



アクセシビリティ・ガイドライン

アクセシビリティ

障害があるかどうかにかかわらず、すべての人がWebにアクセスできること。

障害のある人がWebを利用する際の不便さを無くし、高齢者を含む全ての利用者が同じようにWeb上の情報にアクセスし利用できることは、利便性を高める上で非常に重要である。

障害者や高齢者に配慮したWebページは、結果的にすべての人にとってアクセシブルなページになる。

向上のポイント

さまざまな環境を思い浮かべ、ユーザーの設定や好み、操作を制約しないようにページ作りをすることである。

ガイドライン

- 聴覚的・視覚的な内容には、同等の役割を果たす代替情報を提供する
 - img要素 alt属性で適切な代替テキストを指定する。
 - input要素、type="image" alt属性で適切な代替テキストを指定する。
 - button要素、img要素を含む場合 alt属性で適切な代替テキストを指定する。
 - script要素 noscript要素で適切な代替内容を指定する
 - object要素 説明文(title)やリンク(data)を含める
- 色だけに頼らない。

- 色で表現されるすべての情報は、色が識別できない環境でもテキストやグラフィックが理解できるようにする
- マークアップとスタイルシートを適切に使用する
 - 文書型宣言で文書型適宜を明示する。
 - 文法の正確さをチェックするために、検証サービスを利用する。
 - 非推奨要素や非推奨属性は使わない。
 - 特定の要素をレイアウト目的で使わない。（例、table要素）
 - 文書構造を指定するための要素で文書を適切にマークアップし、視覚表現を指定するための要素や属性ではなくスタイルシートで表現をコントロールする
 - font-size pt、pxなどの指定ではなく、「em」で指定する
 - 見出しは、h1～h6要素を使用する。
 - ul、ol、dl要素 インデント目的などに使わず、構造的にリストとして定義するのが適切な場合に使う。
 - blockquote要素、q要素 引用文はそれを表す要素でマークアップする。
- 自然言語の扱い方を明確にする
 - xml:lang属性とlang属性で言語コードを指定する
 - 文書内の省略語や頭字語は、その語が最初に現れる部分で正式名称を、abbr要素・acronym要素を使用し示す
 - html要素では、xml:lang属性とlang属性で言語コードを指定する
- 正しく変換されるテーブルを作成する
 - ヘッダセルとデータセルを明確にする th要素・td要素
 - thead・tfoot・tbody要素、colspan要素を使用し、セルを構造的にグループ化する
 - 表をレイアウト目的で使わない
 - 表には、table要素のsummary属性やcaption要素で要約をつける。
 - th要素ではabbr属性でセルの省略語を指定する
- 新しいテクノロジーを利用したページは、正しく変換されるようにする
 - スタイルシートなしでも読めるように文書を構造化する
 - link要素でアクセシブルな代替文書を指定する

- object要素では、適切な説明文やリンクを含める
- script要素 noscript要素で適切な代替内容を指定する
- 時間の経過とともに変化する内容は、ユーザーがコントロールできるようにする。
 - 一時停止または中止、スキップできるようにする
 - 画面を点滅させるようなスクリプトは使用を避けるべき
 - 文書を自動更新するようなページはつくらない
 - 自動的にリダイレクトするマークアップは使用しない。サーバー側でリダイレクトするように設定する。

例：<meta http-equiv="refresh" />は使用しない
- 文書内に埋め込まれたユーザーインターフェイスは、直接的なアクセシビリティを備えたものにする
 - 要素の内容に適切な説明文やリンクを含める
 - スクリプトやFlashなどは、ユーザーが再生や一時停止などを操作できるようにする。
- 入出力装置に依存しないようにデザインする
 - クライアントサイド・イメージマップを使用する。（リンク）
 - リンクやフォームコントロール要素やobject要素には、tabindex属性でtab移動順序を指定する
 - a要素のうち重要なものには、accesskey属性でアクセスキーを指定する
フォームコントロール要素も同様
- 暫定的な解決法を採用する
 - 古いブラウザでも正確に操作できるようにアクセシビリティの解決策を採用する
 - ユーザーに知らせることなく新規ウィンドウを開いたりしない
- 前後関係や位置を示す情報を提供する
 - h1～h6要素は文書構造の明確化のために適切に使う
 - 長い文章はp要素を使用して適切に段落分けする
 - ul、ol、dl要素は、構造的にリストとして定義するのが適切な場合に使う。
 - グローバルナビゲーションなどに含まれるリンクは、a要素でリストを利用し適切にグループ化する。

- `thead`・`tfoot`・`tbody`要素、`colspan`要素を使用し、セルを構造的にグループ化する
- `fieldset`要素を利用し、フォームコントロールをグループ化する
- `select`要素では、`optgroup`要素を使用し、`option`要素をグループ化する
- フォームコントロール要素とラベル要素を明確に関連づける
- 明確なナビゲーションの仕組みを提供する
 - `a`要素には具体的なテキストを指定する。
 - 同じテキストが異なるリンク先を示すのは、ユーザーを混乱させるので望ましくない。この場合、`title`属性で異なる説明文を指定する。
 - ページやサイトの意味的な情報を付与するためにメタデータを提供する
MIMEタイプ、説明文、キーワードなど
 - `link`要素で他の文書との関係を指定する。例：目次(`contents`)、前ページ(`prev`)、次ページ(`next`)
 - グローバルナビゲーションは全てのページで同じ位置にする
- 文書を明瞭で簡潔なものにする
 - よりカンタンに理解しやすいようにする
 - 見出しと段落の使用
 - グローバルナビゲーションは全てのページで同じ位置にする
 - スタイルシートを外部ファイル化し、すべての文書に統一的なスタイルを適用する
 - `style`要素は使用しない。`class`名や`id`名を使用し統一する
- 文書がうまく変換されるようにする
 - 文書構造から視覚表現を分離させる
 - テキストによる情報を提供する