本 > コンピュータ・IT > Web作成・開発



## インクルーシブHTML+CSS & JavaScript

多様なユーザーニーズに応えるフロントエン

ドデザインパターン 単行本 (ソフトカバ

ー) - 2017/11/4

Heydon Pickering (著), & 3 その他

カスタマーレビューを書きませんか？

その他 ( ) の形式およびエディションを表示する

単行本 (ソフトカバー)

¥ 3,024

¥ 3,024 より 1 新品

シェアする

¥ 3,024

ポイント: 91pt (3%)

詳細はこちら

通常配送無料 詳細

在庫あり。 在庫状況について

この商品は、Amazon.co.jp が販売、発送します。ギフトラッピングを利用できます。

数量: 1

この注文でお急ぎ便、お届け日時指定便を無料体験 Amazon プラ



# Amazonで「インクルーシブHTML」と検索！

amazon.co.jp  
プライムを始める

本 ▾



プライム・ビデオ  
映画・TV番組見放題！

30日間  
無料体験を試す

カテゴリー ▾

Amazonポイント: 残高を確認

マイストア タイムセール ギフト券



こんにちは。サインイン  
アカウント&リスト ▾

注文履歴

今すぐ登録  
プライム ▾



本 詳細検索 ジャンル一覧 ▾ 新刊・予約 ▾ Amazonランキング ▾ コミック・ラノベ・BL ▾ 雑誌 ▾ 文庫・新書 ▾ Prime Student バーゲン ▾

## amazon ランキング

Web開発のラン...

売れ筋ランキン...

すべてのカテゴリ

本

コンピュータ・IT

Web作成・開発

ホームページ入門書

Web構築・管理

Webプログラミング

Webデザイン

### Web開発 の 新着ランキング

新着アイテムおよび予約商品のベストセラー。ランキングは1時間ごとに更新されます。



1.

インクルーシブHTML+CSS & JavaScript 多様なユーザーニーズに応え...

Heydon Pickering  
単行本 (ソフトカバー)

¥ 3,024 ✓prime

発売日: 2017/11/4



2.



ネットで「女...」  
単行本 (ソフト  
カバー)

¥ 1,620

✓prime

ほしい物ランキン...

明日から始める  
家計の見直し

できる

# Amazonで「インクルーシブHTML」と検索！

## Web開発 の 新着ランキング

新着アイテムおよび予約商品のベストセラー。ランキングは1時間ごとに更新されます。



1.

インクルーシブHTML+CSS & JavaScript 多様なユーザー  
フロントエン...

Heydon Pickering

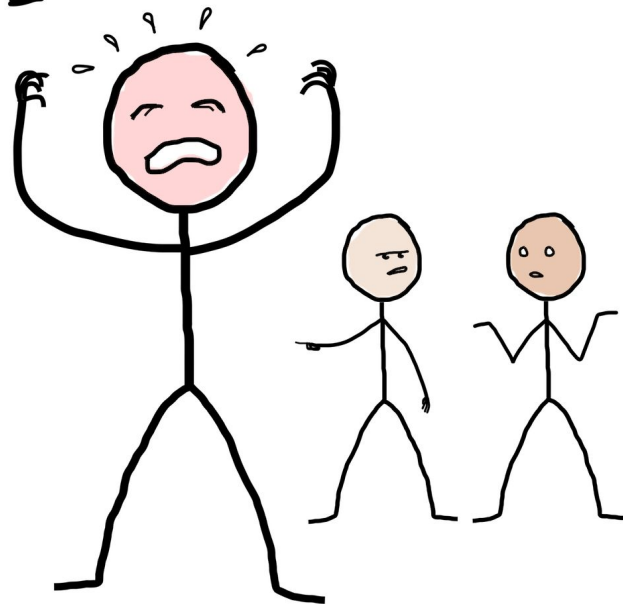
単行本（ソフトカバー）

¥ 3,024 ✓prime

発売日: 2017/11/4

今日からあなたも  
インクルーシブデザイナー！

I'M A DESIGNER!



“What’s their problem?”

“I dunno. All I asked them is where the toilets are.”

ありがとうございました

@bakera

@magi1125

スクリーンリーダー: VoiceOver (iOS 11.1)