

Accessibility.jp 無料セミナー

第3回 WCAG 2.1 勉強会

2021年12月22日(水)

講師：飯塚 慎司

株式会社インフォ・クリエイツ

Accessibility.jp の活動の紹介

- ウェブページを企画・設計・制作・保守する方達のアクセシビリティ配慮を助け、共に考えるサイトです。

<https://accessibility.jp/>

- miCheckerの検証結果の見方が分からず苦勞している方向けに、miCheckerテクニック集を提供しています。

<https://accessibility.jp/resources/tools/michecker-techniques/>

- ちょっとした情報はFacebookから提供しています。

<https://www.facebook.com/accessibility.jp2>

- オンラインセミナーにて情報の提供をしています。

2021/11/24 miChecker勉強会

2021/12/08 WCAG 2.1 勉強会 (前半)

2021/12/22 WCAG 2.1 勉強会 (後半)

目次

- WCAG 2.1 について
- 1.3.4 表示の向き (AA)
- 1.3.5 入力目的の特定 (AA)
- 1.4.10 リフロー (AA)
- 1.4.11 非テキストのコントラスト (AA)
- 1.4.12 テキストの間隔 (AA)
- 1.4.13 ホバー又はフォーカスで表示されるコンテンツ (AA)
- 2.1.4 文字キーのショートカット (A)
- 2.5.1 ポインタのジェスチャ (A)
- 2.5.2 ポインタのキャンセル (A)
- 2.5.3 名前 (name) のラベル (A)
- 2.5.4 動きによる起動 (A)
- 4.1.3 ステータスメッセージ (AA)
- WCAG 2.1で追加された達成基準
- WCAG 2.1早見表
- WCAG 2.1の状況
- W3C First Public Working Draft 21 January 2021
- 1.3.6 目的の特定 (AAA)
- 2.2.6 タイムアウト (AAA)
- 2.3.3 インタラクションによるアニメーション (AAA)
- 2.5.5 ターゲットのサイズ (AAA)
- 2.5.6 入力メカニズム非依存 (AAA)

WCAG 2.1 について

規格

W3C Recommendation – 2018/6/5

<https://www.w3.org/TR/WCAG21/>

翻訳版 – 2018/6/5

<https://waic.jp/docs/WCAG21/>

解説書

Understanding WCAG 2.1 – 2020/12/2 (現在 2021/7/27)

<https://www.w3.org/WAI/WCAG21/Understanding/>

翻訳版 WCAG 2.1 解説書 – 2019/3/6 (少し古いことに注意)

<https://waic.jp/docs/WCAG21/Understanding/>

What's New in WCAG 2.1

<https://www.w3.org/WAI/standards-guidelines/wcag/new-in-21/>

Think!

モバイルデバイスを、
車椅子に固定して利用する方、
ベッド上で主たる活動を行う方、
スクリーンリーダーを使用している方に対して、

やってはいけないことは？



1.3.4 表示の向き (AA)

コンテンツは、その表示及び操作を、縦向き (portrait) 又は横向き (landscape) などの単一の向きに制限しない。ただし、その表示の向きが必要不可欠な場合は例外とする。

意図

アプリケーションによっては画面の向きを縦あるいは横に強いるものがある。車椅子にデバイスを固定しているなどの理由で、デバイスの向きを自由に変えられない利用者や、視覚の問題で画面を特定方向に固定している人にとっては縦・横のどちらかに固定されるのは問題である。



Webではあまり問題にならないが、CSS3のメディアクエリなど使用して端末の向きを取得し利用しているケースでは特に要注意

1.3.4 表示の向き (AA)

対処

- 利用者が選択した向きに基づいて、映像を縦向き又は横向きに表示する。
- メッセージングウェブサイトは縦横の両方の向きでメッセージを表示できる。
- 電子書籍ウェブアプリは縦向きと横向きの両方で本の内容を表示できる。
- **向きが必要不可欠な例** 免許証の画像を簡単かつ正確に読み取るために、ランドスケープモードにする必要があるアプリがある。
- **向きが必要不可欠な例** ピアノの鍵を十分に機能的に使用できるスペースを確保するために、ランドスケープモードが求められるピアノのアプリ。

補足

- 航空機の座席予約では縦向きが一般的だが、デバイスを横向きにして横幅いっぱいに表示させたい利用者もいるかもしれない。

地図ページ上に追加で最寄りルートを表示する例



縦向きへ

アプリケーションによっては、横置きを前提としたレイアウトで自動的に表示する場合がある。

不適合例

縦置きでもレイアウトは変わらない。縦向きの表示では使いづらくなってしまう。



適合例

設定により、縦置きのままだでも利用が容易であるようにレイアウトが変更される。



Think!

あるオンラインショップの配送先の入力フォームを開くと
名字欄なのにフルネームが勝手に入力されてしまった
市外局番なのに10桁の番号がすべて入力されてしまった

電話番号 - -

自動で入力するならば、ちゃんと入れて欲しい

電話番号 - -

こんなことを実現するにはどうしたら良いのでしょうか

1.3.5 入力目的の特定 (AA)

利用者の情報を集める入力フィールドのそれぞれの目的は、次の場合にプログラムによる解釈が可能である:

- 入力フィールドが、ユーザインタフェース コンポーネントの入力目的の節で示される目的を提供している、かつ
- フォーム入力データとして想定される意味の特定をサポートする技術を用いて、コンテンツが実装されている。

意図

入力フィールドにおいて、意図を伝えるものとしては `type="mail"`, `type="tel"` などいくつかあるが、`autocomplete`属性を用いれば、もっと明確にマシンに対して意図を伝えることができるようになる。これより、言語や記憶に障害のある人は入力が必要な情報の記憶の負担を軽減できる。脳の障害によっては、アイコンなどで示されることで理解が容易になる。運動障害がある人は手入力の負担を減らすことができる。

自動入力を使用したコンタクトフォーム - 期待される項目 - 名前、郵便番号、市町村、番地、電話番号、電子メールアドレスなどの自動入力。

別々の請求先住所と配送先住所からの注文 - 目的の異なる住所でも使い分けが可能。

アイコンを使ったコンタクトフォーム - 入力目的を視覚的に示す名前、自宅住所、電話番号などのアイコンを挿入。

1.3.5 入力目的の特定 (AA)

対処

- HTML 5.2 autocomplete 属性を使用する。
HTML 5.2 Autofill field のセクション -
<https://www.w3.org/TR/html52/sec-forms.html#sec-autofill>
- ラベルが示すフォームフィールドの目的を、入力フィールドに設定するオートコンプリートトークンと一致させる。

補足

- 無効にするには autocomplete="off"
- section- に続くのはグループ名、shipping billingの併用も。

ユーザインタフェース コンポーネントの入力目的の一覧

- **name** - フルネーム
- **honorific-prefix** - 敬称又は称号 (例えば、"Mr."、"Ms."、"Dr."、"Mlle")
- **given-name** - Given name (一部の西洋文化において、first name として知られる)
- **additional-name** - Additional names (一部の西洋文化において、middle names として知られる、姓の前にあるファーストネームではない名前)
- **family-name** - Family name (一部の西洋文化において、last name 又は surname として知られる)
- **honorific-suffix** - 接尾語 (例えば、"Jr."、"B.Sc."、"MBASW"、"II")
- **nickname** - ニックネーム、スクリーンネーム、ハンドル: フルネームの代わりに使用する一般的に短い名前
- **organization-title** - 職種 (例えば"Software Engineer"、"Senior Vice President"、"Deputy Managing Director")
- **username** - ユーザ名
- **new-password** - 新しいパスワード (例えば、アカウントの作成又はパスワードの変更の場合)
- **current-password** - username フィールドによって識別されるアカウントに対する現在のパスワード (例えば、ログイン時)
- **organization** - 人、アドレス、又はこのフィールドに関連付けられた他のフィールドの連絡先情報に対応する会社名
- **street-address** - 所在地住所 (複数行、改行を保持)

ユーザインタフェース コンポーネントの入力目的の一覧

- **address-line1** - 所在地住所 (フィールドごとに 1 行、1 行目)
- **address-line2** - 所在地住所 (フィールドごとに 1 行、2 行目)
- **address-line3** - 所在地住所 (フィールドごとに 1 行、3 行目)
- **address-level4** - 四つの行政レベルでのアドレスの中で、最もきめ細かい行政レベル。
- **address-level3** - 三つ以上の行政レベルの住所で、第 3 行政レベル
- **address-level2** - 二つ以上の管理レベルの住所で、第 2 管理レベル。二つの行政レベルをもつ国で、これは通常、関連する所在地住所が見つけれられる、市、町、村、又は他の地方になる
- **address-level1** - 住所で最も広範な行政レベル、すなわち位置関係が発見される中の地方。たとえば、アメリカで、これは州 (state) になる。スイスで州 (canton) になる。英国で、郵便町 (post town) になる。
- **country** - 国コード
- **country-name** - 国名
- **postal-code** - 郵便番号、ZIP コード、CEDEX コード (CEDEX の場合、"CEDEX"を、関連する場合郡を、address-level2 フィールドに追加する)
- **cc-name** - 決済手段に与えられるようなフルネーム
- **cc-given-name** - 決済手段に与えられるような Given name (一部の西洋文化において、first name として知られる)
- **cc-additional-name** - 決済手段に与えられるような Additional names (一部の西洋文化において、middle names として知られる)

ユーザインタフェース コンポーネントの入力目的の一覧

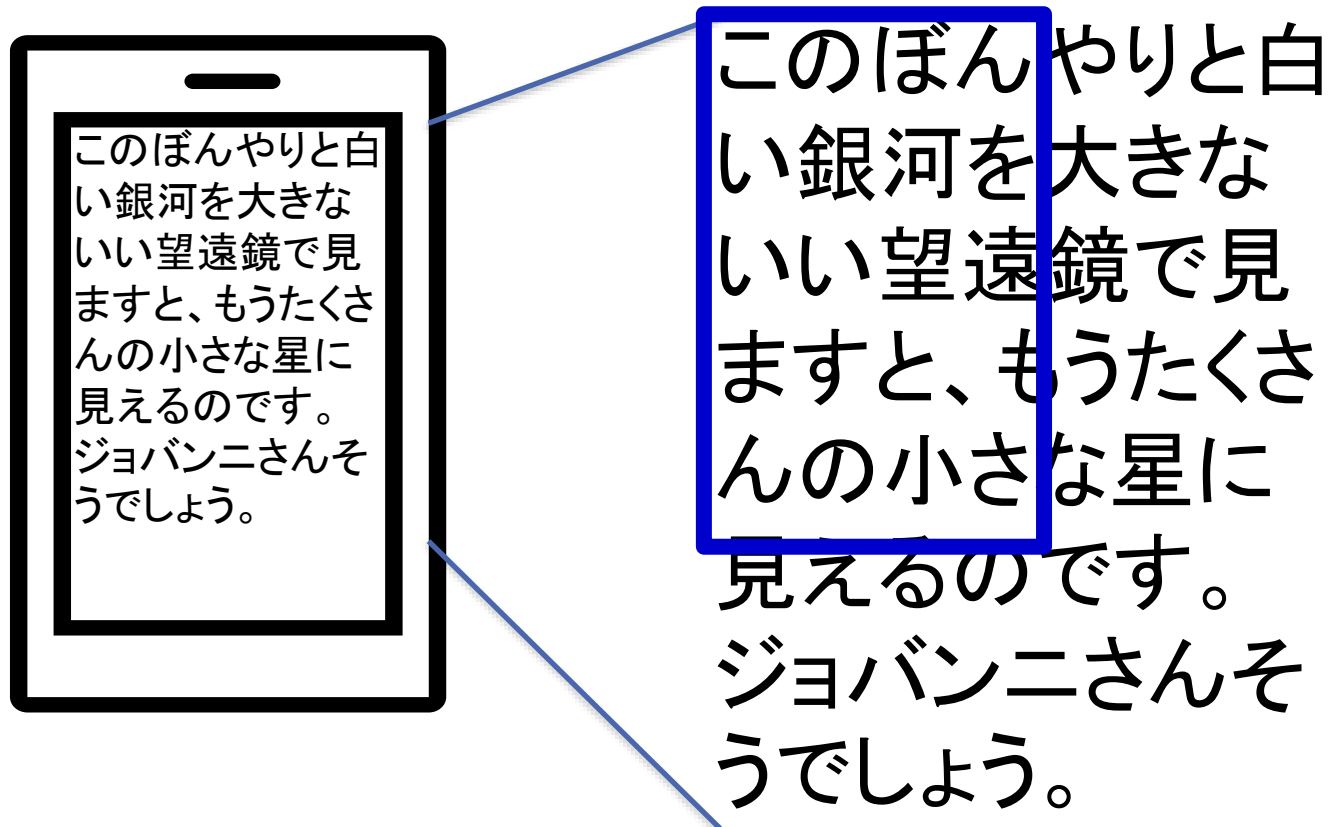
- **cc-family-name** - 決済手段に与えられるような Family name (一部の西洋文化において、last name 又は surname として知られる)
- **cc-number** - 決済手段 (例えばクレジットカード番号) を識別するコード
- **cc-exp** - 決済手段の有効期限
- **cc-exp-month** - 決済手段の有効期限の月コンポーネント
- **cc-exp-year** - 決済手段の有効期限の年コンポーネント
- **cc-csc** - 決済手段のセキュリティーコード (カードのセキュリティーコード (CSC)、カード検証コード (CVC)、カード検証値 (CVV)、署名欄コード (SPC)、クレジットカード ID (CCID) などとして知られている)
- **cc-type** - 決済手段の種類
- **transaction-currency** - 利用者が取引での使用を望む通貨
- **transaction-amount** - (例えば、ビッド又は販売価格を入力する時に) 利用者が取引したい金額
- **language** - 優先言語
- **bday** - 誕生日
- **bday-day** - 誕生日の日のコンポーネント
- **bday-month** - 誕生日の月のコンポーネント
- **bday-year** - 誕生日の年のコンポーネント

ユーザインタフェース コンポーネントの入力目的の一覧

- **sex** - 性別認識 (例えば、女性、ファファフィネ)
- **url** - このフィールドに関連する他のフィールドに入力された会社、個人、住所、又は連絡先情報に対応するホームページ又はその他のウェブページ
- **photo** - 写真、アイコン、会社に対応する他の画像、人、アドレス、又はこのフィールドに関連付けられる他のフィールドの連絡先情報
- **tel** - 国コードを含む完全な電話番号
- **tel-country-code** - 電話番号の国コードのコンポーネント
- **tel-national** - 適用される場合は国の内部の接頭辞とともに、国コードのコンポーネントなしの電話番号
- **tel-area-code** - 適用される場合は国の内部の接頭辞とともに、電話番号のエリアコードのコンポーネント
- **tel-local** - 国コードと市外局番のコンポーネントなしの電話番号
- **tel-local-prefix** - コンポーネントが二つのコンポーネントに分割される場合の、市外局番に続く電話番号のコンポーネントの最初の部分
- **tel-local-suffix** - コンポーネントが二つのコンポーネントに分割される場合の、市外局番に続く電話番号のコンポーネントの 2 番目の部分
- **tel-extension** - 電話番号の内部拡張コード
- **email** - 電子メールアドレス
- **impp** - インスタントメッセージングプロトコルのエンドポイントを表す URL (例えば、"aim:goim?screenname=example"又はxmpp:fred@example.net")name - フルネーム

Think!

文字サイズを拡大するとき、
縦横方向に等しく拡大することは不適合になります。
なぜだか分かりますか？



1.4.10 リフロー (AA)

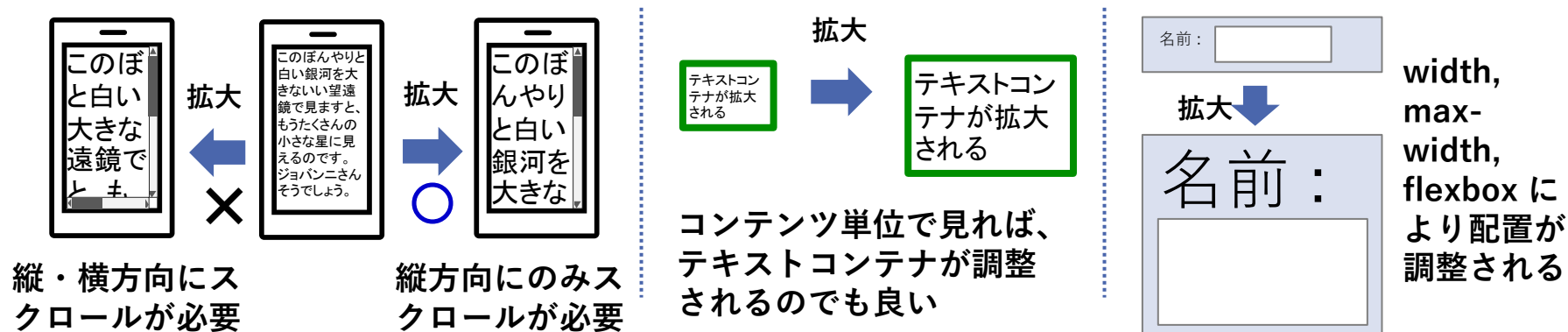
コンテンツは、情報又は機能を損なうことなく、かつ、以下において 2 次元スクロールを必要とせずに提示できる:

- 320 CSS ピクセルに相当する幅の縦スクロールのコンテンツ。
- 256 CSS ピクセルに相当する高さの横スクロールのコンテンツ。

1280ピクセルの幅を持つブラウザであれば、320ピクセルの幅を持つコンテンツの場合、400%に拡大するとリフローが発生する。(それ以下の場合には発生しない) そのため、320云々を行っている。

意図

テキストを拡大しリフローが発生するとき、縦方向と横方向の両方のスクロールが必要な状況になってしまうと、行末まで読んで、行頭に横スクロールして戻るときに、行を見失う可能性がある。これは、弱視でなくとも誰にとっても使いにくいものである。なお、横書きの言語ならば縦方向のみ、縦書きの言語などは横方向のみにする必要がある。



その他、横スクロールが表示される場合、それを不要にするオプションがあるので良い。

1.4.10 リフロー (AA)

対処

- 正しくレスポンシブデザインにする。

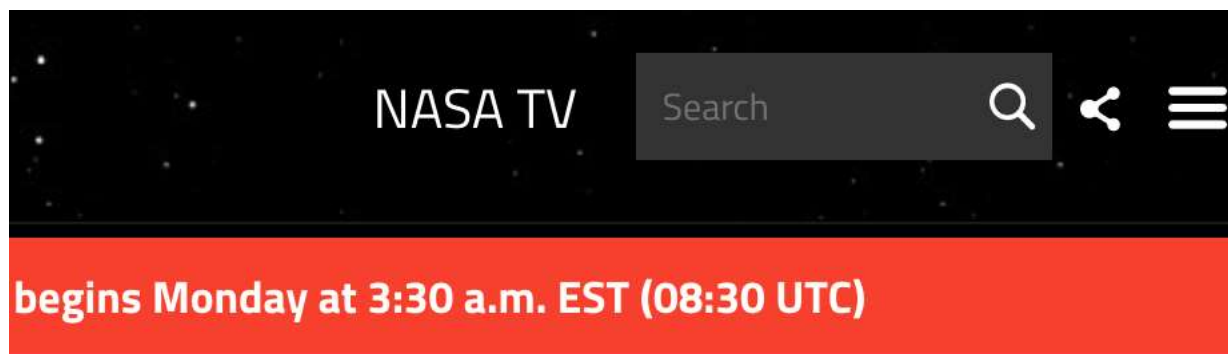
補足

- 横書きの場合、横方向のスクロールはさせるべきではない。(縦書きならば縦方向は好ましくない)
- レスポンシブデザインの場合、拡大していくと文字サイズが変更になる場合がある。そのような場合でも 200%の大きさで表示されるポイントは用意しておくこと。
- 地図、動画、表などは例外である。
- CSSに固定サイズを用いない。拡大・縮小を禁止しない。また、`position: absolute;` でコンテンツを固定しない。

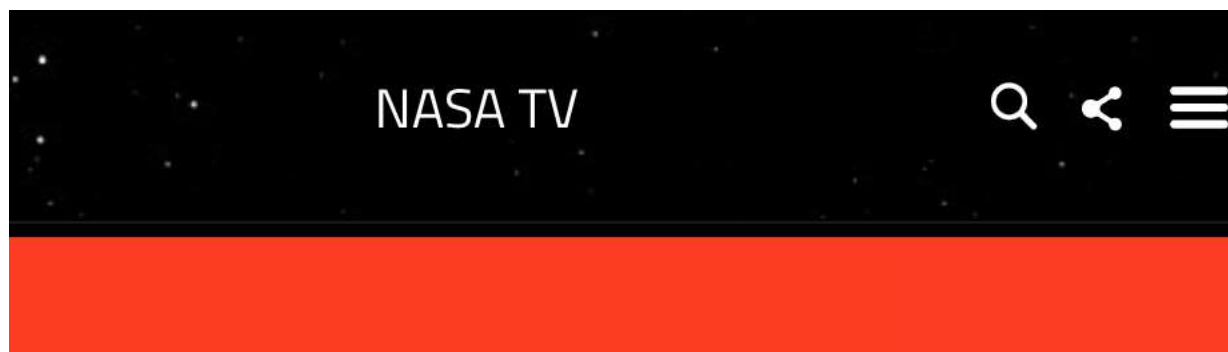
Think!

コントラストに問題がある箇所が3箇所あります

何処か分かりますか？



このように見えているようなもの・・・



1.4.11 非テキストのコントラスト (AA) (1/2)

以下の視覚的提示には、隣接した色との間で少なくとも 3:1 のコントラスト比がある。

ユーザインタフェース コンポーネントユーザインタフェース コンポーネント及び状態 (state) を特定するのに必要な視覚的な情報。ただし、アクティブではないユーザインタフェース コンポーネントや、そのコンポーネントの見た目がユーザエージェントによって提示されていてコンテンツ制作者が変更していない場合は除く。

グラフィカルオブジェクト コンテンツを理解するのに必要なグラフィック部分。ただし、そのグラフィック特有の提示が、情報を伝えるうえで必要不可欠な場合は除く。

意図

見ることに何らかの困難がある方には、低いコントラストのイメージは制作者の意図通りに見えていない可能性がある。

テキストについては1.4.3 コントラスト（最低限）で基準が設けられているが、1.4.11ではコントロールや意味を持つ画像に適用される。

この達成基準で配慮しなくてはならないケースはとても多い。

対処方法

基本的な考え方

- アクティブなユーザーインターフェース
コンポーネントとは

未読と既読あるいはボタンのon/offなど、状態の変化の前後でのコントラスト比を設けることは求められていない。

[リンク前](#) [リンク済み](#)

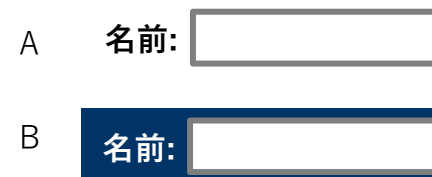
- 境界とは

ボタンには境界があるが、中に十分なコントラスト比を持つテキストやアイコンの手掛かりがあれば、この基準は境界には適用されない。ただし認知障害の理解を支援する効果はあるので、境界を明確にすることは推奨される。



- 隣接している色とは

Aはボーダー(銀)と背景(白)が隣接している色であり、Bは内側の色(白)と背景(紺)が隣接している色と考える。ボーダーは基準の判定には重要ではない。



対処方法

基本的な考え方

- 色の使用とフォーカスの可視化との関係性とは
色の使用の例は次のとおり。



気になる記述 <https://www.w3.org/WAI/WCAG21/Understanding/non-text-contrast.html>

In combination with [2.4.7 Focus Visible](#), the visual focus indicator for a component *must* have sufficient contrast against the adjacent background when the component is focused, except where the appearance of the component is determined by the user agent and not modified by the author.

Note that this Success Criterion does not directly compare the focused and unfocused states of a control - if the focus state relies on a change of color (e.g., changing *only* the background color of a button), this Success Criterion does not define any requirement for the difference in contrast between the two states.

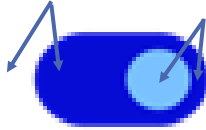
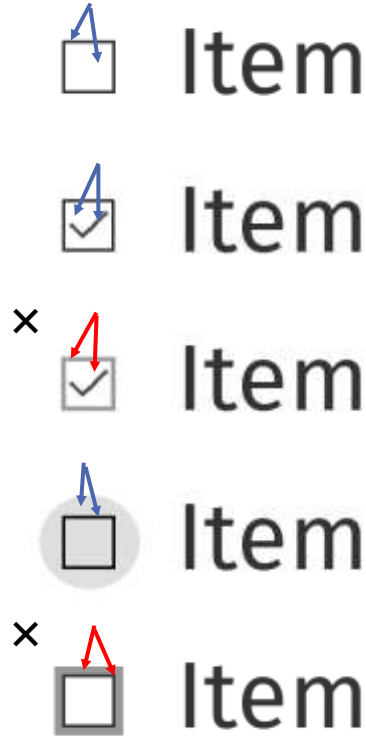
[2.4.7 フォーカスの可視化](#)と組み合わせて、コンポーネントがフォーカスされている時に、そのコンポーネントの視覚的なフォーカスインジケータは、隣接している背景に対して十分なコントラストがなければならない。ただし、コンポーネントの見た目がユーザエージェントにより決定されており、かつコンテンツ制作者により変更されていない場合を除く。もしフォーカス状態が色の変化（例: ボタンの背景色のみを変更する）に依存している場合、コントロールの前の状態から最低でも 3:1 のコントラスト比があるように一つの色から他のものに変更することはフォーカスの可視化の基準を満たす一つの方法である。

この訳は古いかも。大切なことは、フォーカスの前後でのコントラスト比は求めていることだと思います。

アクティブなユーザインタフェース コンポーネントの例1

タイプ	説明	事例
リンクテキスト	リンクテキストは 1.4.3 コントラスト (最低限) のスコープに含まれ、かつ下線はリンクを示すのに十分。	 An example of link text, "Link text", underlined in blue.
既定のフォーカススタイル	リンクは 2.4.7 フォーカスの可視化 でフォーカスインジケータを持つ。	 An example of link text, "Link text", with a dashed blue border indicating focus.
ボタン	位置、テキストスタイル、コンテキストなどの識別インジケータを持つボタンには、それがボタンであることを示す対照的な視覚インジケータは必要ない。(推奨はされる)	 An example of a button, "Button", with a solid blue background.
テキスト入力 (最小限)	テキスト入力に、下部のボーダーなど、入力であるという視覚的なインジケータがあるところでは、そのインジケータは 3:1 のコントラスト比を満たさなければならない。	 An example of a text input field with a light gray background and a thin bottom border. The label "Name:" is to the left. Arrows point to the bottom border.
テキスト入力	完全なボーダーなど、テキスト入力がインジケータを持つところでは、そのインジケータは 3:1 のコントラスト比を満たさなければならない。	 An example of a text input field with a white background and a thin black border. The label "Name:" is to the left. Arrows point to the border.
テキスト入力のフォーカススタイル	フォーカスインジケータは必須。もしフォーカスインジケータがコンテンツ制作者によりスタイル付けされている場合は、3:1 のコントラスト比を満たさなければならない。	 An example of a text input field with a white background and a thick blue border indicating focus. The label "Name:" is to the left. Arrows point to the border.
背景色を使用しているテキスト入力	ボーダーがなく、かつ背景色のみで区別されるテキスト入力は隣接している背景に対して 3:1 のコントラスト比を持たなければならない。	 An example of a text input field with a dark blue background and a white border. The label "Name:" is to the left. Arrows point to the background.

アクティブなユーザインタフェース コンポーネントの例2

タイプ	説明	事例
トグルボタン	トグルボタンの内部背景(#070CD5)は、外部の白背景とも、内部の丸いトグル(#7AC2FF)ともコントラスト比を満たす。	
ドロップダウンインジケータ	下向きの三角がドロップダウンであることを示す。いずれのケースもコントラスト比を満たす。	
チェックボックス	• チェックボックスの黒枠と内部の白はコントラスト比を満たし空であることがわかる • 加えて、チェック印もコントラスト比を満たしチェック済みであるとわかる • チェックボックスの枠はグレー(#9D9D9D)でありコントラスト比は2.7:1で要件を 満たさない。 • ホバー状態を示す背景はボーダーとのコントラスト比を満たすことが求められる。淡いグレー(#DEDEDE)はボーダーと15:1のコントラスト比があり規定を満たす。 • フォーカス状態を示す背景はボーダーとのコントラスト比を満たすことが求められる。グレー(#AAA)はボーダー(#FFF)とのコントラスト比が2.3:1しかなく規定を 満たさない。	

グラフィカルオブジェクト

ポイントは、オブジェクトのすべての箇所がコントラスト比の対象になるわけではなく、利用者の理解に必要な箇所のみである。判断には習熟が必要であると推測される。

一般的な例

- 電話のマークと背景のコントラスト比が 3:1以上が必要



- 磁石であることがわかる必要がある。それには、先端の白の部分と他の赤の部分と比べてわかることが必要

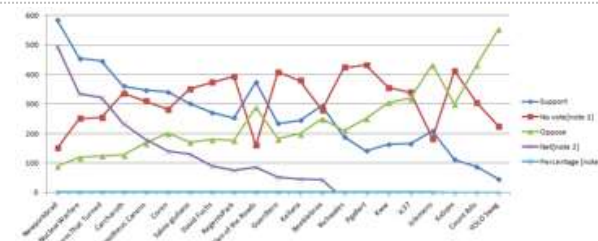


コントラスト比が満たされていないと、右のように見えるかもしれない。
これでは磁石と分らない。

- 通貨安を示す記号 このグラフィックを理解するには、白い背景に対する下向き矢印、及び黄色背景に対するポンドアイコンのコントラストが必要



- 各線及びマークが白い背景に対して 3:1 以上ある必要がある。緑と水色は背景に対しコントラスト比を満たさない可能性がある。線同士のコントラスト比はこの例では求められないが、もし、重なりや隣接の影響で、認識することに問題を与える場合は問題となる。



- 円グラフにおいては、一つ一つの扇の部分と他と区別できることがポイントとなる。



円グラフの各要素同士のコントラスト比が不足しているので不適合



円グラフが見えないとしても、数値が提供されているので不適合にはならず非適用（問題なし）



隙間によってコントラスト比が保たれるので適合

グラフィカルオブジェクト

理解するのに必要とは

- ラベル及びチャート内の値など、グラフィックに埋め込まれた又は重ねられたテキストが同じ情報を伝えている
- グラフィックが見栄えのためにあり、利用者にコンテンツを理解するため又は機能を使用するために見る又は理解することを求めない。
- 「長い説明」 ボタンを押下すると表示されるグラフに続く表など、情報が他の形式で提供されている。
- グラフィックはロゴ又はブランド名の一部である（その提示に「必要不可欠」と考えられている）。

グラデーション

- コントラスト比3:1を満たさない箇所は背景と同じ色であると理解して、情報の欠落を確認する。

動的な例

- マウスを当てると追加情報を提供するコンポーネントがあったとする。
この場合、この達成基準の対象になる。



例外

- **ロゴタイプ及び旗**:組織又は製品のブランドのロゴはその組織の象徴であるため、免除する。旗は、十分なコントラストを持つために色が変更された場合、識別できないかもしれない。
- **感覚的**:人又は風景の写真などの現実の場面の絵を変更するという必須要件はない。
- **他のものを表している**:グラフィックを他のどの方法でも表現できない場合、それは必要不可欠である。
例に含まれるのは:
 - ウェブサイトがどのように表示されていたかを明示するスクリーンショット。
 - 生物学で見られる色を使用した医療情報の図 (Wikipediaの医療的な図解の例)。
 - ヒートマップなどの、測定結果を表す色のグラデーション (Wikipediaからのヒートマップの例)。

インフォ・グラフィック

インフォ・グラフィックの例



円の大きさがネットワークの規模を示している。
このケースでは大きさは見て取れない可能性がある



3:1以上のボーダーを追加して、コントラスト比を確保している。

Think!

次の設定をしたとき、どんな問題が発生するか考えてみましょう。

- 行の間隔（行送り）をフォントサイズの1.5 倍に設定
- 段落に続く間隔をフォントサイズの 2 倍に設定
- 文字の間隔（字送り）をフォントサイズの 0.12 倍に設定
- 単語の間隔をフォントサイズの 0.16 倍に設定する

1.4.12 テキストの間隔 (AA)

以下のテキストスタイルプロパティをサポートするマークアップ言語を用いて実装されているコンテンツにおいては、以下をすべて設定し、かつ他のスタイルプロパティを変更しないことによって、コンテンツ又は機能の損失が生じない:

- 行の間隔 (行送り) をフォントサイズの少なくとも 1.5 倍に設定する
- 段落に続く間隔をフォントサイズの少なくとも 2 倍に設定する
- 文字の間隔 (字送り) をフォントサイズの少なくとも 0.12 倍に設定する
- 単語の間隔をフォントサイズの少なくとも 0.16 倍に設定する

意図

テキストのレイアウトは読みに困難のある利用者にとってはとても重要である。例えば、障害によっては、行が詰まっていると、読み進めるうちに行を間違えてしまうことがある。あるいは、単語と単語の間に十分なスペースがないと、単語の単位を見誤ることがある。文字の間隔も重要であり、あまり詰まっていると単語そのものが認識しにくくなることもある。そのような利用者は行間を広げるなどのツールを利用していることがあるが、そうしたツールが有効に働くようにすることがこの達成基準の一つの目的である。

1.4.12 テキストの間隔 (AA)

対処

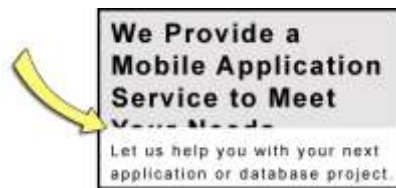
- 利用者が達成基準のように設定することを妨げない。かつ、失敗事例のようなことが起きないようにする。
(テキストはそれを包括するボックスに収まり、他のボックスとも重ならないようにする。)

- 検査には次のブックマークレットを使用する。

<http://html5accessibility.com/tests/tsbookmarklet.html>

補足

失敗事例



切り取られるテキスト

テキストの一部が切り取られてしまっている。



切り取られるテキスト

テキストのはみ出た部分が失われている



テキストの重なり

テキストの重なりが生じ、読むことが困難になっている。

Think!

マウスをホバーさせると表示されるツールチップ
結構、大きいのが出てくることがあります
あれが邪魔だと思うことはありませんか？
拡大し操作している人には、特に邪魔になります。

例えば、

- ・ 下に表示されているものが見えない
- ・ 良く見えず拡大しようとマウスをその上に移動するとツールチップが消えてしまう・・・

では、どうなれば良いのでしょうか？



1.4.13 ホバー又はフォーカスで表示されるコンテンツ (AA)

ポインタホバー又はキーボードフォーカスを受け取ってから外すことで、追加コンテンツを表示させてから非表示にさせる場合は、以下の要件を満たす:

非表示にすることができる

ポインタホバーやキーボードフォーカスを動かさずに追加コンテンツを非表示にする メカニズム が存在する。ただし、追加コンテンツが 入力エラー を伝える場合や、他のコンテンツを不明瞭にしたり置き換えたりしない場合は除く。

ホバーすることができる

ポインタホバーによって追加コンテンツを表示させることができる場合、その追加コンテンツを消すことなく、ポインタを追加コンテンツ上で動かせる。

表示が継続される ホバーやフォーカスが解除される、利用者が非表示にする、又はその情報が有効でなくなるまでは、追加コンテンツが表示され続ける。

例外: 追加コンテンツの視覚的提示がユーザエージェントによって制御されていて、かつコンテンツ制作者が変更していない場合は例外とする。

意図

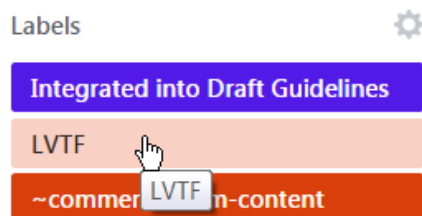
マウスを当てた時、キーボードでフォーカスを与えた時、ポップアップなど追加コンテンツを表示するケースでは、

- 利用者はインタラクションを起こそうとは思っていない可能性がある
- 利用者は新しいコンテンツが現れたことに気がつかない可能性がある
- 新しいコンテンツが利用者のタスクを邪魔する可能性がある

1.4.13 ホバー又はフォーカスで表示されるコンテンツ (AA)

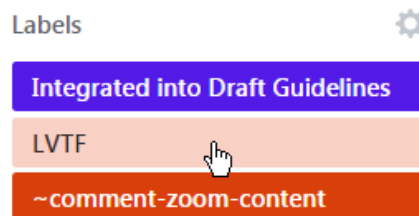
対処

- **非表示にすることができる** エラーメッセージなど追加の表示が明らかな空白エリアに表示されている場合を除いて、Escキーを押したり、閉じるボタンを押すと非表示にすることができるようにする。



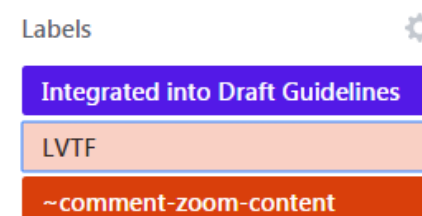
不適合例

ツールチップがコメントを隠してしまっている



対処例

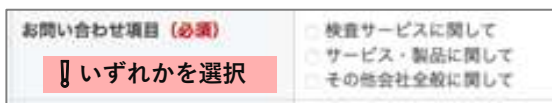
ESCキーを押すとツールチップが消えるようになっている



キーボード操作の例

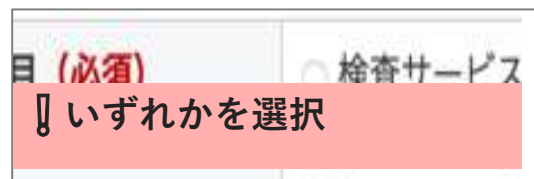
キー操作時でもESCキーを押して消すことができる

拡大していない通常時



未入力箇所を示すメッセージがラベル部分に追加表示されている

200%に拡大

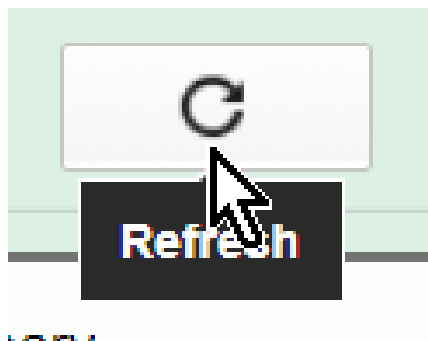


200%拡大時はメッセージがラジオボタンを隠してしまっている。
ESCキーで非表示にしたい

1.4.13 ホバー又はフォーカスで表示されるコンテンツ (AA)

- **ホバーすることができる** ポインタホバーにより表示されたコンテンツの上で、ポインタを動かすことができる。

不適合例



ポインタを拡大しているときに顕著。ポインタがポップアップに重なり、ポップアップの内容が読めないことがある。ポップアップを読もうとマウスを動かすとポップアップも消えてしまい永遠に読めない。

対処例



マウスポインタはボタンからは外れているが、ポップアップの上にいる限りは表示される。マウスを適当な位置に動かしポップアップを読むことができた。ESCキーで閉じることでもある。

- **表示が継続される** 意図してホバーやフォーカス状態を解除したり、利用者が非表示にしたり、ポップアップが無効なものにならない限り表示を継続する。

Example of a pop-over on hover and focus

<https://www.w3.org/WAI/WCAG21/working-examples/script-hoverable/>

Think!

音声認識ソフトでWebメールを読んでいます

「ご飯ができたよ～」と声がしたので
「いま、いきま～す」と答えたら・・・

キーボードの「e」を押したときのように動作してしまい、
読んでいたメールがアーカイブされてしまった！

「どうしよう!」と思わず呟いたら
「d」が認識されて次のメールは消えてしまった!

こんな事故を避けるには、どんなルールが必要でしょう

2.1.4 文字キーのショートカット (A)

文字 (大文字と小文字を含む)、句読点、数字、又は記号のみを使用したキーボードショートカットがコンテンツに実装されている場合、少なくとも次のいずれかを満たしている:

解除 ショートカットを解除するメカニズムが利用できる

再割り当て 一つ以上の非印字文字 (例えば Ctrl や Alt など) を使用するようにショートカットを再割り当てするメカニズムが利用できる

フォーカス中にのみ有効化 ユーザインタフェース コンポーネントのキーボードショートカットは、そのコンポーネントがフォーカスをもっているときのみ有効になる。

意図

アプリケーションのようなサイトが増えるとなれば、ショートカットキーは多用される恐れがある。例えば、Google Drive のページでは次のような問題がある。



"d"を押すとマイドライブが開く。もし、音声入ソフトを実行したままで、たまたま"dog"と発声すると、最初の"d"に反応してマイドライブが開いてしまう可能性がある。



2.1.4 文字キーのショートカット (A)

対処

ショートカットを利用する場合は、次のいずれかが必要。

- 解除することができる。
- Ctrl や Alt と組み合わせないと動作しないように再割り当てできる。
- そのコンポーネントにフォーカスが当たっているときだけ有効にする。

補足

- Gmailの対応の仕方が参考になる。
[設定] -[すべての設定を表示]



- "d"を"s"に変えるだけでは不足。
- フォーカスの当たった申し込みボタンがEnterだけでなく、"yes"のyキーで認識するのは問題ない。

Think!

次のうち、問題になるのはどれ？



ピンチ、ズーム



ロングプレス



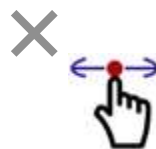
二本指あるいは
三本指でのタップ



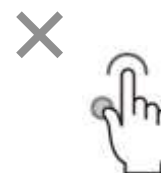
タップ



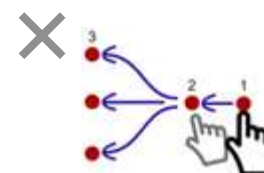
二本指あるいは
三本指でのスワイプ



右、左へのフリック



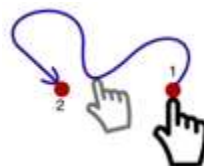
スプリットタップ
(一箇所を押さえない
がらのタップ)



どこかを經由して
ポインタを運んでいる



ダブルタップ



1から2にポインタを運ぶ

2.5.1 ポインタのジェスチャ (A)

マルチポイント又は軌跡ベースのジェスチャを使って操作する機能はすべて、軌跡ベースのジェスチャなしのシングルポインタで操作することができる。ただし、マルチポイント又は軌跡ベースのジェスチャが必要不可欠である場合は例外とする。

意図

利用者によってはジョイスティックを使ったり音声入力など多用なポインティングデバイスを利用することがある。そうした方法でも正確に操作できるようにするためには、マルチポイントや追跡ベースのジェスチャによる入力は、シンプルなシングルポイントの操作方法に置き換えられることを求めている。

2.5.1 ポインタのジェスチャ (A)

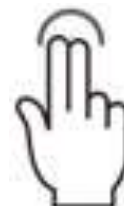
配慮が必要になるマルチポイントの例



ピンチ、ズーム



スプリットタップ
(一箇所を押さえないがらのタップ)

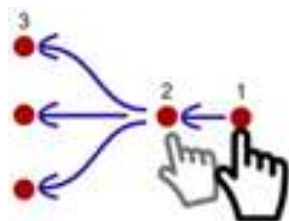


二本指あるいは
三本指でのタップ



二本指あるいは
三本指でのスワイプ

同様に配慮が必要な追跡ベースのジェスチャの例 (path-based gesture)

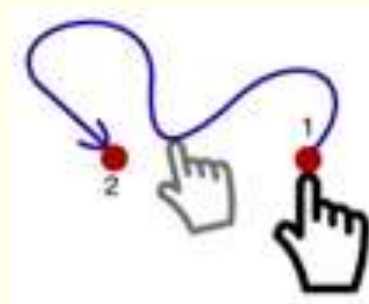


どこかを經由して
ポインタを運んでいる



右、左など方向に意
味がある

○ この例は、経路によらず2に到達することだけを情報としているので追跡ベースのジェスチャではありません。



WCAG 2.2 では、これも不適合になるはずです。

他にスワイプ、スライダー、ページめくりのような動作も該当します

2.5.1 ポインタのジェスチャ (A)

(続き)

対処

- ・シングルポイントでの操作や他の例のような配慮をして動作を保証する。

好ましいシングルポイントの例



タップ



ダブルタップ



ロングプレス

マウスの場合は、シングルクリック、クリック&ホールド、ダブルクリックが含まれる。

他の対応例

- [+] [-] ボタンによるズーム機能
- ▲▼ ボタンによる音量調整



スライダーの操作だけでなく、シングルポイントで上下ボタンでも操作できる [\(参考\)](#)

補足

- ・達成基準2.1.1 キーボードに関する要件を満たしたとしても、2.5.1は適用除外とはならないことに注意すること。

Think!

ボタンを押す、というシンプルな動作でも難しい場合があります

次のような困難を持つ場合、

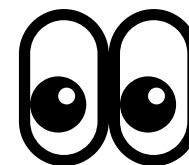
シングルポイントにはどんな困難があるでしょう。



運動
(巧緻性)



認知



視覚

もう一つ!

指で、押す動作と離す動作では、
どちらが安定しているでしょう。

2.5.2 ポインタのキャンセル (A)

シングルポインタを使って操作できる機能は、以下の要件の少なくとも 1 つを満たす。

ダウンイベントがない 機能を実行する目的でポインタのダウンイベントを使用していない。

中止又は元に戻すことができる 機能の完了にはアップイベントを使用し、かつ機能の完了前に中止する、又は機能の完了後に元に戻すためのメカニズムが利用できる。

アップイベントで反転 アップイベントによって、先のダウンイベントのすべての結果が反転する。

必要不可欠 ダウンイベントによって機能を完了させることが必要不可欠である。

意図

さまざまな障害によって誤ったポインタ入力をしてしまうことがある。そうした時にキャンセルできるようにすることを求めている。

誤入力の例



手に震えがあり意図しないボタンを押してしまった



認知に制約があり押した後
に間違いに気がついた



視覚に弱さがあり誤って
押してしまった

手に震えのある人などは、狙って押すよりも、選んでから離す方がスムーズに操作できるということが基本です。

2.5.2 ポインタのキャンセル (A)

対処

- アップイベントで動作させる、あるいは終了させる
 - 触れてしまっただけでトリガーとさせない
- アップイベントで中止または元に戻す
 - 誤って押してしまっても離すまでにキャンセルを選択することができる
 - ドラック & ドロップでは元に戻したり領域外でアップすることでキャンセルとする
- アップイベントで反転する
 - 押している間だけポップアップ表示するなど
 - アップイベントがキャンセルを意味する
- ダウンイベント
 - キーボードやピアノ、ゲームのような特別な状況
 - このような場合、別にキャンセルの仕組みが必要になる

Think!

音声認識ソフトを使っています

次のテキストフィールドを選択するために

名前：

出身地：

「名前を入力」と話すと、名前の欄に移動しました。
しかし「出身地を入力」と話しても、何も反応しません。
なぜでしょう。

label要素やtitle属性がない?! そんなことはありません。
ちゃんと、title属性はついています。

2.5.3 名前 (name) のラベル (A)

ユーザインタフェース コンポーネントがテキスト又は文字画像を含むラベルを持つ場合、視覚的に提示されたテキストが名前 (name) に含まれている。

意図

視覚的に見えているラベル（可視ラベル）を頼って操作している人が、これを音声入力などのプログラムで活用できるようにするための要求事項である。

可視ラベルが、プログラムが認識しているテキスト（Accessible Name）と異なるために、音声入力で操作しようと可視ラベルを読み上げてもうまく動作してはくれない。例えば次の例は、製品名のフィールドに音声入力で移動しようと「製品名に移動」と発声しても移動はしない。プログラムは「商品名」と認識しているからである。

製品名 :

製品名 : `<input type="text" name="product" title="商品名">`

発話した内容が画面に表示されているものではなく Accessible Name であったため思わぬ動作をすることもある。

可視テキストを含まないコンポーネントには適用されない。

2.5.3 名前 (name) のラベル (A)

対処

- 名前とラベルを一致させる。

製品名 : `<input type="text" name="product" title="製品名">`

あるいは

`<label>製品名 : <input type="text" name="product"></label>`

- 名前は可視ラベルのテキストを**最初**に含むようにする。

ラベル

名前

製品名 : `<input type="text" name="product" title="製品名の入力欄">`

ラベル (label) - テキスト、又はテキストによる代替を伴うコンポーネントで、ウェブコンテンツ内のコンポーネントを識別するために利用者に提示されているもの。

名前 (name) - ソフトウェアが、ウェブコンテンツのコンポーネントを利用者に識別させることができるテキスト。**(name属性のことではない)**

ラベルはすべての利用者に提示される一方で、名前 (name) は隠されていて、支援技術に対してだけ明らかにされる場合もある。多くの場合 (すべてではないが)、ラベルと名前 (name) は同じである。

2.5.3 名前 (name) のラベル (A)

補足

- リンクテキストがアクセシブルな名前 (accessible name) の先頭に一致する

表示されているもの : 選択する: [テレビ](#)

悪い例 `<p>選択する : <a href="…"><span class="sr-only">日本製</span>テレビ</a><p>` nameは 日本製テレビ

良い例 `<p>選択する : <a href="…">テレビ<span class="sr-only">日本製</span></a><p>` nameは テレビ日本製

音声入力ソフトを使っているとき、見えている"テレビ"と発声しても、悪い例では認識しない。

- 見出しと紐づく一般的なリンクテキスト

良い例 `<h4 id="poor">Insufficient Link Names Invade Community</h4>
<p>Citizens are reeling from the growing invasion of useless "read more" links appearing in their online resources. More...`

視覚的に見出しが見えている人は More が何のMoreである想像がつくが、音声で聞いている人にはリンクだけからはわからない。

2.5.3 名前 (name) のラベル (A)

補足

- ラベルはコンボボックス、ドロップダウンリスト、テキストフィールドのすぐ左か上、ラジオボタンやチェックボックスの後ろ、ボタンの中かすぐ下にあることが強く望まれる。そうでない配置をする場合は慎重に検討する。(以下、良い例)

Date of Birth:

Country of origin:

Date of Birth:

Country of origin:

Gender

☐ Male ☐ Female

- 5段階評価のような場合、ボタンに隣接したラベルを配置できない。このような場合は title 属性を用いてラベルをつける。(これで3.3.2も満たす)

Rate your response

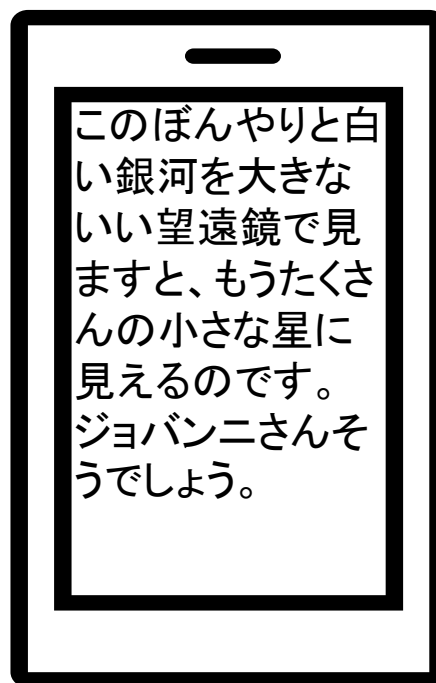
Hated it ☐ ☐ ☐ ☐ ☐ Loved it

```
<fieldset>
<legend>Rate your response</legend>
<label for="hated">Hated it</label>
<input type="radio" name="meal" id="hated" value="hated">
<input type="radio" name="meal" id="poor" value="poor" title="Disliked">
<input type="radio" name="meal" id="neutral" value="neutral" title="So-so">
<input type="radio" name="meal" id="okay" value="okay" title="Liked">
<input type="radio" name="meal" id="loved" value="loved">
<label for="loved">Loved it</label>
</fieldset>
```

Think!

デバイスを傾けて次のページあるいは前のページに進むとき

それに変わる方法は？



2.5.4 動きによる起動 (A)

デバイスの動き又は利用者の動きで操作できる機能は、ユーザインタフェースコンポーネントでも操作でき、かつ偶発的な起動を防ぐために動きへの反応を無効化することができる。ただし、以下の場合には例外とする。

サポートされたインタフェース アクセシビリティ サポートされたインタフェースを通じて機能进行操作するために動きが用いられる。

必要不可欠 その機能にとって動きが必要不可欠であり、この達成基準に従うと動作を無効化してしまう。

意図

カメラに向かってジェスチャをしたりデバイスを傾けたり振ったりといった動作は、身体のコントロールが難しい方やカメラの位置を確認できない方には困難な場合がある。また、傾けるつもりがなくても傾けてしまい誤動作させてしまう可能性もある。よって、他の方法で同じことができるようにし、また、機能の無効化が必要となる。

他の方法とはキーボードやマウスの操作、音声入力、支援技術を使った操作である。



2.5.4 動きによる起動 (A)

対処

- 利用者はジェスチャ入力に基づくUNDO機能などをオフにすることができる。
- テキストを入力後、デバイスを振ると入力を元に戻す機会が与えられる。キャンセルボタンも同じ機能を提供する。
- デバイスを傾けて次、又は前のページに進むことができる。同じ機能を有するボタンやリンクも提供されている。
- デバイスを左右前後に動かして、インタラクティブに写真の表示を変更できる。これらと同じ機能を実行するためのコントロールもある。
- デバイスに向かってジェスチャをしてコンテンツをナビゲートできる。同じ事をするためのコントロールもある。
- 例外には万歩計などある。



Think!

音声ブラウザでの操作中

Webを閲覧中、10000人目の訪問者で、
画面に「あたり! 積丹塩ウニ!」と表示されました。

しかし、何も読み上げなく気が付きませんでした

画面には「いらないの? 消えちゃうよ」と警告が出ました

しかし、何も読み上げなく気が付きませんでした

こんなことを防ぐにはどうしたら良いのでしょうか

4.1.3 ステータスメッセージ (AA)

マークアップ言語を使って実装されたコンテンツでは、ステータスメッセージは、役割 (role) 又はプロパティを通してプログラムによる解釈が可能であり、フォーカスを受けとらなくても支援技術によってユーザに提示することができる。

意図

画面に追加でエラーメッセージなど表示する場合がある。このとき、フォーカスが当たっていれば支援技術が音声化するなどして利用者は知ることができるが、フォーカスが当たっていないと利用者は何も気が付かない可能性がある。そのため、この達成基準では、フォーカスをあてずにメッセージを追加する場合には、支援技術が知ることができるように、役割 (role) やプロパティをメッセージに割り当てるように求めている。

メッセージとはエラーステータスに限らない。どのような情報でも利用者にとって必要な情報であれば対象になる。例えば、アクションの成功や結果、アプリケーションの待ち状態、プロセスの進行状況も対象になる。

4.1.3 ステータスメッセージ (AA)

対処

状況 A: ステータスメッセージが動作の成功もしくは結果、又はアプリケーションの状態を通知する場合:

role="status" --- ログイン、プロセスの各フェーズ(予約、解約、確認など)
ショッピングカートの個数の更新、操作結果、アプリの状態など

状況 B: ステータスメッセージが提案、又はエラーの存在に関する警告を伝える場合:

role="alert" --- 未入力、リストにない値、範囲外の値、修正候補の提示

状況 C: ステータスメッセージがプロセスの進行に関する情報を伝える場合:

role="log" --- 進捗の表示など

補足

- 動的サイトではステータスの変化は常に生じるので、本達成基準は注意が必要。
(sr-only と組み合わせて role でステータスを提供すると良いかもしれない)
- 2が1に変更されるような場合、aria-atomicを用いて「人数は1人」と全体を読み上げさせる。
<p **aria-atomic="true"**>人数は2人</p>
- role="alert"だけでは支援技術は読まない可能性がある。aria-liveを併用する。
<div id="alertmessage" **role="alert" aria-live="assertive"**></div>

4.1.3 ステータスメッセージ (AA)

required 属性について

- コンテンツによっては required属性を用いて必須要素を伝えている。この実装では、ブラウザが状況を伝えるダイアログを表示し、スクリーンリーダーは即座にそれを読み上げ、利用者は必須項目を知ることが出来る。ただし、4.1.3の観点でみた場合、これはダイアログでありステータスメッセージではない。よって、非適用と判定する。

```
<input type="email" value="" name="EMAIL"
placeholder="example@website.com" required class="required email"
id="mce-EMAIL">
```

この方法は、修正案としては採用できる。その場合はつぎのように説明する。

「次の実装により本項目は利用者にとって有効なものであり、4.1.3は非適用とすることができる。」

(ソースにrequired属性を追加する)

(required属性が認識されない場合のために、aria-required="true" も追加する)

(autocompleteが適切であれば追加する。説明には「(autocompleteは 1.3.5の要件)」と加える。)

WCAG2.1で追加された達成基準（レベル順）

達成基準数

- 適合レベル A : 25 → 30 (+5)
- 適合レベル AA : 13 → 20 (+7)
- 適合レベル AAA : 23 → 28 (+5)

Level	番号	達成基準名	説明
A	2.1.4	文字キーのショートカット	利用する場合、解除、再割り当て(alt,ctrlなど)、フォーカス時のみ有効
	2.5.1	ポインタのジェスチャ	ジェスチャに替えて、シングルポインタでも利用可能にする
	2.5.2	ポインタのキャンセル	タッチしただけでスタートは駄目。うまくキャンセルできるように。
	2.5.3	名前 (name) のラベル	ラベルは名前に含まれている。(label要素、name属性の話ではない)
	2.5.4	動きによる起動	例えば顔認識で何か操作するような場合、同じ事がhtmlで可能
AA	1.3.4	表示の向き	表示の向きは縦向き又は横向きに固定しない
	1.3.5	入力目的の特定	フォームにおいて、名前、住所、電話番号などの属性を指定する
	1.4.10	リフロー	デバイスを縦横変えたとき、縦横にスクロールが必要な状況が発生しない
	1.4.11	非テキストのコントラスト	色で区別したエリアなど
	1.4.12	テキストの間隔	行 1.5倍、段落 2倍、文字間隔 0.12倍、単語の間隔 0.16倍
	1.4.13	ホバー又はフォーカスで表示されるコンテンツ	title属性等で表示されるようなコンテンツ 非表示にできること、追加コンテンツの上でポインタを動かせること、表示が継続すること
	4.1.3	ステータスメッセージ	ステータスはroleを用いて利用者に提供する(主としてARIA)
AAA	1.3.6	目的の特定	コンポーネント、アイコン、リージョンなどARIAなどで目的を明確にする。
	2.2.6	タイムアウト	無操作によりデータの損失の可能性がある場合残り時間を警告する。
	2.3.3	インタラクションによるアニメーション	必要不可欠でない限り、モーションアニメーションは無効にできる。
	2.5.5	ターゲットのサイズ	ポインタ入力のターゲットの最小サイズは44x44CSSピクセル以上とする。
	2.5.6	入力メカニズム非依存	入力方法がタッチ入力などに制限されてはならない。

WCAG2.1で追加された達成基準（達成基準順）

達成基準数

- 適合レベル A : 25 → 30 (+5)
- 適合レベル AA : 13 → 20 (+7)
- 適合レベル AAA : 23 → 28 (+5)

番号	Level	達成基準名	説明
1.3.4	AA	表示の向き	表示の向きは縦向き又は横向きに固定しない
1.3.5	AA	入力目的の特定	フォームにおいて、名前、住所、電話番号などの属性を指定する
1.3.6	AAA	目的の特定	コンポーネント、アイコン、リージョンなどARIAなどで目的を明確にする。
1.4.10	AA	リフロー	デバイスを縦横変えたとき、縦横にスクロールが必要な状況が発生しない
1.4.11	AA	非テキストのコントラスト	色で区別したエリアなど
1.4.12	AA	テキストの間隔	行 1.5倍、段落 2倍、文字間隔 0.12倍、単語の間隔 0.16倍
1.4.13	AA	ホバー又はフォーカスで表示されるコンテンツ	title属性等で表示されるようなコンテンツを非表示にできること、追加コンテンツの上でポインタを動かせること、表示が継続すること
2.1.4	A	文字キーのショートカット	利用する場合、解除、再割り当て(alt,ctrlなど)、フォーカス時のみ有効
2.2.6	AAA	タイムアウト	無操作によりデータの損失の可能性がある場合残り時間を警告する。
2.3.3	AAA	インタラクションによるアニメーション	必要不可欠でない限り、モーションアニメーションは無効にできる。
2.5.1	A	ポインタのジェスチャ	ジェスチャに替えて、シングルポインタでも利用可能にする
2.5.2	A	ポインタのキャンセル	タッチただけでスタートは駄目。うまくキャンセルできるように。
2.5.3	A	名前 (name) のラベル	ラベルは名前に含まれている。(label要素、name属性の話ではない)
2.5.4	A	動きによる起動	例えば顔認識で何か操作するような場合、同じ事がhtmlで可能
2.5.5	AAA	ターゲットのサイズ	ポインタ入力のターゲットの最小サイズは44x44CSSピクセル以上とする。
2.5.6	AAA	入力メカニズム非依存	入力方法がタッチ入力などに制限されてはならない。
4.1.3	AA	ステータスメッセージ	ステータスはroleを用いて利用者に提供する（主としてARIA）

WCAG 2.1早見表

WCAG 2.1 早見表 (レベルA & AA)

† WCAG 2.1 で新たに追加されたもの

1 知覚可能

1.1 テキストによる代替	
1.1.1 非テキストコンテンツ	A
1.2 時間依存メディア	
1.2.1 音声のみ及び映像のみ (収録済)	A
1.2.2 キャプション (収録済)	A
1.2.3 音声解説、又はメディアに対する代替 (収録済)	A
1.2.4 キャプション (ライブ)	AA
1.2.5 音声解説 (収録済)	AA
1.3 適応可能	
1.3.1 情報及び関係性	A
1.3.2 意味のある順序	A
1.3.3 感覚的な特徴	A
1.3.4 表示の向き†	AA
1.3.5 入力目的の特定†	AA
1.4 判別可能	
1.4.1 色の使用	A
1.4.2 音声の制御	非干渉 A
1.4.3 コントラスト (最低限)	AA
1.4.4 テキストのサイズ変更	AA
1.4.5 文字画像	AA
1.4.10 リフロー†	AA
1.4.11 非テキストのコントラスト†	AA
1.4.12 テキストの間隔†	AA
1.4.13 ホバー又はフォーカスで表示されるコンテンツ†	AA

2 操作可能

2.1 キーボード操作可能	
2.1.1 キーボード	A
2.1.2 キーボードトラップなし	非干渉 A
2.1.4 文字キーのショートカット†	A
2.2 十分な時間	
2.2.1 タイミング調整可能	A
2.2.2 一時停止、停止、非表示	非干渉 A
2.3 発作と身体的反応	
2.3.1 3回の閃光、又は閾値以下	非干渉 A
2.4 ナビゲーション可能	
2.4.1 ブロックスキップ	A
2.4.2 ページタイトル	A
2.4.3 フォーカス順序	A
2.4.4 リンクの目的 (コンテキスト内)	A
2.4.5 複数の手段	AA
2.4.6 見出し及びラベル	AA
2.4.7 フォーカスの可視化	AA
2.5 入力モダリティ†	
2.5.1 ポインタのジェスチャ†	A
2.5.2 ポインタのキャンセル†	A
2.5.3 名前 (name) のラベル†	A
2.5.4 動きによる起動†	A

3 理解可能

3.1 読みやすさ	
3.1.1 ページの言語	A
3.1.2 一部分の言語	AA
3.2 予測可能	
3.2.1 フォーカス時	A
3.2.2 入力時	A
3.2.3 一貫したナビゲーション	AA
3.2.4 一貫した識別性	AA
3.3 入力支援	
3.3.1 エラーの特定	A
3.3.2 ラベル又は説明	A
3.3.3 エラー修正の提案	AA
3.3.4 エラー回避 (法的、金融、データ)	AA

4 堅牢 (robust)

4.1 互換性	
4.1.1 構文解析	A
4.1.2 名前 (name)・役割 (role) 及び値 (value)	A
4.1.3 ステータスメッセージ†	AA

WCAG 2.1の状況

	WCAG 2.0	WCAG 2.1	WCAG 2.2
Status	Recommendation	Recommendation	Working Draft
A, AA, AAA	61 (25, 13, 23)	78 (30, 20, 28)	88 (33, 25, 30) (understandingより)
Guidelines ガイドライン	2008/12/11 翻訳版あり	2018/06/05 翻訳版あり (2018/6/5)	2021/05/21 なし
Understanding 解説書	2016/10/07 翻訳版あり	2021/7/27 翻訳版あり (2019/3/6)	2021/12/3 なし
Techniques 達成方法集	2016/10/07 翻訳版あり	2021/7/27 翻訳版あり (2019/10/1)	
ISO/IEC	ISO/IEC 40500:2012	なし	-
JIS	JIS X 8341-3:2016	なし (2021年度の改訂作業も無し)	-
コメント			2.3は作る予定はない 2.xと3.0は共存するもの

* 日付のあるものは official version が発行された日

米国では WCAG2.1 は Understanding/Techniques が出揃っていなかったため、訴訟等の際に求める基準としてはずっと控えめなものだった。今、Techniques も出揃ったため、今後は、2.1 の適用が加速する可能性が高い。一方、ISO/IEC 化の動きはあるようだが明確ではなく、欧州では ISO/IEC 化を必須のものとするのか、それとも、W3C のリコメンデーションにそのまま従うのか議論があるらしい。

W3C First Public Working Draft 21 January 2021

- First draft 2020/9/15 - 2021/2/26
- WCAG 2.xの互換ではない。2.x と 3.0 は両方存在しても良い。
- UAAG, ATAGを含む統括的なもの。
- Bronze, Silver, Gold のスコアリングシステムを持つ。
- 今回のレビューはBronzeまでしか含まない。
- Silver, Gold では障害を持つ人にとってのユーザビリティに関する要件を含む。
- テストは Atomic tests と Holistic tests の二種類がある (名称は募集中)
- Atomic tests は従来のもの。Holistic test ではユーザビリティなど含む広いもの。
- Scoring
 Very Poor (0) Any critical errors or less than 50% of related tests pass
 Poor (1) No critical errors, approx. 50% to 79% of related tests pass
 Fair (2) No critical errors, approx. 80% to 89% of related tests pass
 Good (3) No critical errors, approx. 90% to 98% of related tests pass
 Excellent (4) No critical errors, approx. 99% to 100% of related tests pass
- Overall scores クリティカルエラーはなく、かつ、3.5以上でブロンズレベルに適合 (一つのカテゴリーで90%以上という条件も検討されている)

Guidelinesの構成

7.1 Text alternatives

- Text alternative available

7.2 Clear words

- Common clear words

7.3 Captions

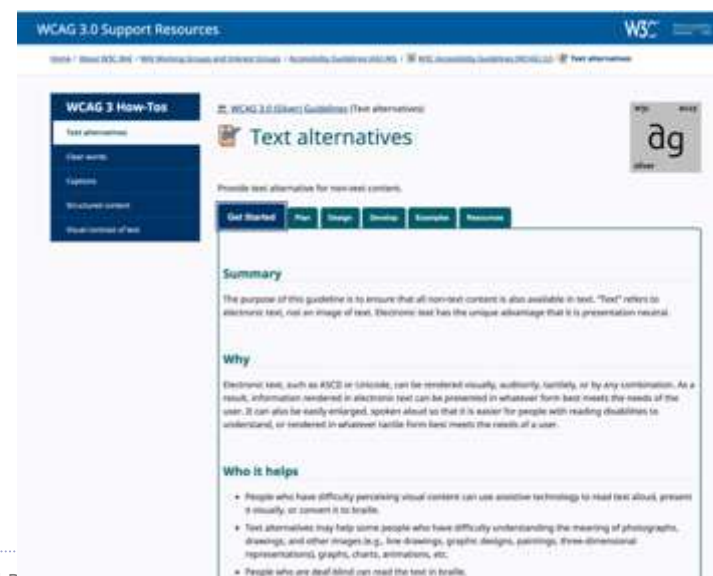
- Translates speech and non-speech audio
- Conveys information about the sound

7.4 Structured content

- Headings organize content
- Uses visually distinct headings
- Conveys hierarchy with semantic structure

7.5 Visual contrast of text

- Luminance contrast between background and text



1.3.6 目的の特定 (AAA)

マークアップ言語で実装されたコンテンツでは、ユーザインタフェース コンポーネント、アイコン、領域の目的はプログラムによる解釈が可能である。

意図

シンボル等を含め、すべてプログラムで解釈可能にすることである。そうすることで、さらに多くの障害を持つ人、特に認知に関する障害を持つ人たちを支援しようと考えている。

対象者

認知の障害により次のような困難を抱えている人

- 記憶
- 集中力及び注意力
- 言語関連
- 実行機能及び意思決定

次のようなインターフェースを改善する

- 利用者が慣れ親しんでいるシンボルとグラフィック
- 少ない機能と少ない認知過負荷
- キーボードショートカット

対処

- ARIA11: ページのリージョンを特定するために ARIA ランドマークを使用する及び
- 主要コンテンツ上のシンボルを有効にするセマンティクスを提供する (COGA Techniques 4.2) 及び
- microdata を利用してユーザインタフェース コンポーネントをマークアップする (リンク追加予定)

補足

- ariaをさらに活用したインプリメントが要求される。例えば、roleとしてbanner, search, navigation, main, complementary, contentinfo などがある。さらに、navigationなどの活用も必要になる。

2.2.6 タイムアウト (AAA)

データの損失を引き起こす恐れのある利用者の無操作の残り時間が警告される。ただし、利用者が 20 時間以上何もしなくてもデータが保持される場合は、この限りではない。

意図

タイムアウトに関する情報をセッションの最初に必ず示すことを求めている。そうすることで、次のような障害を持つ人たちを支援しようと考えている。ただし、20時間以上を保証する場合は適用外である。

対象者

- 言語に関連する障害
- 記憶に関連する障害
- 注意及び集中に関連する障害
- 物事の実行機能及び意思決定に影響を与える障害

対処

- 20時間以上、入力した情報を保持する。
- タイムアウトを発生させない。
- 20時間保持することが現実的でない場合、セッションの最初でセキュリティなど理由を示した上でアプリケーションがもっと短い時間でタイムアウトすることを通知する。例えば、1時間無操作状態であるとタイムアウトプロセスが走るなど知らせる。

補足

利用者の無操作 (user inactivity)

利用者がいかなる操作も行わない時間が続くこと。

そのトラッキングの方法は、ウェブサイトやアプリケーションによって決定される。

2.3.3 インタラクションによるアニメーション (AAA)

アニメーションが、機能又は伝達されている情報に必要な不可欠でない限り、インタラクションによって引き起こされるモーションアニメーションを無効にできる。

意図

一部のアニメーションは注意をさまたげたり、前庭（内耳）障害を発生させ、めまい、吐き気、頭痛を誘発することがある。代表的なものに視差効果がある。

視差効果 - 背景と前景が異なる速度で動く状態

対処

- 視差効果など不必要な動きを無効化するオプションが提供されている。
- ページめくりのアニメーションなど、reduce-motion CSSを使用してモーション軽減設定を有効にする。
- アニメーションが不可欠な場合、プレビューすることを可能にする。

補足

- AAA ではありますが、視差効果は使われがちであり、一定数の困っている人もいますので、対応に含めることを強く推奨します。

2.5.5 ターゲットのサイズ (AAA)

ポインタ入力のターゲットのサイズが 44 × 44 CSS ピクセル以上である。ただし、以下の場合には例外とする。

同等のものが存在する そのターゲットと同等のリンク又はコントロールが同じページに 44 × 44 CSS ピクセル以上のサイズで存在する。

インラインである そのターゲットが文中、又はテキストブロック内に存在する。

ユーザエージェントのコントロールである ターゲットのサイズがユーザエージェントによって定められており、かつコンテンツ制作者が変更していない。

必要不可欠 そのターゲットを特定の方法で提示することが、情報伝達にとって必要不可欠である。

意図

ボタンのようなターゲットのサイズが小さいと、マウスや手指での操作は一層難しくなる。口に咥えたスタイラスペンで操作するような方にとっても難易度が増す。指先より小さなターゲットは、視認しにくくするので、一層操作は難しくなる。特に、次のような状況では重要である。

- 頻繁に使用される。
- 結果を簡単に元に戻せない。
- 手が届きにくい位置、たとえば画面の端に近い場所に配置されている。
- 連続したタスクの一部である。

こうした利用者を考えるとターゲットのサイズは大きい方が良いか、一方で大きくすることが難しい場合もある。この判断を適切に行うことがこの達成基準ではポイントになる。

2.5.5 ターゲットのサイズ (AAA)

対処

- ポインタ入力のターゲットのサイズは 44 x 44 CSSピクセル以上にする。ただし、次は例外とする。

- 同等のものが存在する



[停止] [再生]

再生を示す ▶ ボタンがサイズを満たしていない。しかし、すぐ近くに [再生] と書かれたテキストボタンがある。

- インラインである

注意：本情報は2012年の統計データ(■)に基づく

注意：本情報は2012年の統計データ(■)に基づくもので、現在の状況は不明である。(データは■をクリックすると開く)

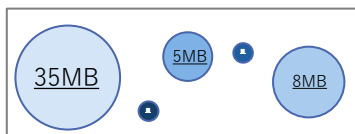
リフローの対象となるインラインのコンテンツである。

- ユーザエージェントのコントロールである



ブラウザ標準のプレーヤーで表示されるボタンなど。

- 必要不可欠



円グラフの大きさで情報量を示している。クリックすると、詳細を見られる。

2.5.6 入力メカニズム非依存 (AAA)

プラットフォームで提供されている入力モダリティの使用を、ウェブコンテンツが制限しない。ただし、その制限が必要不可欠な場合、コンテンツのセキュリティのために必要な場合、又は利用者による設定を尊重するうえで必要な場合は例外とする。

意図

非依存というのは、どのような入力メカニズムでも選択できるということだけでなく、それが一連の操作の中で自由に選べるということを求めている。例えば、大まかな操作はマウスで行い、細かな操作はキーボードの方が適しているかもしれない。利用者と利用者の環境によっては、どの入力デバイスが適しているかなどは決められない。

この達成基準が満たされている場合、次のようなメリットがある。

- 利用者は、好みの入力メカニズムを使用してウェブコンテンツとやりとりができる。
- 利用者は、自らの意図又は状況の求めに応じて、入力メカニズムを切り替えることができる。
- 利用者は、オペレーティングシステムがサポートしていれば、任意で入力メカニズムを追加及び削除することができる。

対処

- JavaScript において、高レベルの、特定の入力方法に依存しないイベントハンドラ (focus、blur、click) のみを使用する (オペレーティングシステム／ユーザエージェントは、すべての入力メカニズムに対してこれらのイベントを発火させる)。
- JavaScript において、キーボードもしくはキーボードのような入力、及びポインタ入力のためのイベントハンドラを、同時に登録する (Pointer Events Level 2 の EXAMPLE 1 を参照のこと)。

EXAMPLE 1: Feature detection and event binding (<https://w3c.github.io/pointerevents/>)

```
/* Bind to either Pointer Events or traditional touch/mouse */
if (window.PointerEvent) {
  // if Pointer Events are supported, only listen to pointer events
  target.addEventListener("pointerdown", function(e) {
    // if necessary, apply separate logic based on e.pointerType
    // for different touch/pen/mouse behavior
    ...
  });
} else {
  // traditional touch/mouse event handlers
  target.addEventListener('touchstart', function(e) {
    // prevent compatibility mouse events and click
    e.preventDefault();
    ...
  });
  ...
  target.addEventListener('mousedown', ...);
  ...
}
// additional event listeners for keyboard handling
...
```