




〇〇〇〇

Web アクセシビリティ・ガイドライン

WCAG 2.0 対応

(2022 年 3 月)



目次

はじめに	3
アクセシビリティとは？	3
本書の構成と内容	3
1. タイトルと言語（家の表札のようなものなのです）	4
1-1. タイトルは <code>title</code> 要素を用いてページの内容を的確に説明しましょう。	4
1-2. ページ言語を、 <code>html</code> 要素の <code>lang</code> 属性を用いて正しく指定しましょう。	4
2. 見出し（ワードや PDF の目次機能と同じなのです）	5
2-1. 見出しに見出し要素（ <code>h1</code> ～ <code>h6</code> ）を使用しましょう。	5
3. リンク（安心して次のページへ進むために）	6
3-1. リンクのボディ（ <code><a></code> と <code></code> の間）において、リンクの目的を十分に説明している又は不十分な場合の代替手段を提供しましょう。	6
3-2. リンク先が別のウィンドウで開く場合には、別ウィンドウで開くということを、リンクテキストの中で説明しましょう。	8
3-3. 隣り合った画像とテキストリンクが同じリンク先の場合、一つの <code>a</code> 要素でマークアップしましょう。	9
4. 画像（ペットボトルを想像してみてください）	9
4-1. 画像に、その画像の目的を適切に表現した代替テキストを提供しましょう。	9
4-2. 色の違いで情報を伝えている画像がある場合、その情報をパターン（模様）やテキストでも提供しましょう。	11
5. 色（コントラスト比の計算は大切です。でも簡単です！）	13
5-1. テキスト（及び文字画像）とその背景色のコントラスト比を維持しましょう。	13
5-2. テキストの色の違いで情報を伝える場合、それ以外にも視覚的な手がかりや追加の説明テキストを提供しましょう。	16
6. 本文へジャンプ機能（すぐに本文を読めるように）	17
6-1. 複数のウェブページ上で繰り返されているコンテンツのブロックについて、スキップする機能を用意しましょう。	17
7. 構造化（マシンリーダーだから便利なんです）	18
7-1. リストに適切なリスト要素（ <code>ul</code> 、 <code>ol</code> 、 <code>dl</code> 要素）を使用しましょう。	18
7-2. テーブルの情報及び関係性は、プログラムにより解釈が可能な状態でマークアップしましょう。	19
7-3. 表形式のコンテンツを、適切なテーブル要素を使ってマークアップしましょう。	19
7-4. テキストの見た目や色によって情報を伝える場合は、セマンティックな要素を使用してマークアップしましょう（段落の表現、行間の調整、強調など）	20
7-5. 単語の途中にスペース、タブ文字、改行文字などの使用がなく、支援技術が単語を正しく認識できますか？	20
7-6. HTML で書かれたソースコードが仕様に準じていることを確認しましょう。	21
7-7. HTML で書かれたソースコードが支援技術にも解釈が可能なようにマークアップしましょう。	21
8. 映像・音声	22
8-1. 音声のみ（映像無し）あるいは映像のみ（音声無し）のコンテンツに、書き起こしや台本などの代替コンテンツを提供しましょう。	22
8-2. 動画コンテンツの音声にキャプションを提供しましょう。	22

8-3. 動画コンテンツの映像に、時間に依存した情報の説明を含む代替コンテンツ又は音声解説を提供しまし う。	22
9. 入力フォーム	24
9-1. フォーム・コントロール（テキストフィールドやラジオボタンなど）に適切なラベルを関連付けましょ う。	24
9-2. セレクトメニューを選択すると他のコントロールの値が自動更新される場合、 select 要素の onchange イベ ントを使用しましょう。	24
9-3. 入力が必要である項目を未入力にして送信した場合、どの箇所を入力しなかったのかが伝わるように表示し ましょう。	24
9-4. 特定の書式や値に制限のあるテキスト入力項目で意図的に入力エラーを起こした場合、エラーの箇所や内容 をテキストで提供しましょう。	25
10. キーボード（マウスのない世界なんて・・・）	26
10-1. フォーカスが左から右、上から下に順序正しく移動するようにしましょう。	26
10-2. キーボード操作でダイアログを開いた場合、継続してそのダイアログを操作できるようにしましょう。	26
10-3. キーボード操作でもマウスと変わりなく操作できるようにしましょう。	27
10-4. フォーカスを与えただけで操作に混乱を与えるようなコンテキストの変化（内容が変わる、操作方法が変わ るなど）が発生しないようにしましょう。	28
11. 制限時間（時間のかかる場合もあるんです）	30
11-1. 操作や読むことに制限時間がある場合、解除、調整、延長のいずれかの機能を提供しましょう。	30
11-2. 動きがある、点滅している、スクロールする、自動更新するコンテンツがある場合、一時停止などの必要な 機能を提供しましょう。	30

はじめに

本ガイドラインは、障害をもつ利用者でも利用できる Web ページについて、つまりアクセシビリティに対応した Web ページを作成するための、ポイントをまとめたものです。

ガイドラインで示しているポイントを守れば、最低限のアクセシビリティに対応した Web ページを作成できます。また、アクセシビリティの配慮は障害をもつ方のみならず、多くの利用者にとっての利益に繋がります。怪我や病気などにより一時的に身体が不自由になった場合でも利用できます。また、見出しなどで素早く必要な情報にたどり着け、表示デバイスや利用環境の違いによる不便などを解消できるため、誰にとっても使いやすさや理解のしやすさにつながります。

このガイドラインに従って Web ページを作成すれば、だれでも利用しやすい Web ページおよび Web サイトが作成できます。最後にチェックリストで最終チェックもできます。

アクセシビリティとは？

一般にアクセシビリティとは、アクセスのしやすさを意味します。転じて、製品やサービスの利用しやすさという意味でも使われますが、ここでは、障害をもつ利用者でも、PDF 文書が利用できるよう配慮し「事前に準備をすること」と考えてよいでしょう。

大切なのは「利用のしやすさ」ではなく「アクセスのしやすさ」です。障害をもつ利用者や怪我や病気で一時的に身体が不自由な状況になったとき「まったく利用できない」ということが発生しないように、そのような状況でも、最低限のアクセス手段を事前に準備しておくこと、と考えてもよいでしょう。

本書の構成と内容

本書は、最低限のアクセシビリティに対応した Web ページを作成ためのポイントを、項目別に解説しています。確認ポイントや、項目によっては、悪い例と良い例を示して解説しています。また、参考として、実際に読み上げ操作やキーボード操作の演習事例も示しています。

各項目には、達成基準/達成方法の項目を設けて、WCAG2.0 の実際達成基準の文書や、達成方法の技術解説書へのリンクを設けていますので、必要に応じて、Web アクセシビリティのより詳細な情報へアクセスすることも可能です。

1. タイトルと言語 (家の表札のようなものなのです)

1-1. タイトルは title 要素を用いてページの内容を的確に説明しましょう。

例えば、複数のタブを開いているとき、タイトルがない、あるいは全て同じように見えたなら不便ではありませんか？実は、音声ブラウザの利用者にとってのページタイトルは、最初に聞こえてくる情報なのです。目的のページに辿り着いたかどうかを確認する上で、とても重要です。「家」に例えれば、その家の「表札」のようなものなのです。



確認ポイント

1. head 要素の中に title 要素を定義してください。
2. title 要素のテキストはページの内容を的確に説明するテキストにしてください。
3. 他のページと同じテキストにならないようにしてください。
4. 同じようなタイトルのテキストになってしまう場合は、できるだけテキストの最初のほうで他のページと違うことを示すようにしてください。

達成基準/達成方法

- A : [2.4.2](#) / [G88](#), [H25](#)

1-2. ページ言語を、html 要素の lang 属性を用いて正しく指定しましょう。

たとえば、日本語のページなのに、ブラウザが別の言語だと勘違いして、翻訳を提案してきたり、意図しない外国のフォントで表示されることはありませんか？音声ブラウザでも同様に、lang 属性を見て、音声合成エンジンを切り替えます。外国語の音声合成で読み上げても内容の理解かなり難しくなります。



確認ポイント

1. html 要素に lang 属性を定義してください。
2. lang 属性の属性値は正しい値を設定してください。(ja や en)

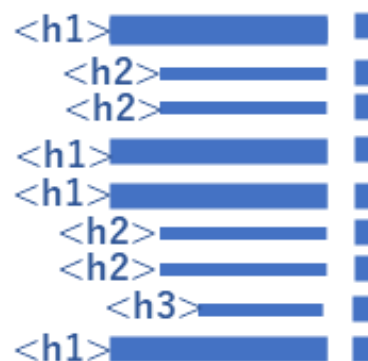
達成基準/達成方法

- A : [3.1.1](#) / [H57](#)

2. 見出し（ワードや PDF の目次機能と同じなのです）

2-1. 見出しに見出し要素（h1～h6）を使用しましょう。

ワードや PDF では、見出しを検出して自動で目次(しおり)が作成されますが、HTML の見出し要素(h1～h6)もそれと同じです。目次があれば、読みたいところへすぐにジャンプして読むことができます。通常のブラウザは、目次を作成する機能は見かけませんが、音声ブラウザの利用者は、目次のように見出し要素を利用して、そのページの大雑把な構造を把握します。もし見出しが適切に設定されていないと、目次としてページの大雑把な構造を把握する役割が果たせなくなります。また、見出し要素を使わずに、大きな文字で見出しのようにすると、もはや目次は存在しないことになります。



確認ポイント

1. 見出し要素(h1～h6)は必ず定義してください。(※ただしページの構造上見出しがない場合は除く)
2. そのページの見出し要素を集めたとき、目次として機能するようにしてください。
 - 視覚的また文書の構造で見出しと考えられるところは見出し要素を使ってください。
 - 見出しレベルは、論理的な構造から考えられるレベルに合わせてください。
 - 見出しの視覚的表現のために見出しレベルを変更することはしないでください。
 - 見出しではないのにフォントを変えるために見出し要素を使わないでください。
 - 見出しがその下の内容を的確に表しているテキストにしてください。
 - 章立てに合わせて、飛び飛びの見出しレベルを設定することは避け、正しい階層構造で見出し要素を使用してください。

ナレーターで読み上げてみましょう

1. Windows+Ctrl+Enter をおして、ナレーターを起動します。
2. Web ブラウザー(Edge 推奨)を起動して読み上げたいページを表示します。
3. h または Shift+h を押して、見出しのジャンプと読み上げをしてみましょう。
4. Windows+Ctrl+Enter をおして、ナレーターを終了します。

達成基準/達成方法

- A : [1.3.1](#) / [H42](#)
- AA : [2.4.6](#) / [G130](#), [G131](#)

3. リンク（安心して次のページへ進むために）

ご承知の通り、リンクは次のページに進むためのものです。しかし「〇〇〇〇」といった、伏字になっているリンクがあったら、どうでしょうか？このリンクを開いたら、どのページへ進むのか？なにが起こるのかわからないので、不安ではないでしょうか？実は、残念ながら音声ブラウザの利用者は、このような状況によく遭遇します。そのようなことが無いように、つぎの項目を確認していきましょう。



3-1. リンクのボディ（`<a>`と``の間）において、リンクの目的を十分に説明している又は不十分な場合の代替手段を提供しましょう。

確認ポイント

1. リンクのボディがテキストのみの場合
 1. リンクの内容や機能（開くページ等）を的確に説明しているテキストにしてください。
2. リンクのボディが画像のみの場合
 1. 画像に alt 属性を定義してください。
 2. alt 属性がリンクの内容や機能（開くページ等）を的確に説明されているようにしてください。
3. リンクのボディがテキストと画像の両方の場合
 1. テキストと画像の alt 属性の内容が一体で、リンクの内容や機能（開くページ等）を的確に説明されているようにしてください。
 2. テキストと画像の alt 属性の内容が同じ内容で繰り返しの内容にならないようにしてください。

悪い例

- PDF や外部ファイルを開くリンクなのにその説明がない。
- リンクボディに画像が設定されているが画像の alt 属性が定義されていない。
- リンクボディにテキストと画像が設定されているが、テキストと画像の代替テキストが同じで、読み上げると2回同じテキストが読み上げる状態になっている。
- リンクボディのテキストが「サイト B へ」となっているのに、実際は「サイト A」が開く

補足

- 隣接する画像とテキストリンクが同じ URL を指しているにも関わらず分割されていることがよくあります。このような場合、画像の代替テキストを空にしてしまうと達成方法 H30 で不適合になってしまいますので、画像とテキストリンクは一つのリンクにした上で、画像の代替テキストを空にします。
`花の一覧`

- 表示されているテキストだけでリンクの目的を十分に提供できない場合は、次のいずれかの方法でリンクを補足してください。
 - **aria-label 属性により情報を提供する (https://waic.jp/docs/WCAG-TECHS/ARIA8)**
href 属性を持つ a 要素に aria-label 属性を与えると、リンクテキストの代わりに aria-label 属性で指定されたテキストを音声ブラウザは読み上げます。

```
<a href="sample" aria-label="このテキストを読み上げる">このリンクテキストは読み上げない</a>
```

- **CSS を用いた非表示のテキストにより情報を提供する (https://waic.jp/docs/WCAG-TECHS/C7)**
overflow プロパティの値が hidden に指定された縦横 1 ピクセルのボックスの中にテキストが入るようにし、さらにそのテキストをビューポートの外側に配置する指定を入れることで、非表示のテキストを指定する CSS セレクタを提供することができます。例えば、screen-reader-text をその目的に用意したとして次の様に用います。

```
<a href="sample">表示テキスト<span class="screen-reader-text">隠しテキスト</span></a>
```

この場合、表示されているのは「表示テキスト」ですが、音声ブラウザは「表示テキスト 隠しテキスト」と続けて読み上げてくれます。

クラス visually-hidden は様々提供されていますが、whitehouse.gov では screen-reader-text として同様のものが提供されていますので、こうしたものを参考にすると良いでしょう。

- ※ visibility:hidden 及び display:none を隠しテキストの目的で使用することはできません。これらの方法で非表示にしたテキストは、音声ブラウザも無視してしまいます。

ナレーターで読み上げてみましょう

1. Windows+Ctrl+Enter をおして、ナレーターを起動します。
2. Web ブラウザー (Edge 推奨) を起動して読み上げたいページを表示します。
3. Tab または Shift+Tab を押して、リンクジャンプと読み上げをしてみましょう。
4. Windows+Ctrl+Enter をおして、ナレーターを終了します。

達成基準/達成方法

- A : 1.1.1 / H30
- A : 2.4.4 / C7, G189, G53, G91, H24, H30, H33, H77, H78, H79, H81, SCR30

3-2. リンク先が別のウィンドウで開く場合には、別ウィンドウで開くということを、リンクテキストの中で説明しましょう。

リンクを開くと、別ウィンドウ(別タブ)で開くリンクは、便利な機能ですが、開いたリンクが別ウィンドウ(別タブ)だった場合、前のページに戻ろうとしたときに、通常の戻る操作では戻れないため、混乱してしまいます。また、開いたウィンドウが隠れてしまったり、どのタブが、いま開いたタブなのかわからなくなってしまうことがあります。音声ブラウザなど支援技術を利用する利用者も同様に、隠れてしまったウィンドウや、いま開いたタブを探すわけですが、この探す作業に一苦労することが多いのです。リンクテキストに明確に説明されていれば、開く前に心構えができて、探す時間を無駄にせずに済むのです。



確認するポイント

1. `target="_blank"` や `target="_new"` の指定をした場合は、別ウィンドウで開かれることを、リンクテキスト中で説明してください。リンクとして画像を指定して `alt` 要素で説明してもかまいません。
2. ただし、`display:none` や `visibility:hidden` の指定は、音声ブラウザからも隠されてしまうため利用しないでください。

キーボードで操作してみよう。

1. Web ブラウザー(Edge 推奨)を起動して別ウィンドウで開くリンクのあるページを開きます。
2. Tab キーで目的のまで移動します。
3. Enter でリンクを開きます。別ウィンドウまたは別タブでリンクが開きます。
4. ここで、前ページに戻ろうとして Alt+左矢印を押しても戻りません。
5. あ、別ウィンドウ開いたかな？と思って Alt+F4 を押すとブラウザが閉じようとしています。
6. あ、別タブか！と思って Ctrl+w を押すと、タブが閉じて前のページに戻ります。

達成基準/達成方法

- A : 2.4.4 / G91

3-3. 隣り合った画像とテキストリンクが同じリンク先の場合、一つの a 要素でマークアップしましょう。

同じリンク先であるにも関わらず、それぞれリンク要素でマークアップすると、同じリンクが2つあることになるため、どちらを開くべきか迷い、支援技術の利用者が混乱する可能性があります。

確認ポイント

1. 確認しようとしているリンクの前後にリンクがある場合は、その前後リンクが同じでないことを確認してください。
2. リンクが連続しているところで、`target="_blank"` や `target="_new"` の指定をした場合は、別ウィンドウで開かれることを、リンクのボディーの中で説明してください。リンクボディー内に画像を指定して `alt` 要素で説明してもかまいません。

補足

- `display:none` や `visibility:hidden` の指定は、音声ブラウザからも隠されてしまうため利用しないでください。

達成基準/達成方法

- A: [2.4.4](#) / H2

4. 画像（ペットボトルを想像してみてください）

画像はページを構成するうえで大切な要素です。音声ブラウザの利用者は、その画像の `alt` 属性のテキストを読み上げることで画像について知ることができます。

4-1. 画像に、その画像の目的を適切に表現した代替テキストを提供しましょう。

突然ですが、お店で買ってきたペットボトルを想像してみてください。ペットボトルの中身が無色透明な場合、「普通のミネラルウォーター」なのか「炭酸水」なのか「桃フレーバーのミネラルウォーター」なのか？なにで判断するのでしょうか？ペットボトルに巻かれている商品ラベルではないのでしょうか？ 画像の `alt` 属性は、このペットボトルのラベルのような存在です。ペットボトルにラベルがなかったら、とても困ります。一方で、「桃フレーバーのミネラルウォーター」のラベルなのに、本当の中身は「普通のミネラルウォーター」だったら、悲しみに包まれてしまうのではないのでしょうか？実は、残念ながら音声ブラウザの利用者は、このような状況によく遭遇します。そのようなことが無いように、つぎの項目を確認していきましょう。