2017年6月10日 12:45〜
倉敷物語館　会議室

第7回 okayama-js

実践 WAI-ARIA

自己紹介

SAWADA STANDARD DESIGN

澤田 望

パーマ

メガネ（老眼）

ガリガリ

中年男性

岡山で独立（2014）　Webアクセシビリティ検証／
イラスト制作／
岡山情報ビジネス学院 非常勤講師
などなど

キヤノン　（～2013）　サイトデザイン監修／
Webアクセシビリティ対応

WAIC　（2011～）　（ウェブアクセシビリティ基盤委員会）
WG2（実装）／WG4（翻訳）委員

今日の目標

アクセシブルではないマークアップ
に対して、WAI-ARIAの何を足せば
アクセシブルになるの？

⇧
ここの解決

WAI-ARIAとは

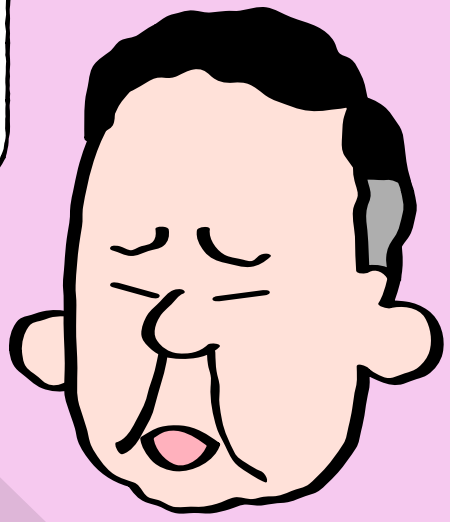
WAI-ARIAとは

Web
Accessibility
Initiative

Accessible
Rich
Internet
Applications　の略

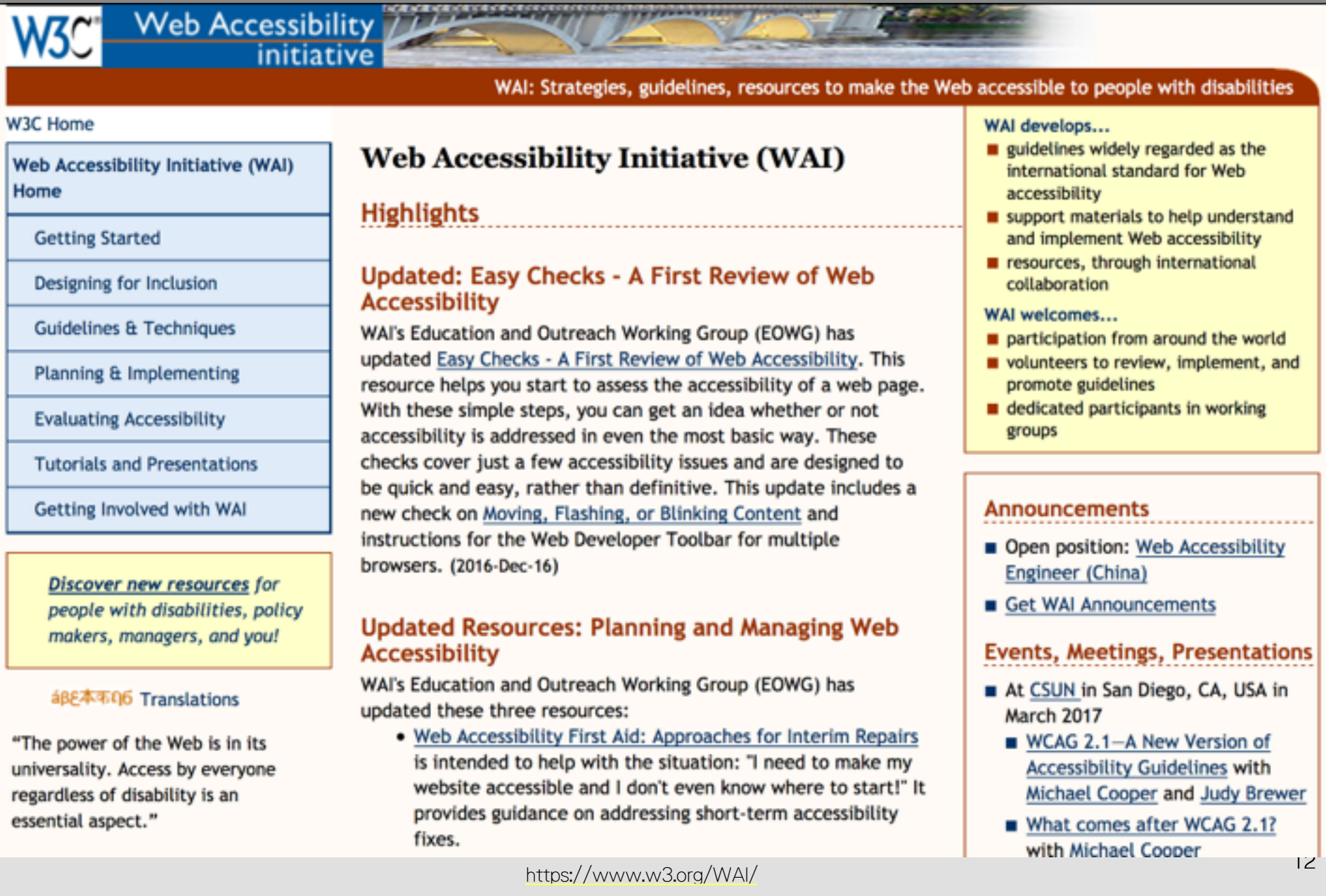
WAI-ARIA

わい・エリア



WAI-ARIA

WAI



https://www.w3.org/WAI/

大藤さんの説明

W3Cにおいてアクセシビリティ関連の活動を行っている部門のWAIによって作成された、

アクセシブルな

リッチ・

インターネット・

アプリケーション

を制作するための

（要素と属性を追加する）仕様

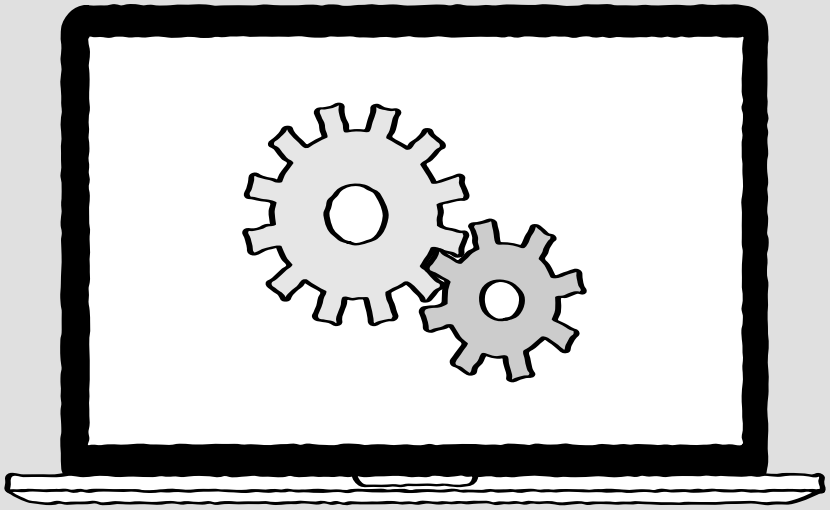
わかりやすい「WAI-ARIA 1.0」仕様解説書

なぜ仕様を追加する必要があるのか？

HTMLは元々文書をマークアップするための言語

なぜ仕様を追加する必要があるのか？

- JavaScript／CSS等で複雑になったWebアプリケーションの構造
- 「何が／いつ／どう」変化したかという情報



HTMLの要素や属性だけでは表現しきれない

なぜ仕様を追加する必要があるのか？

WAI-ARIAの〇〇〇属性を付加

＋

- role属性
- aria-〇〇〇属性
- tabindex属性

（tabindex自体はHTMLの仕様）

構造／変化をUAに伝えられるようになる

WAI-ARIAのイメージ

```
<div>動画プレーヤー</div>
```

⇩ role属性の追加

```
<div role="dialog">動画プレーヤー</div>
```

⇩ UAが解釈するイメージ

```
<dialog>動画プレーヤー</dialog>
```

WAI-ARIAを使うと何が変わるのか？

- UAが正しく解釈し、正しく振る舞うようになる
- スクリーンリーダー利用者に正しく伝わる
- CSSやJavaScriptを使った開発が楽になる＊

ちなみに電子書籍にも応用可能（らしい）

W3C Candidate Recommendation

Digital Publishing WAI-ARIA Module 1.0

W3C Candidate Recommendation 15 December 2016

This version:
https://www.w3.org/TR/2016/CR-dpub-aria-1.0-20161215/

Latest published version:
https://www.w3.org/TR/dpub-aria-1.0/

Latest editor's draft:
http://w3c.github.io/aria/aria/dpub.html

Implementation report:
http://w3c.github.io/test-results/dpub-aria/README.md

Previous version:
https://www.w3.org/TR/2016/WD-dpub-aria-1.0-20160317/

Editors:
Matt Garrish, DAISY Consortium
Tzviya Siegman, Wiley, tsiegman@wiley.com
Markus Gylling, DAISY Consortium, mgylling@daisy.org
Shane McCarron, Spec-Ops, shane@spec-ops.io

Copyright © 2015-2016 W3C® (MIT, ERCIM, Keio, Beihang). W3C liability, trademark and document use rules apply.

Abstract

Accessibility of web content requires semantic information about widgets, structures and behaviors in order to allow assistive technologies to convey appropriate information to persons with disabilities. This specification defines a [WAI-ARIA] module encompassing an ontology of roles, states and properties specific to the digital publishing industry. These semantics are designed to allow an author to convey user interface behaviors and structural information to assistive technologies and to enable semantic navigation, styling and interactive features

https://www.w3.org/TR/dpub-aria-1.0/

WAI-ARIAの仕様

WAI-ARIA 1.0：勧告

W3C Recommendation

Accessible Rich Internet Applications (WAI-ARIA) 1.0

W3C Recommendation 20 March 2014

This version:
http://www.w3.org/TR/2014/REC-wai-aria-20140320/
Latest version:
http://www.w3.org/TR/wai-aria/
Previous version:
http://www.w3.org/TR/2014/PR-wai-aria-20140206/
Editors:
James Craig, Apple Inc.
Michael Cooper, W3C
Previous Editors:
Lisa Pappas, Society for Technical Communication
Rich Schwerdtfeger, IBM
Lisa Seeman, UB Access

出版以来の報告されたエラーや問題に対するエラッタを確認されたい。

この文書はまた、単一ページのバージョンとして利用可能である。

翻訳も参照のこと【訳注：これは日本語非公式訳です】。

https://momdo.github.io/wai-aria/

WAI-ARIA 1.1：勧告候補

W3C Candidate Recommendation

Accessible Rich Internet Applications (WAI-ARIA) 1.1 日本語訳

W3C Candidate Recommendation 27 October 2016

This version:
https://www.w3.org/TR/2016/CR-wai-aria-1.1-20161027/

Latest published version:
https://www.w3.org/TR/wai-aria-1.1/

Latest editor's draft:
http://w3c.github.io/aria/aria/aria.html

Implementation report:
http://w3c.github.io/test-results/wai-aria/all.html

Previous version:
https://www.w3.org/TR/2016/WD-wai-aria-1.1-20160721/

Latest Recommendation:
https://www.w3.org/TR/2014/REC-wai-aria-20140320/

Editors:
Joanmarie Diggs, Igalia, S.L., jdiggs@igalia.com
James Craig, Apple Inc., jcraig@apple.com

https://momdo.github.io/wai-aria-1.1/

WAI-ARIAで定義されている仕様

role属性　：82種

（WAI-ARIAロール）　（70種使用可）

aria-〇〇〇属性　：67種

（WAI-ARIAステートおよびプロパティ）

tabindex属性

（tabindex自体はHTMLの仕様）

WAI-ARIA 1.0から1.1へのおもな変更点

追加 role属性

- cell
- feed
- figure
- none
- searchbox
- switch
- table
- term

aria-〇〇〇属性

- aria-colcount
- aria-colindex
- aria-colspan
- aria-current
- aria-details
- aria-errormessage
- aria-keyshortcuts
- aria-modal
- aria-placeholder
- aria-roledescription
- aria-rowcount
- aria-rowindex
- aria-rowspan

非推奨

aria-〇〇〇属性

- aria-grabbed
- aria-dropeffect

role属性

（WAI-ARIAロール）

role属性

- 要素の意味や役割を定義するための属性
- 既存の要素の役割も上書きして変更できる

role属性の分類

- ランドマーク・ロール　：　8種
- 文書構造ロール　：26種
- ウィジェット・ロール　：36種
- 抽象ロール（使用禁止）　：12種

ランドマーク・ロール

- UAや支援技術が行うナビゲーションの重要な目印（ランドマーク）として使用するためのロール

ランドマーク・ロール

- banner（header要素）
- complementary（aside要素）
- contentinfo（footer要素）
- form（form要素）
- main（main要素）
- navigation（nav要素）
- region
- search

※applicationは文書構造ロールへ移動

ランドマーク・ロールの注意点など

- HTML5の場合は同じロールの要素が既に存在するため（regionとsearchを除く）、そのまま使うのが〇
- banner／main／contentinfoはdocumentロールまたはapplicationロールの内部にひとつしか配置できない

ランドマーク・ロールの例

role="banner"

<h1>

D2 DEV&DESIGN DRAFT

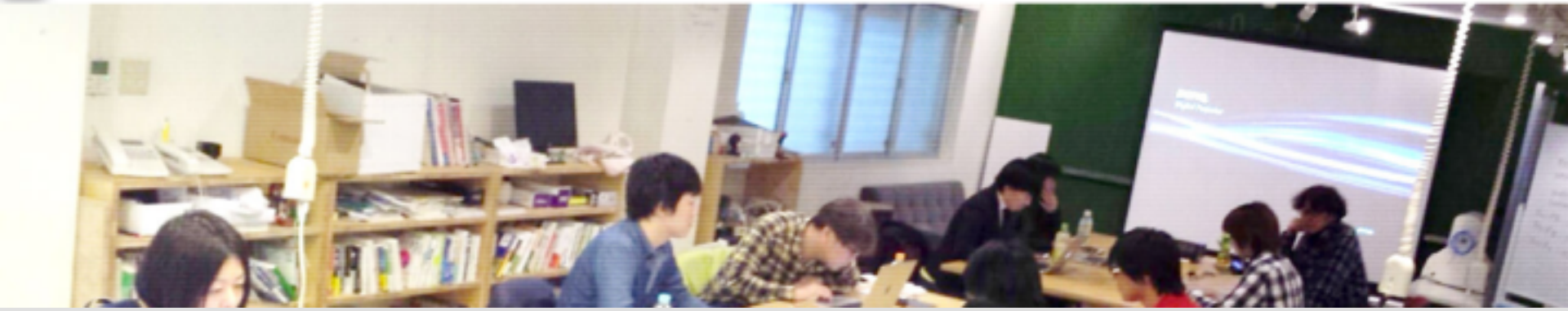
Fork me on GitHub

role="navigation"

D2DRAFTって？ ▼ | イベント ▼ | イベント一覧 ▶ | お問合せ ▼

role="main"

<p>



http://d2draft.net/

mainロールの実装例

```
<div>～</div>

<div role="main">～</div>

<main>～</main>

<main role="main">～</main>
```

VoiceOverで聞いてみよう

http://sawada-std-design.com/works/d2d-study/accessibility/wai-aria/role-landmark.html

文書構造ロール

- 文書構造を示すためのロール
- 文書構造的なセマンティクス（意味）を持たない要素にセマンティクスを定義したいとき、もしくは元の要素のセマンティクスを変えたいときに使用
- あくまで「定義」だけなので、見た目の変更はCSSが必要

文書構造ロール

- application
- article（article要素）
- cell
- columnheader
- definition
- directory
- document（body要素）
- feed
- figure
- group
- heading（h1～6要素）
- img（img要素）
- list（ul・ol要素）
- listitem（li要素）
- math
- none
- note
- presentation
- region（section要素）
- row（tr要素）
- rowgroup（thead・tbody・tfoot要素）
- rowheader
- separator（hr要素）
- table
- term
- toolbar

文書構造ロールの注意点など

- applicationロールは必ずアプリケーション全体を含む領域に対して設定（アクセシブルネームが必要）
- directoryロールは静的なリスト（動的：treeロール）
- presentation／noneロールはセマンティクスを削除（今後はnoneに移行）
- アプリケーション内に複数のtoolbarロールを含む場合はラベルが必要

文書構造ロールの例

role="main"

role="button"

role="button" 評価

role="button" 時間

role="link"

role="heading" 福太郎

4.1 ★★★★☆ (247) · お好み焼き店

大阪らしい空気が漂う、ねぎ焼・お好み焼・鉄板焼の店

２丁目３−１７

17:00開店

role="link"

role="heading" 美津の

3.5 ★★★½☆ (159) · お好み焼き店

バラエティ豊富な鉄板焼き料理が楽しめるお好み焼き店

１丁目４−１５

カジュアル · グループ

role="link"

role="heading" 味乃家

4.3 ★★★★½ (199) · お好み焼き店

創業60年以上の、焼き方にこだわる老舗お好み焼き店

https://www.google.co.jp/

文書構造ロールの例

role="link"

role="heading"美津の
3.5 ★★★½☆ (159) · お好み焼き店
バラエティ豊富な鉄板焼き料理が楽しめるお好み焼き店
１丁目４−１５
カジュアル · グループ

Firebug - 大阪市 お好み焼き - Google 検索

CSS スクリプト DOM ネット Cooki

div._M6k.... div.r-i0in... div div._Nld div#rso

```
S4IOTAA">
▼ <div class="_iPk _Ml">
  ▶ <div class="_li">
      <div class="_rl" aria-
      level="3" role="heading">美津の</div>
    ▶ <span class="rllt__details lqhpac">
    </div>
  </a>
</div>
</div>
</div>
```

ウィジェット・ロール

- ユーザーインターフェイス部品として機能する各種ウィジェットを示すためのロール
- あくまで「示す」ものなので、機能させるためにはJavaScriptが必要

ウィジェット・ロール

- alert
- alertdialog
- button
- checkbox
- combobox
- dialog
- grid（table要素）
- gridcell（td要素）
- link
- listbox（select要素）
- log
- marquee
- menu
- menubar
- menuitem
- menuitemcheckbox
- menuitemradio
- option（option要素）
- progressbar
- radio
- radiogroup
- scrollbar
- searchbox
- slider
- spinbutton
- status
- switch
- tab
- tablist
- tabpanel
- textbox
- timer
- tooltip
- tree
- treegrid
- treeitem

ウィジェット・ロールの注意点など

- textboxロールはENTER時の改行／送信に注意（aria-multiline属性）
- comboboxロールのコンボボックスを編集可能にする場合はaria-autocomplete属性を指定
- tabとtabpanelの関連付けには、tab側にaria-controls属性を指定するか、またはtabpanel側にaria-labelledby属性を指定

ウィジェット・ロールの注意点など

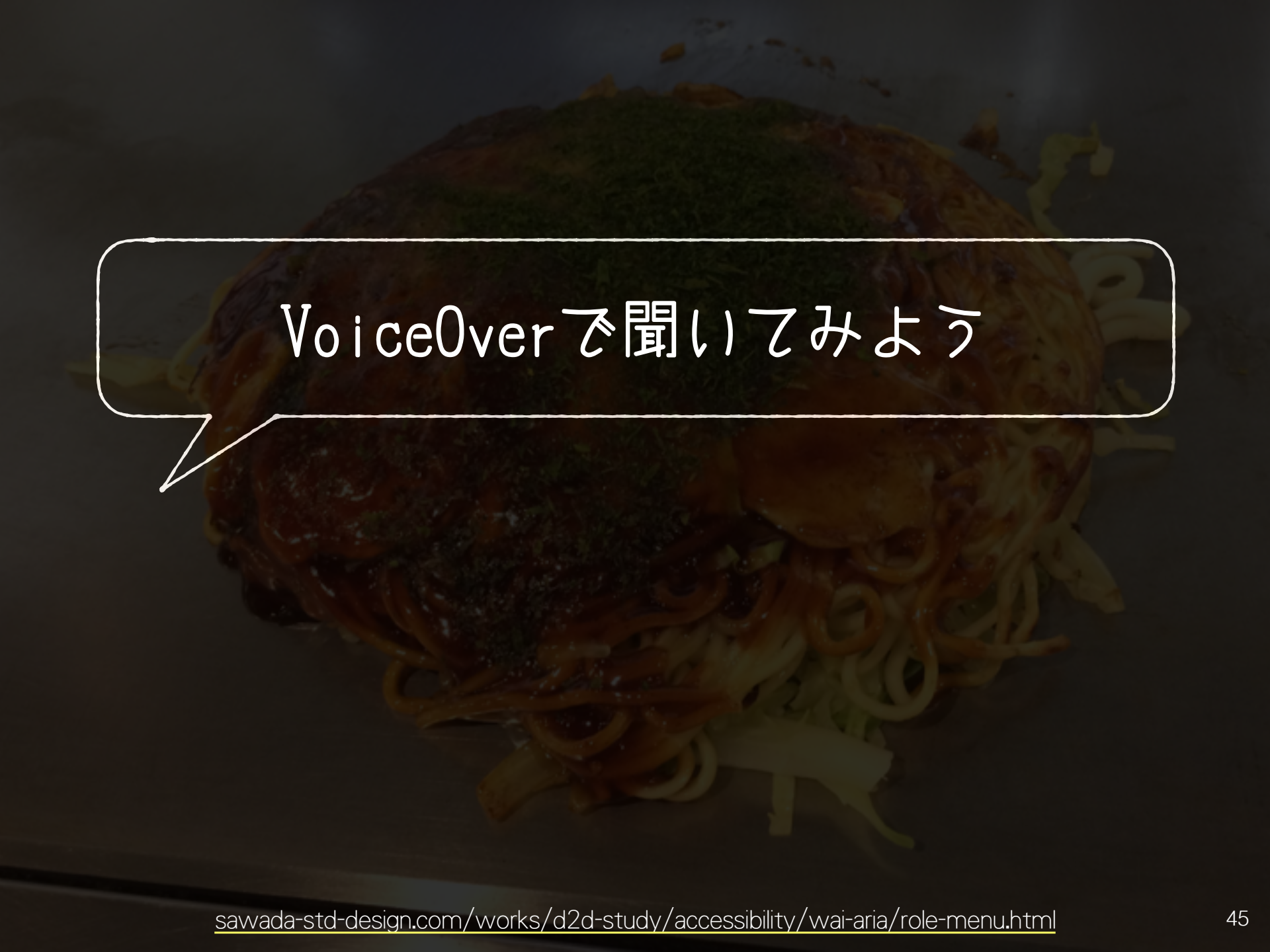
- ページ遷移かフォーカスが移動する場合：link
  それ以外の場合：button
- dialogロールのダイアログにはラベルが必須
  （aria-label属性やaria-labelledby属性でも可）
- アラートを閉じない場合：alert
  アラートを閉じる場合：alertdialog

ウィジェット・ロールの例

menuロールの実装例

```
<ul>
    <li>～</li>
    <li>～</a></li>
    <li>～</li>
</ul>

<ul role="menu">
    <li role="menuitem">～</li>
    <li role="menuitem">～</li>
    <li role="menuitem">～</li>
</ul>
```

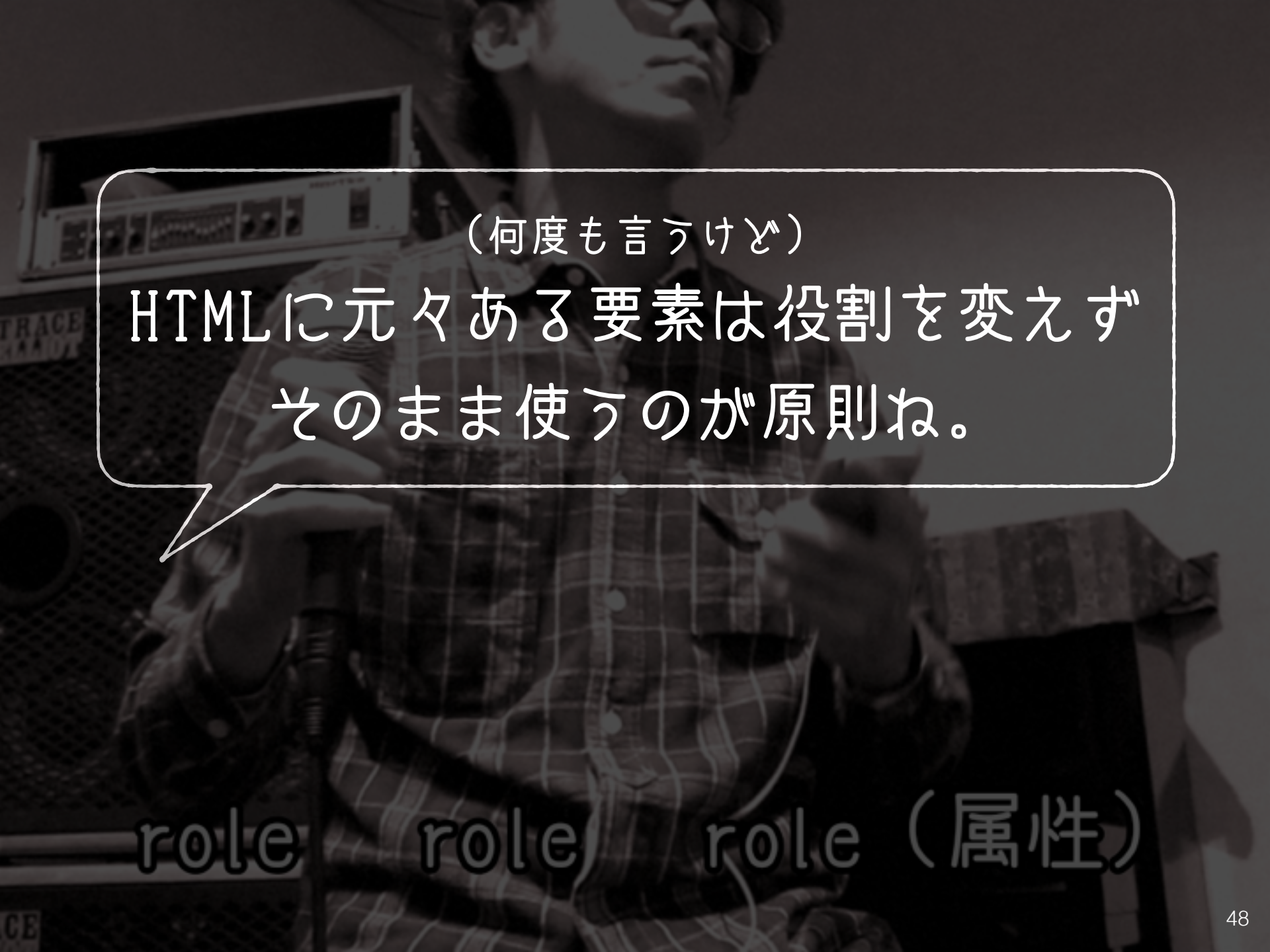


sawada-std-design.com/works/d2d-study/accessibility/wai-aria/role-menu.html

抽象ロール

- ロール全体の分類・定義を行うために用意された特殊なロールで、実際の制作においてrole属性の値として指定することはできない

role属性があれば何でもできそう！


（何度も言うけど）
HTMLに元々ある要素は役割を変えず
そのまま使うのが原則ね。
role
role
role（属性）

HTMLの要素に対するrole属性利用の可否

W3C Working Draft

ARIA in HTML 日本語訳

W3C Working Draft 11 November 2016

This version:
https://www.w3.org/TR/2016/WD-html-aria-20161111/

Latest published version:
https://www.w3.org/TR/html-aria/

Latest editor's draft:
https://w3c.github.io/html-aria/

Previous version:
https://www.w3.org/TR/2016/WD-html-aria-20161110/

Editor:
Steve Faulkner, The Paciello Group

Copyright © 2016 W3C® (MIT, ERCIM, Keio, Beihang). W3C liability, trademark and permissive document license rules apply.

概要

この仕様は、[HTML51]要素の[wai-aria-1.1]属性の使用に対するウェブ開発者の規則（著者適合性要件）を定義する。また、適合チェックツールに対する要件も定義する。

この文書の位置付け

https://momdo.github.io/html-aria/

HTMLの要素に対するrole属性利用の可否

W3C Working Draft

2. ARIA in HTMLの使用に対する文書適合性要件

次の表は、HTML文書でARIAマークアップを使用するための要素ごとの文書適合基準要件を規定し、HTML Accessibility API Mappings仕様で定義されるHTML要素に適用される暗黙のARIAセマンティックスについて説明する。1列目のセルでそれぞれの言語機能（要素または属性）は同じ行の2列目のセルで与えられるARIAセマンティックス（ロール、ステート、およびプロパティ）を意味する。

> 注
>
> 暗黙のARIAセマンティックスと一致するARIA `role`および/または`aria-*`属性の設定は不要であり、かつこのプロパティが既にブラウザーで設定されるので推奨されない。

> 注
>
> 次の表の (new) および (changed) マーカーは、（ARIA 1.1で）新しいものまたは（ARIA 1.0 / 1.1の間で）変更されたARIAロール、ステートおよびプロパティを示す。

HTML言語の機能によるARIA属性利用の規則

HTML言語の機能	暗黙のARIAセマンティックス - 使用すべきでない	使用してもよいARIAロール、ステートおよびプロパティ
		ロール：`button`, `checkbox`, `menuitem`, `menuitemcheckbox`, `menuitemradio`, `option` - (changed), `radio`, `switch` -

https://momdo.github.io/html-aria/#docconformance

HTMLの要素に対するrole属性利用の可否

HTML言語の機能によるARIA属性利用の規則

HTML言語の機能	暗黙のARIAセマンティックス - 使用すべきでない	使用してもよいARIAロール、ステートおよびプロパティ
hrefをもつa要素	role=link	ロール：button, checkbox, menuitem, menuitemcheckbox, menuitemradio, option - (changed), radio, switch - (new), tabまたは treeitem グローバルaria-*属性および許可されるロールに受け入れ可能なすべてのaria-*属性およびlinkロール
hrefをもたないa要素	対応するロールなし	すべてのrole グローバルaria-*属性および許可されるロールに受け入れ可能なすべてのaria-*属性
address	role=contentinfo	グローバルaria-*属性

aria-〇〇〇属性

（WAI-ARIAステートおよびプロパティ）

aria-〇〇〇属性

- オブジェクトの状態（state）や性質／特性（property）を示す属性
- ロールとの組み合わせで、ユーザーに「何が／いつ／どう」変化したのかを通知できる

aria-〇〇〇属性の分類

- グローバル・ステート＆プロパティ属性：21種
- 関係属性　：16種
- ウィジェット属性　：24種
- ライブ領域属性　：　4種
- ドラッグアンドドロップ属性　：　2種

グローバル・ステート&プロパティ属性

- すべてのHTML要素に使用可能
- すべてのロールに使用可能

グローバル・ステート&プロパティ属性

- aria-atomic
- aria-busy（ステート）
- aria-controls
- aria-current（ステート）
- aria-describedby
- aria-details
- aria-disabled（ステート）
- aria-dropeffect
- aria-errormessage
- aria-flowto
- aria-grabbed（ステート）
- aria-haspopup
- aria-hidden（ステート）
- aria-invalid（ステート）
- aria-keyshortcuts
- aria-label
- aria-labelledby
- aria-live
- aria-owns
- aria-relevant
- aria-roledescription

グローバル・ステート&プロパティ属性の例

https://www.mitsue.co.jp/service/

関係属性

- 要素と要素の関係を明確に示すために使用される属性
- 文書構造からだけでは解釈しにくい関係性を定義

  （例：マークアップ順ではない読み上げ順）

関係属性

- aria-activedescendant
- aria-colcount
- aria-colindex
- aria-colspan
- aria-controls
- aria-describedby
- aria-details
- aria-errormessage
- aria-flowto
- aria-labelledby
- aria-owns
- aria-posinset
- aria-rowcount
- aria-rowindex
- aria-rowspan
- aria-setsize

関係属性の注意点など

- aria-labelledby属性は複数指定可能（半角スペース区切り）
- ラベルを画面に表示できる場合：aria-labelledby
  ラベルを画面に表示できない場合：aria-label
- 短いラベルの場合：aria-labelledby
  文章で説明する場合：aria-describedby

aria-describedby属性を追加した例

aria-labelledby属性を追加した実装例

```
<svg role="img" aria-labelledby="ttlOko dscOko"
xmlns= “～">
<title id="ttlOko">タイトル</title>
<desc id="dscOko">説明文</desc>
```

ウィジェット属性

- ウィジェット・ロールの要素で使用される各種属性
- ユーザーからの入力やアクションを受け取るユーザーインターフェイス要素に固有の属性
- あくまで「示す」ものなので、機能させるためにはJavaScriptが必要（HTML5の属性で同等なものはそちらを使うのがオススメ）

ウィジェット属性

- aria-autocomplete
- aria-checked
- aria-disabled
- aria-errormessage
- aria-expanded
- aria-haspopup
- aria-hidden
- aria-invalid
- aria-label
- aria-level
- aria-modal
- aria-multiline
- aria-multiselectable
- aria-orientation
- aria-placeholder
- aria-pressed
- aria-readonly
- aria-required
- aria-selected
- aria-sort
- aria-valuemax
- aria-valuemin
- aria-valuenow
- aria-valuetext

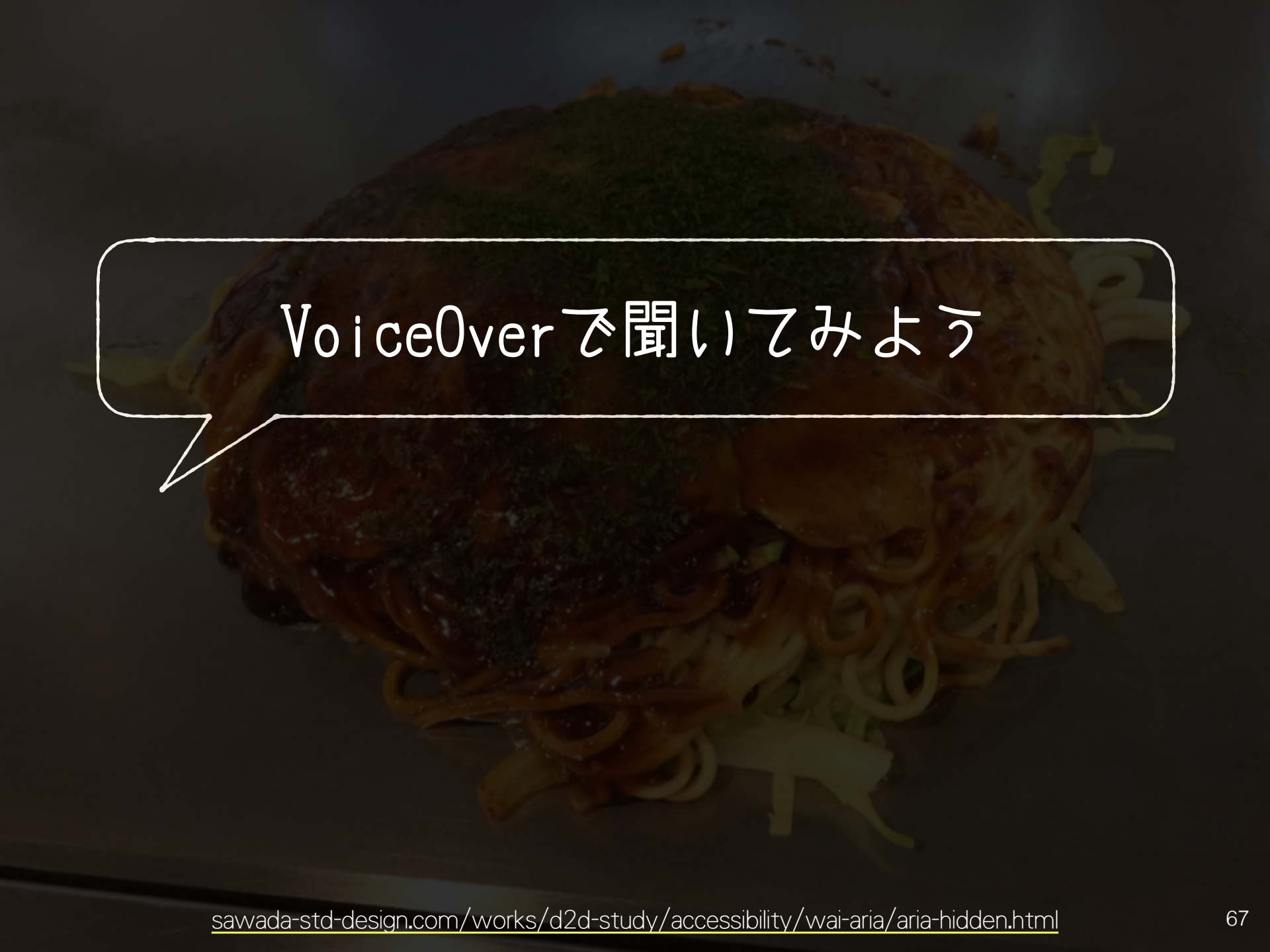
ウィジェット属性の注意点など

- aria-hidden属性の使用は、見えているコンテンツを支援技術（スクリーンリーダーなど）から隠すことで、支援技術利用者とそうでない利用者の体験価値が同等になる場合に限る（訳：気軽に隠すなよ）
- HTML5のrequired属性が使えるなら（スクリーンリーダーでの対応状況も）aria-required属性は不要

aria-hidden属性の実装例

```
<del>～</del>
<del aria-hidden="true">～</del>
```

※ del要素に対応していないスクリーンリーダーが多いため、読み上げさせないためにaria-hiddenを設置している例



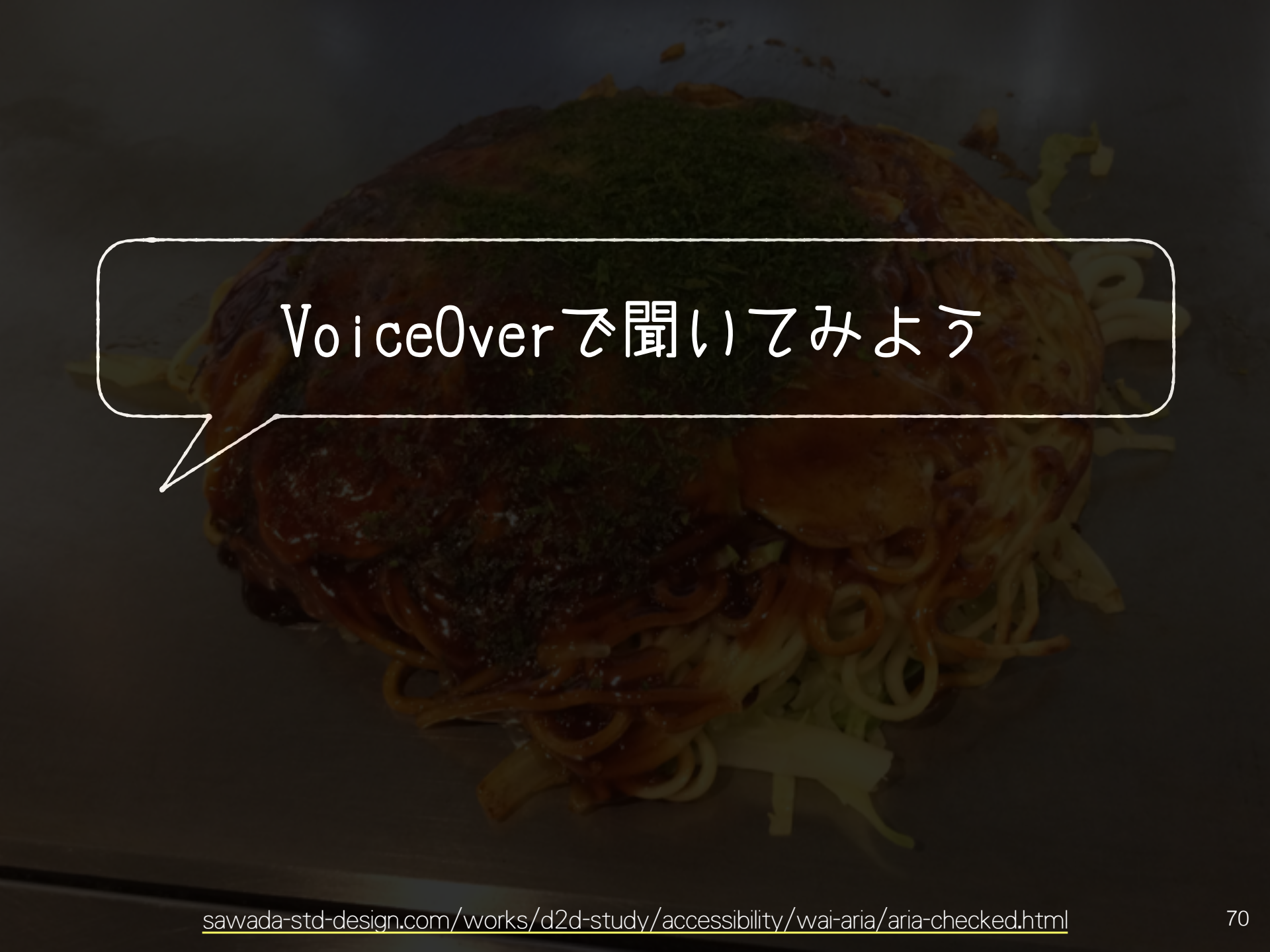
sawada-std-design.com/works/d2d-study/accessibility/wai-aria/aria-hidden.html

ウィジェット属性の注意点など

- aria-multiline属性：ENTER時の改行／送信に注意
- aria-haspopup属性でポップアップさせる場合は要素またはaria-owns属性で親子関係にする
- aria-expanded属性で展開／折り畳み行う場合は要素またはaria-controls属性で親子関係にする

aria-checked属性の実装例

```
<ul role="menu">
    <li role="menuitem checkbox" aria-checked="false">～</li>
    <li role="menuitem checkbox" aria-checked="true">～</li>
    <li role="menuitem checkbox" aria-checked="false">～</li>
</ul>
```



sawada-std-design.com/works/d2d-study/accessibility/wai-aria/aria-checked.html

ライブ領域属性

- リッチインターネットアプリケーションにおけるライブ領域（勝手に更新されるエリア）に固有の属性
- フォーカスの当たっていない要素に発生したコンテンツの更新情報も支援技術に提供できる

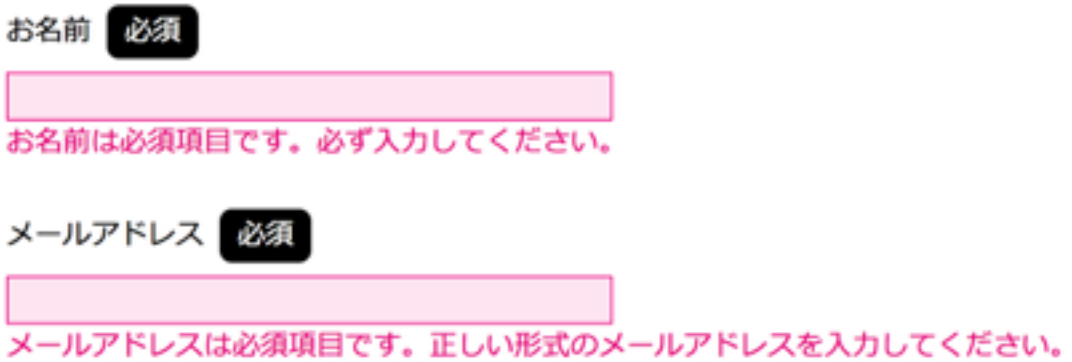
ライブ領域属性

- aria-atomic
- aria-busy
- aria-live
- aria-relevant

ライブ領域属性の注意点など

- aria-live属性値assertive：即座に更新を通知すべき重要度の高い情報

ライブ領域属性の例



Firebug - D2DRAFT "Dev & Design Draft" | D2DRAFT

コンソール HTML CSS スクリプト DOM ネット Cookie

編集 span#val... div form.has-... section#i... main body html

```
<div>
  <label for="name">
  <input id="name" class="error" name="name" value="" data-validation-error-msg-
  container="#validation_message_name" required="required" data-validation-error-
  msg="お名前は必須項目です。必ず入力してください。" data-
  validation="required" aria-invalid="true" style="" type="text">
  <span id="validation_message_name" class="error-alert
  has-error" role="alert" aria-atomic="true" aria-live="assertive" tabindex="-1">
    <span class="help-block form-error">お名前は必須項目です。必ず入力してくださ
    い。 </span>
  </span>
</div>
```

ドラッグアンドドロップ属性

- 支援技術がドラッグ＆ドロップ機能をサポートできるようにするための属性
- とはいえ、WAI-ARIA 1.1で非推奨に. . .
  （WAI-ARIAの将来のバージョンで新しい機能に置き換わることが期待される）

ドラッグアンドドロップ属性

- aria-dropeffect（非推奨）
- aria-grabbed（非推奨）

ロールに対するステート＆プロパティ利用の可否

W3C Working Draft

ARIA in HTML 日本語訳

W3C Working Draft 11 November 2016

This version:
https://www.w3.org/TR/2016/WD-html-aria-20161111/

Latest published version:
https://www.w3.org/TR/html-aria/

Latest editor's draft:
https://w3c.github.io/html-aria/

Previous version:
https://www.w3.org/TR/2016/WD-html-aria-20161110/

Editor:
Steve Faulkner, The Paciello Group

Copyright © 2016 W3C® (MIT, ERCIM, Keio, Beihang). W3C liability, trademark and permissive document license rules apply.

概要

この仕様は、[HTML51]要素の[wai-aria-1.1]属性の使用に対するウェブ開発者の規則（著者適合性要件）を定義する。また、適合チェックツールに対する要件も定義する。

この文書の位置付け

https://momdo.github.io/html-aria/

ロールに対するステート＆プロパティ利用の可否

W3C Working Draft

3. 許可されるARIAロール、ステートおよびプロパティ

この節は非規範的である。

> 注
>
> 以下の表は、HTMLで使用するために認められるARIAロール、ステートおよびプロパティに参考情報を提供する。すべてのARIAロール、ステートおよびプロパティは ARIA 1.1仕様で規範的に定義されている。表におけるARIAロール、ステートおよびプロパティへのリンクは、規範ARIA 1.1定義を参照する。

> 注
>
> 次の表の (new) および (changed) マーカーは、（ARIA 1.1で）新しいものまたは（ARIA 1.0 / 1.1の間で）変更されたARIAロール、ステートおよびプロパティを示す。

ARIAロール、ステートおよびプロパティ

ロール	説明	必須プロパティ	サポートされるプロパティ
			• aria-atomic • aria-busy（ステート） • aria-controls • aria-current（state）- (new) • aria-describedby

https://momdo.github.io/html-aria/#allowed-aria-roles-states-and-properties

ロールに対するステート＆プロパティ利用の可否

ARIAロール、ステートおよびプロパティ

ロール	説明	必須プロパティ	サポートされるプロパティ
すべて	ARIA***グローバルステートおよびプロパティ***は、任意のHTML要素で使用できる。	なし	～略（澤田）～
alert	重要かつ通常は時間依存の情報をもつlive regionの種類。関連するalertdialogとstatusを参照のこと。	なし	aria-expanded（ステート）
alerTDialog	初期のフォーカスがダイアログ内の要素に移動する場合、警告メッセージを含むダイアログの種類。関連するalertとdialogを参照のこと。	なし	• aria-expanded（ステート） • aria-modal - (new)
application	widgetロールによってサポートされる標準的な相互作用パターンに従わない、キーボードまたはジェスチャーイベントなどの、ユーザー入力を要求する1つ以上のフォーカス可能な要素を含む構造。	なし	aria-activedescendant
	文書、ページ、またはサイトの独		

tabindex属性

tabindex属性

- インタラクティブなオブジェクトはすべてフォーカス可能にしなければいけない
- 要素にフォーカスを与える属性としてtabindex属性を使用
- HTML5ではすべての要素にtabindexの指定が可能

tabindex属性の値

- tabindex=“正の整数” ：tabキーの移動順序
- tabindex=“0” ：フォーカス可能
  （順序はマークアップ順）
- tabindex=“-1” ：フォーカス不可

ライブ領域属性の例

お名前 必須

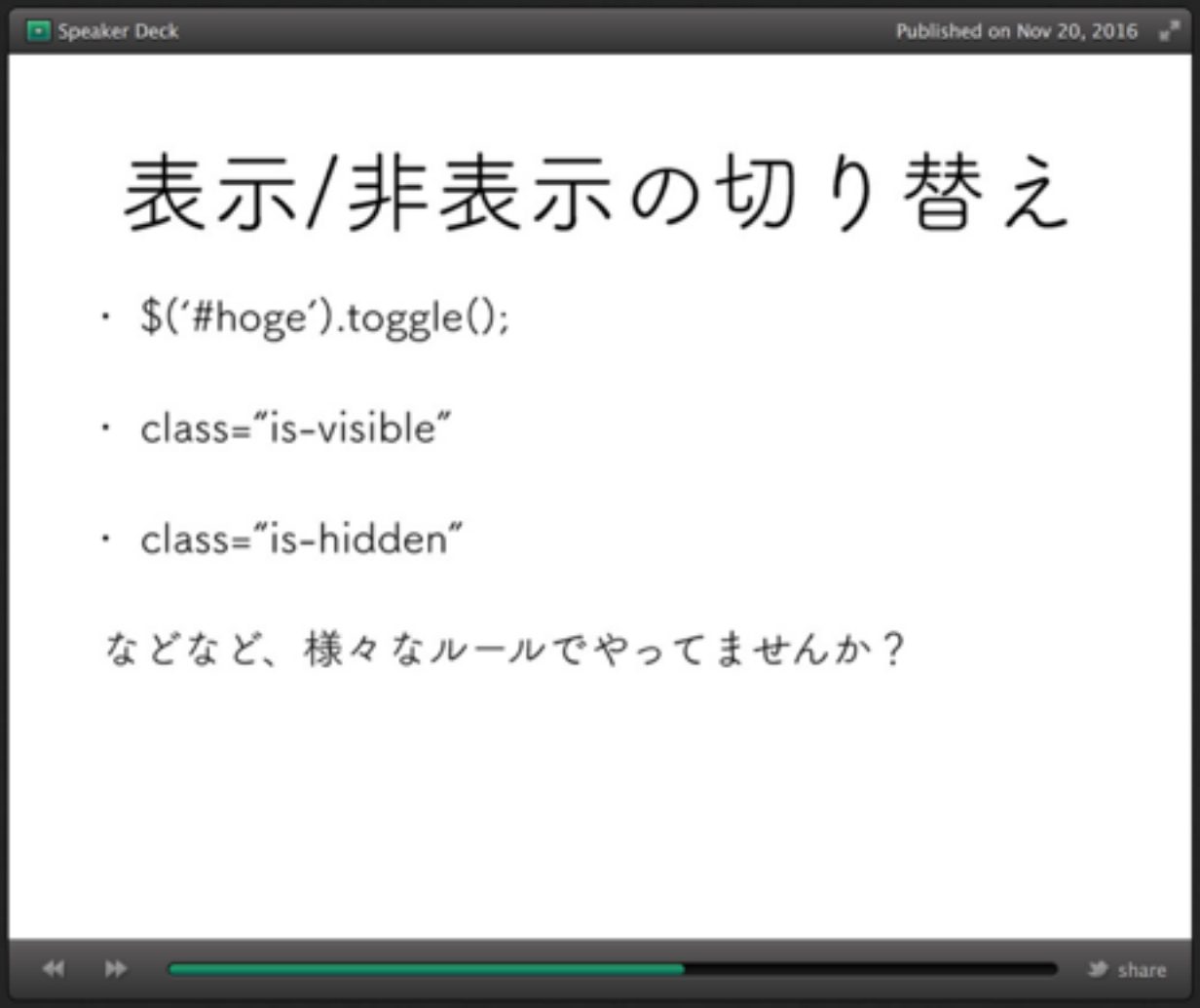
お名前は必須項目です。必ず入力してください。

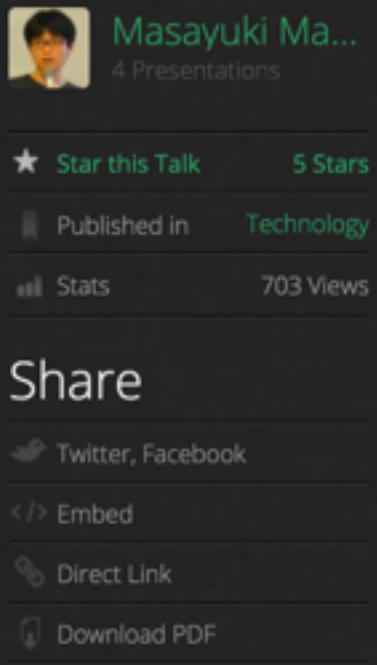
メールアドレス 必須

メールアドレスは必須項目です。正しい形式のメールアドレスを入力してください。

CSSの属性セレクタでの利用

WAI-ARIAを活用したJavaScriptによるDOM操作

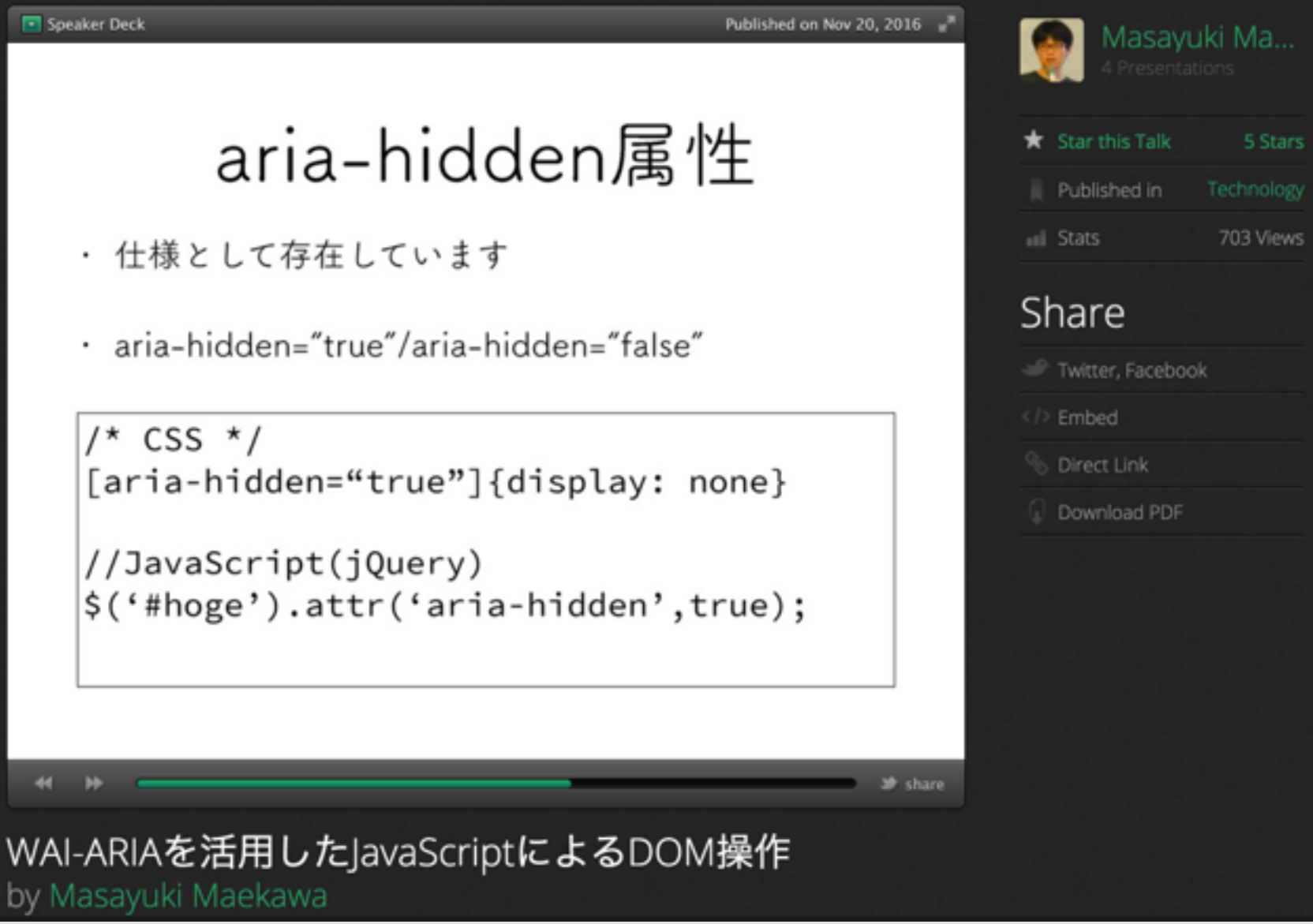




WAI-ARIAを活用したJavaScriptによるDOM操作
by Masayuki Maekawa

https://speakerdeck.com/maepon/wai-ariawohuo-yong-sitajavascriptniyorudomcao-zuo

CSSの属性セレクタについて

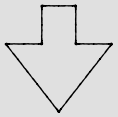


https://speakerdeck.com/maepon/wai-ariawohuo-yong-sitajavascriptniyorudomcao-zuo

CSSの属性セレクタについて

- 属性セレクタでrole属性やaria-〇〇〇属性が使用できる

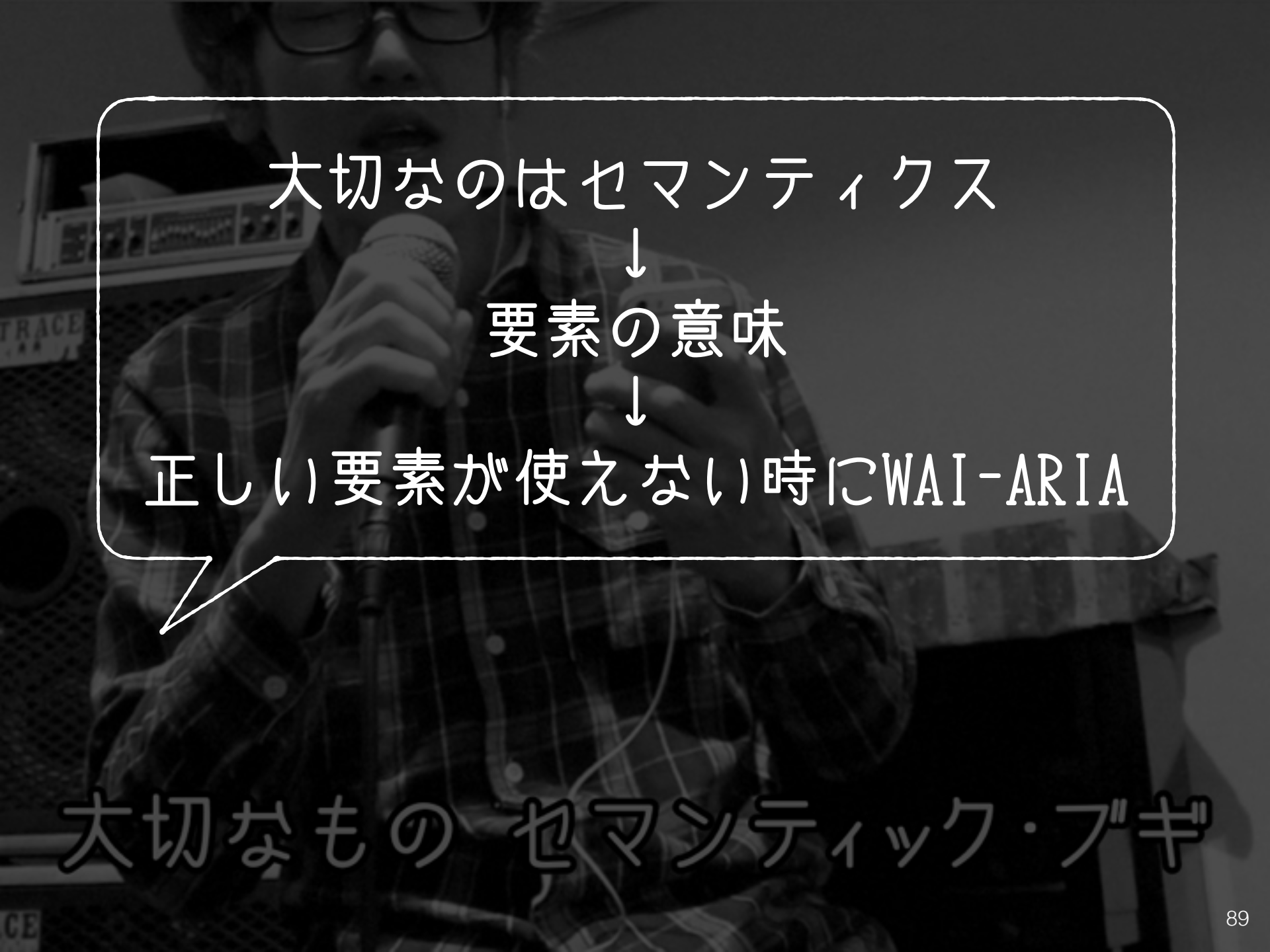
```
li[ aria-checked="true" ] { ～ ; }
```



CSS／JSの開発が楽になる？

sawada-std-design.com/works/d2d-study/accessibility/wai-aria/aria-checked-css.html

まとめ


大切なのはセマンティクス
↓
要素の意味
↓
正しい要素が使えない時にWAI-ARIA
大切なもの セマンティック・ブギ

サポート状況など

アクセシビリティ・サポーテッド（AS）情報

アクセシビリティ・サポーテッド（AS）情報

「アクセシビリティ・サポーテッド（AS）情報」について

この資料は、「アクセシビリティ・サポーテッド検証結果」をふまえて、それぞれの実装方法がアクセシビリティ・サポーテッドであるかどうか、つまり達成基準を満たすことのできる実装方法であるかどうかを判断するための参考資料として、作業部会2（実装）が公開するものです。

次の「アクセシビリティ・サポーテッド検証結果」をダウンロードする前に、こちらの資料を先にご覧いただくことを推奨します。

- 「アクセシビリティ・サポーテッド（AS）情報」 2014年6月版 HTML形式
- 「アクセシビリティ・サポーテッド（AS）情報」 2014年6月版 Excel形式（：41KB）

「アクセシビリティ・サポーテッド検証結果」について

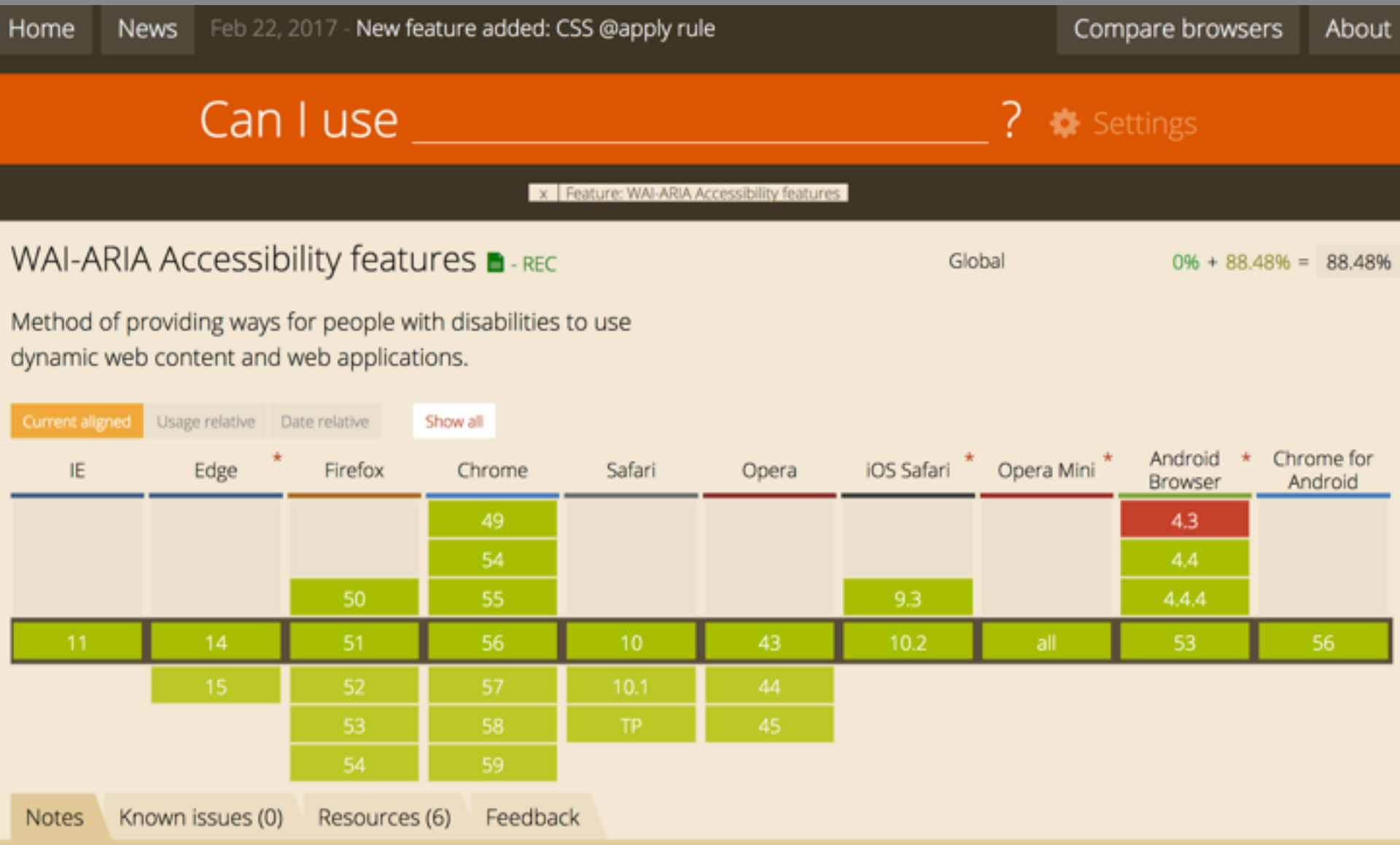
この資料は、主要なブラウザ及び支援技術を用いて各実装方法のサポート状況を独自に検証した結果を示す参考資料として、作業部会2（実装）が公開するものです。

サイト内関連リンク
- 新着情報
- 組織について
- 活動内容
- 公開資料&リンク集
  - JIS X 8341-3:2016 関連文書
  - JIS X 8341-3:2010 関連文書
  - WCAG 2.0 関連翻訳文書
  - アクセシビリティ・サポーテッド（AS）情報

http://waic.jp/docs/as/

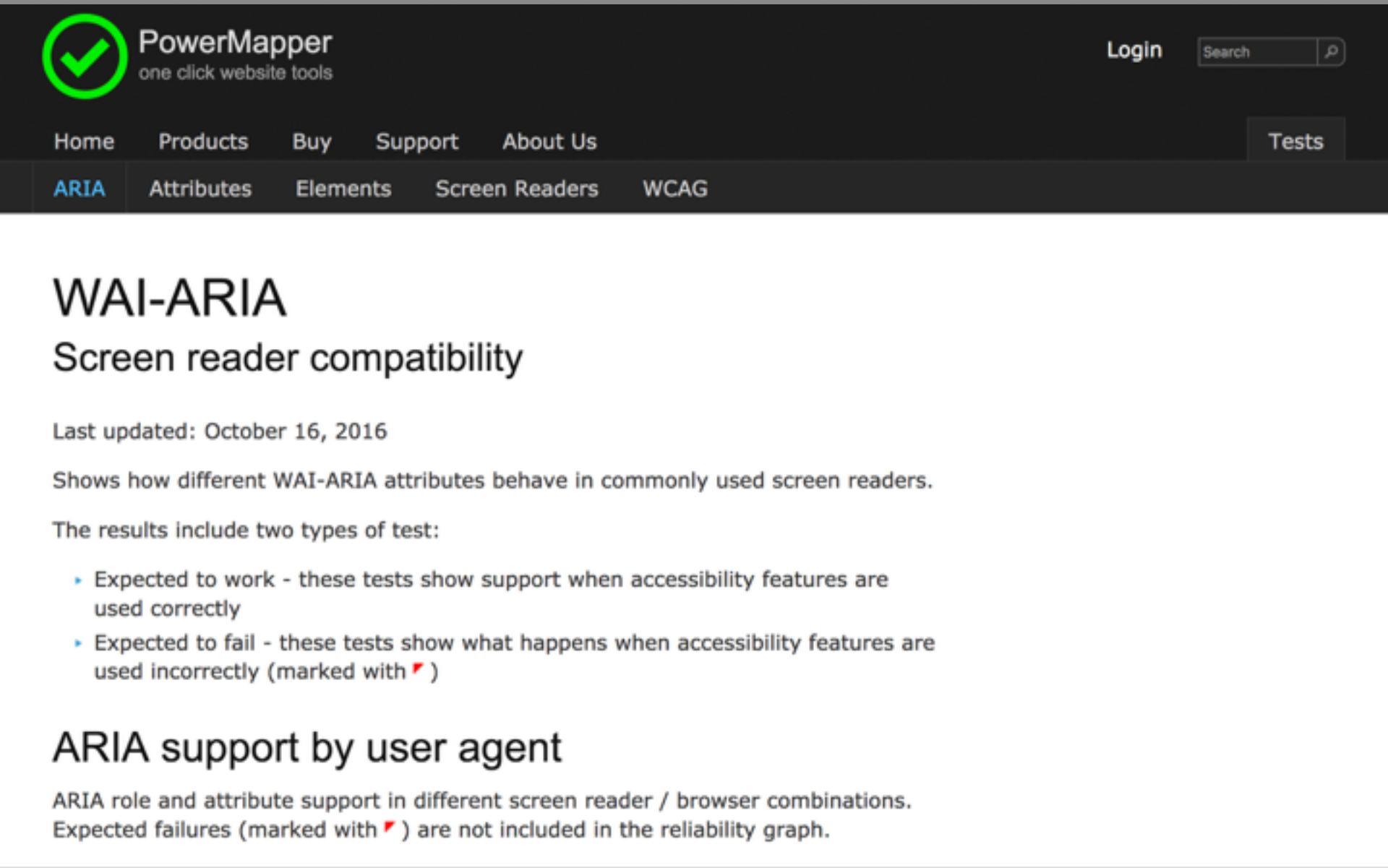
アクセシビリティ・サポーテッド情報の整備はまだまだ（未着手）

WAI-ARIAのサポート状況（参考）



http://caniuse.com/#feat=wai-aria

WAI-ARIAのサポート状況（参考）



https://www.powermapper.com/tests/screen-readers/aria/

WAI-ARIAのサポート状況（参考）

ARIA support by user agent

ARIA role and attribute support in different screen reader / browser combinations.
Expected failures (marked with ◤) are not included in the reliability graph.

The solid area in the graph shows percentage of tests that pass in all tested interaction modes. The cross hatched area shows partial passes that only work in some interaction modes. An example of a partial pass is when form labels are read when tabbing, but ignored in browse mode.

Combo	Versions	Reliability
JAWS IE	JAWS 17.0.2619 with IE11	79%
JAWS Firefox	JAWS 17.0.2619 with FF48	71%
NVDA IE	NVDA 2016.2 with IE11	64%
NVDA Firefox	NVDA 2016.2 with FF48	67%
VoiceOver Mac	VoiceOver OSX 10.11 with Safari 9.1.2	67%
VoiceOver iOS	VoiceOver iOS 9.1 with Safari iOS 9.1	55%
WindowEyes IE	WindowEyes 9.2 with IE11	74%
Dolphin IE	Dolphin SR 15.05 with IE11	52%
SaToGo IE	SaToGo 3.4.96.0 with IE11	25%
Average	Including older versions	64%

The average includes all versions, but some browser/AT combinations have tests for

https://www.powermapper.com/tests/screen-readers/aria/

WAI-ARIAのサポート状況（参考）

ARIA attributes

aria-describedby

Screen Reader	NVDA		JAWS		VoiceOver		Win-Eyes	Dolphin	SaToGo
Browser	IE	FF	IE	FF	Mac	iOS	IE	IE	IE
Reliability when used correctly (85% average)	100%	100%	75%	90%	67%	100%	63%	50%	100%
A "click here" link with aria-describedby attribute	✓	✓	✗	✓	✓	✓	✗	✗	✓
INPUT type=text with aria-describedby attribute	✓	✓	✓	✓	✓	✓	✓	✗	✓
IMG with aria-describedby	✗	✗	✗	✗	✓	✗	✓	✗	✗

aria-label

Screen Reader	NVDA		JAWS		VoiceOver		Win-Eyes	Dolphin	SaToGo
Browser	IE	FF	IE	FF	Mac	iOS	IE	IE	IE
Reliability when used correctly (65% average)	72%	69%	64%	64%	78%	58%	69%	56%	6%
A "click here" link with aria-label attribute	✓	✓	✓	✓	✓	✓	✓	✓	✗
APPLET with ARIA-LABEL	✓	✗	✓	✗	✗	✗	✗	✗	✗

https://www.powermapper.com/tests/screen-readers/aria/

とはいえ、日本での状況はまた
違います（PC-Talker一人勝ち）ので、
試しながら導入していきましょう。

PC-Talkerについてはこちら...

WAI-ARIAのサポート状況（参考）

White Stage

みるくがGeeklogテーマ作成・発表&アクセシビリティ情報を発信

2017年February25日(Sat) 22:12 JST

ダウンロード

リンク

お問い合せ

サイトマップ

WAI-ARIA対応のタブ型 UI を4種スクリーンリーダーで読み上げた検証結果1

OSとスクリーンリーダーのバージョンとブラウザの組み合わせは下記の通りになります。

- Windows7/NVDA2016.3jp/Firefox48.0.2
- Windows7/PC-Talker7/Internet Explorer11
- iOS 9.3.5/VoiceOver/Safari
- Android5.0.1/TalkBack5.0.4/Google Chrome53.0.2785.124

検証1　Tabs | jQuery UIの場合

jQuery UI - v1.12.1 - 2016-09-14 が使われているこちらのタブメニュー部分は、li要素にrole="tab"、a要素にrole="presentation"の記述になっています。また、li要素にはタブパネルdiv要素のidの値に関連付けられたaria-controls属性とa要素のidの値に関連付けられたaria-labelledby属性とaria-selected属性とaria-expanded属性が付いており、タブパネルにはa要素のidの値に関連付けられたaria-labelledby属性とaria-hidden属性が付いています。

読み上げ方	NVDA	PC-Talker	VoiceOver	TalkBack
上から続けて読み上げた場合	クリック可能 タブ 展開 選択 「Nunc tincidunt」 タブ 折り畳み「Proin dolor」 タブ 折り畳み「Aenean lacinia」 (選択したタブに関連する(表示されている)タブパネルのコンテンツ内容)Proin elit arcu, rutrum commodo…	「Nunc tincidunt」の選択 タブで決定「Nunc tincidunt」 「Nunc tincidunt」 「Proin dolor」の選択 タブで決定「Proin dolor」 「Proin dolor」 「Aenean lacinia」の選択 タブで決定「Aenean lacinia」 「Aenean lacinia」 (選択したタブに関連する(表示されている)タブパネルのコンテンツ内容)Proin elit arcu, rutrum commodo…	選択中の「Nunc tincidunt」タブ 全3個中1個 拡大 「Proin dolor」タブ 全3個中2個 隠されました 「Aenean lacinia」タブ 全3個中3個 隠されました (選択したタブに関連する(表示されている)タブパネルのコンテンツ内容)Proin elit arcu, rutrum commodo…	「Nunc tincidunt」 タブ 選択済 「Proin dolor」タブ 「Aenean lacinia」タブ (選択したタブに関連する(表示されている)タブパネルのコンテンツ内容)Proin elit arcu, rutrum commodo…
Tabと矢印キーで移動した或いはタップで読み上げた場合	タブコントロール「Nunc tincidunt」タブ 展開 選択 1-3 「Proin dolor」タブ 折り畳み 選択 2-3 展開 「Aenean lacinia」タブ 折り畳み 選択 3-3 展開 (選択したタブに関連する(表示されている)タブパネルのコン	続けて読み上げた場合と同じ結果	続けて読み上げた場合と同じ結果	タブリストに3個のアイテムが表示されてます 「Nunc tincidunt」 タブ 選択済 ダブルタップすると有効になります 「Proin dolor」タブ ダブルタップすると有効になります 「Aenean lacinia」タブ ダブ

https://white-stage.com/staticpages/index.php/a11y-tabmenu-sr1

ARIA Validator

https://chrome.google.com/webstore/detail/aria-validator/oigghlanfjgnkcndchmnlnmaojahnjoc

では、次は

グループワークの説明です！

の前に、ちょっと休憩。

楽しいワークショップ

グループ分け

人数が多いので、

5人／5チーム（A～E）

に分かれてください。

（最初から机の配置で分けておく？）

何をするのか？

JavaScriptを使ったサンプルコード
（jQueryですが）

- アコーディオンメニュー
- タブナビゲーション

にWAI-ARIAの属性を追加して、
さらにアクセシブルな状態にする

作業手順

1. jsdo.it にあるサンプルコードを
2. フォークして
3. みんなで相談して
4. 編集（WAI-ARIA付加）して
5. チーム代表のアカウントで保存して
6. 発表

（今回目指す）アクセシブルな状態とは？

1. 知覚／理解できること
   - 視覚でもスクリーンリーダーでも認識できる
   - そのUIオブジェクトは何なのか
   - 開／閉、選択／非選択の状態
   - コントローラ（ラベル）とパネル（コンテンツ）の関係

2. キーボードで操作できること
   - フォーカスインジケータが見える
   - 正しいフォーカス順序
   - キーボードトラップされない
   - onfocusやhoverでは実行しない

サンプルコードの状態（まぁまぁアクセシブル）

1. 知覚／理解できる
   - 見える（テキストと背景色のコントラストは4.5：1以上）
   - そのUIオブジェクトは何なのかは分からない
   - 開／閉、選択／非選択状態は視覚でのみ伝わる状態
   - コントローラ（ラベル）とパネル（コンテンツ）の関係も不明

2. キーボードで操作できる
   - フォーカスインジケータが見える
   - 正しいフォーカス順序（tabindex属性付加済）
   - キーボードトラップされない
   - onfocusやhoverでは実行しない

ついでに

3. より仕様に則した実装に変更
   - 独自のclassを使ったロジック

     → role属性／aria-〇〇〇属性を使用したロジック

   - 独自のclassを使ったスタイルの定義

     → CSSの属性セレクタを使用したスタイルの定義

WCAG 2.0 達成方法集（2017年4月20日更新！）

W3C Working Group Note

WCAG 2.0 達成方法集

WCAG 2.0の達成方法と不適合事例

W3C Working Group Note 2016年 10月 7日

このバージョン:
http://waic.jp/docs/WCAG-TECHS/

最新バージョン:
https://www.w3.org/TR/WCAG20-TECHS/

前のバージョン:
https://www.w3.org/TR/2016/NOTE-WCAG20-TECHS-20160317/

編集者:
Michael Cooper, W3C
Andrew Kirkpatrick, Adobe Systems Inc.
Joshue O Connor, InterAccess

過去の編集者:
Loretta Guarino Reid (until May 2013 while at Google, Inc.)
Gregg Vanderheiden (until May 2013 while at Trace R&D Center, University of Wisconsin-Madison)
Ben Caldwell (until September 2010 while at Trace R&D Center, University of Wisconsin-Madison)
Wendy Chisholm (until July 2006 while at W3C)
John Slatin (until June 2006 while at Accessibility Institute, University of Texas at Austin)

この文書は、以下の規定ではないフォーマットでも提供されている:

http://waic.github.io/wcag20/Techniques/Overview.html

WCAG 2.0 達成方法集（2017年4月20日更新！）

W3C Working Group Note

WAI-ARIA Technology Notes

ARIA1: ユーザインターフェース コントロールに対する説明ラベルを提供するために、aria-describedby プロパティを使用する

ARIA2: aria-required プロパティによって必須項目を特定する

ARIA4: ユーザインターフェース コンポーネントの役割（role）を明らかにするために、WAI-ARIA ロールを使用する

ARIA5: ユーザインターフェース コンポーネントの状態（state）を明らかにするために、WAI-ARIA ステート及びプロパティ属性を使用する

ARIA6: オブジェクトのラベルを提供するために aria-label を使用する

ARIA7: リンクの目的を示すために aria-labelledby を使用する

ARIA8: リンクの目的を示すために aria-label を使用する

ARIA9: 複数の語句をつなげて一つのラベルにするために、aria-labelledby を使用する

ARIA10: 非テキストコンテンツに対して代替テキストを提供するために、aria-labelledby を使用する

ARIA11: ページの領域を特定するために ARIA ランドマークを使用する

ARIA12: 見出しを特定するために role=heading を使用する

ARIA13: 領域とランドマークに名前（name）を付けるために、aria-labelledby を使用する

http://waic.github.io/wcag20/Techniques/aria.html

W3Cオススメの使い方

W3C Working Draft

TABLE OF CONTENTS

1. **Introduction**

2. **Design Patterns and Widgets**

2.1 Generally Applicable Keyboard Recommendations

2.2 Accordion (Sections With Show/Hide Functionality)

2.3 Alert

2.4 Alert and Message Dialogs

2.5 Breadcrumb

2.6 Button

2.7 Checkbox

2.8 Combo Box

2.9 Dialog (Modal)

2.10 Dialog (Non-Modal)

2.11 Grids : Interactive Tabular Data and Layout Containers

2.12 Link

2.13 Listbox

2.14 Menu or Menu bar

2.15 Menu Button

2.16 Radio Group

2.17 Slider

2.18 Slider (Multi-Thumb)

2.19 Spinbutton

2.20 Table

2.21 Tabs

2.22 Toolbar

2.23 Tooltip Widget

2.24 Tree View

WAI-ARIA Authoring Practices 1.1

W3C Working Draft 14 December 2016

This version:
https://www.w3.org/TR/2016/WD-wai-aria-practices-1.1-20161214/

Latest published version:
https://www.w3.org/TR/wai-aria-practices-1.1/

Latest editor's draft:
http://w3c.github.io/aria-practices/

Previous version:
https://www.w3.org/TR/2016/WD-wai-aria-practices-1.1-20160317/

Editors:
Matt King, Facebook, mck@fb.com
James Nurthen, Oracle Corporation, james.nurthen@oracle.com
Michiel Bijl, The Paciello Group, mbijl@pacielogroup.com
Michael Cooper, W3C, cooper@w3.org
Joseph Scheuhammer, Inclusive Design Research Centre, OCAD University (Previous Editor)
Lisa Pappas, SAS (Previous Editor)
Rich Schwerdtfeger, IBM Corporation (Previous Editor)

Copyright © 2015-2016 W3C® (MIT, ERCIM, Keio, Beihang). W3C liability, trademark and document use rules apply.

Abstract

This document provides readers with an understanding of how to use *WAI-ARIA 1.1* [WAI-ARIA] to create accessible rich internet applications. It describes considerations that might not be evident to most authors from the WAI-ARIA specification alone and recommends approaches to make widgets, navigation, and behaviors accessible using WAI-ARIA roles, states, and properties. This document is directed primarily to Web application developers, but the guidance is also useful for user agent and assistive technology developers.

This document is part of the WAI-ARIA suite described in the WAI-ARIA Overview.

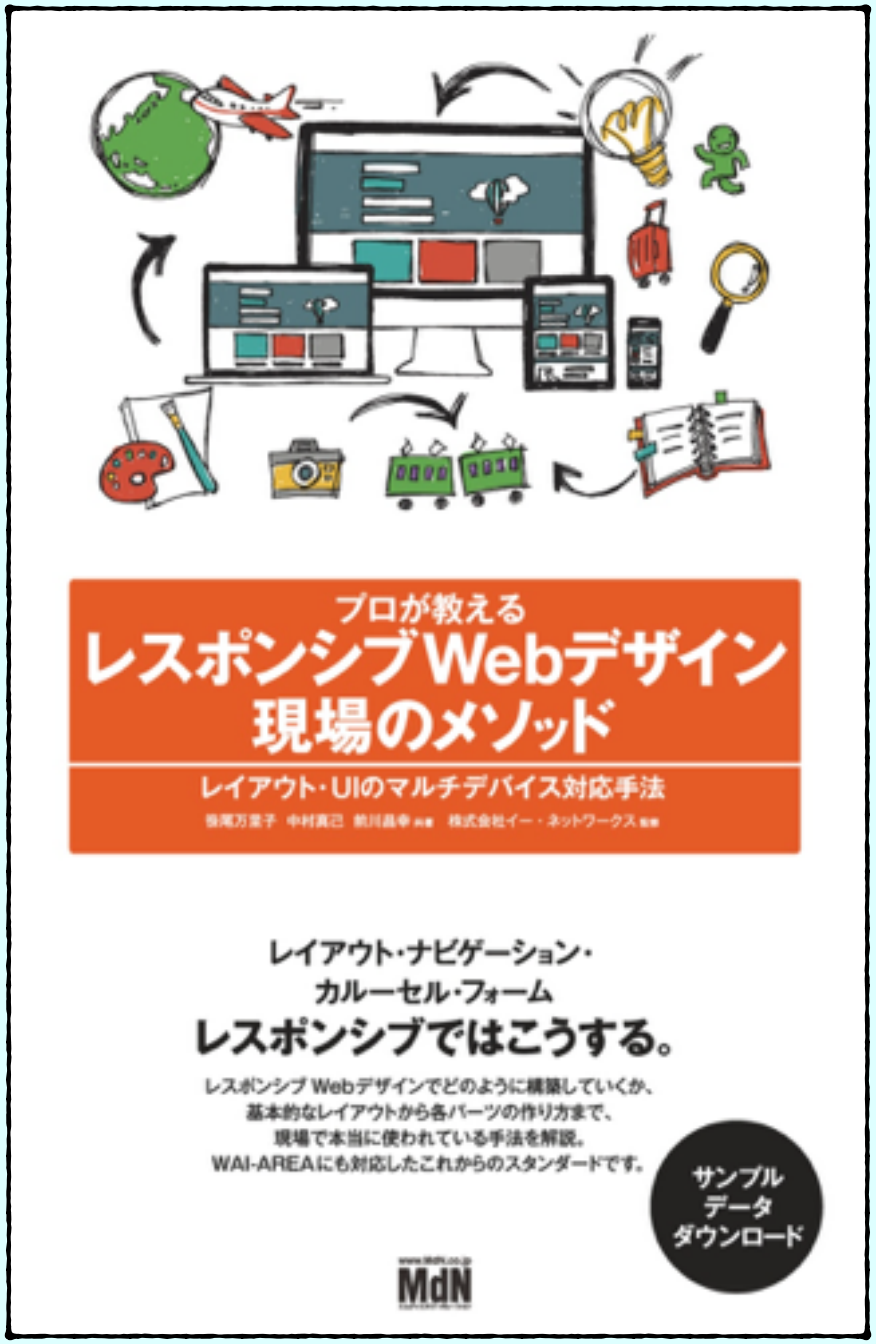
Status of This Document

https://www.w3.org/TR/wai-aria-practices-1.1/

サンプルコードは、

参考にして

この本を~~コピって~~作りました。



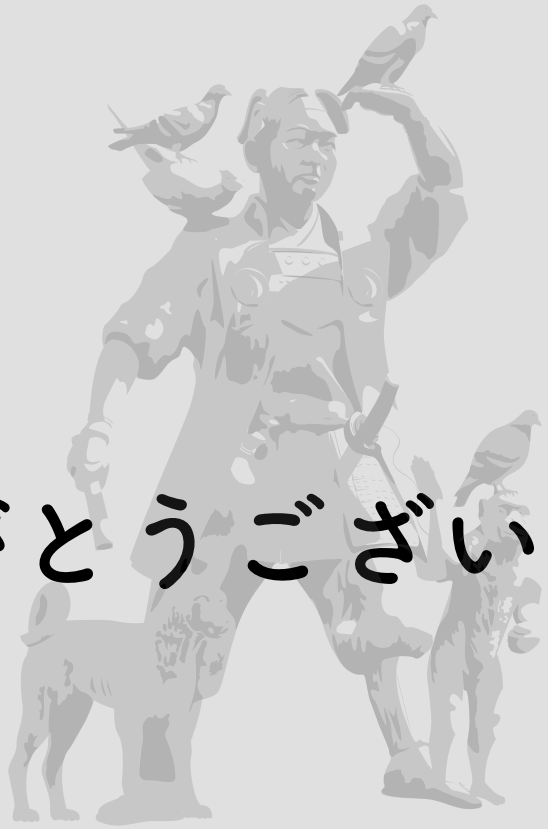
http://amzn.asia/3XvRLYO

サンプルコード置き場

- アコーディオンメニュー：bit.ly/2rEAWUf
- タブナビゲーション　　：bit.ly/2rEAZ2n
- このスライド　　：bit.ly/2rTUGSi

出来上がったら

スクリーンリーダーで聞いてみよう！



ありがとうございました

@SawadaStdDesign

SAWADA STANDARD DESIGN