

第1章 導入

1 はじめに

1.1 ウェブアクセシビリティとは？

「ウェブアクセシビリティ」という言葉は、あまり聞き慣れない言葉だと思います。これは、「高齢者や障害者など、心身の機能に制約のある人でもウェブで提供されている情報に問題なくアクセスし利用できること」を意味しています。

高齢者や障害者にとっても、ウェブは重要な情報源になりつつあります。しかし、実際に提供されているウェブページは、高齢者や障害者にとって非常に使いにくい、あるいは使えない作りになっていることが少なくありません。すぐに分かる例として、非常に小さい文字でたくさんの情報を提供しているページがよく見受けられますが、このようなページは弱視の人や多くの高齢者が判読できない場合があります。このようなウェブページを「アクセシビリティが確保されていないページ」と言います。

1.2 障害者・高齢者がウェブを利用するときの問題

ウェブコンテンツを利用する際、どのようなことが問題になるのかは、利用者の身体条件によって異なります。ここでは、障害者・高齢者がウェブを利用する時、どのような問題が発生するかを身体条件別に整理して説明します。実際には非常に多くの問題が発生しますが、ここでは代表的な問題を選んでご紹介します。

全盲の人の場合

全盲の人は、HTML のテキスト情報を音声化してくれる読み上げソフトや、点字ディスプレイを使ってウェブを利用します。これらの方法では、ウェブページ上の画像や Flash 等に代替テキスト（Alt 文）がない場合に、内容を知ることができません。

テキストで書いた情報であっても、テーブルでレイアウトする際に段組が適切でないと、情報の意味する順序と読み上げの順序とに矛盾が生じてしまい、理解が難しくなります。また、データ表など二次元の配置のある情報も、一行ずつ上から順番にしか読むことができない読み上げソフトの利用者にとっては、理解しにくい情報になります。

ページ上部やページ左側などにナビゲーションリンクがたくさんあり、読み上げソフトではなかなか本文にたどり着けないサイトが多いことも、問題となっています。全盲の利

ユーザーは、ページ内ではリンク部分だけを読み上げて情報を探す傾向があるため、リンク文の表現が不適切だと必要な情報を探し出すことが困難になります。



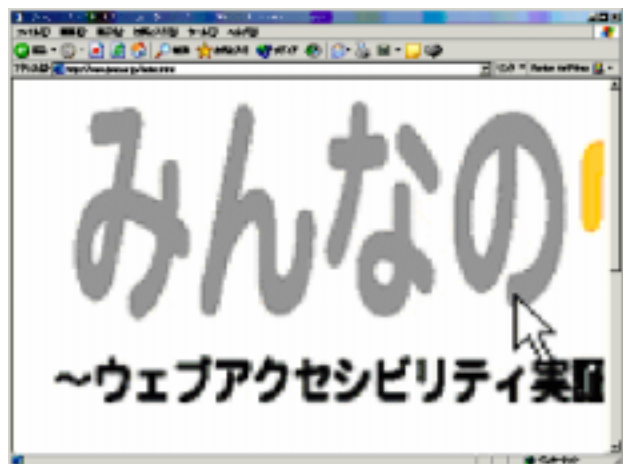
[一般的なブラウザでの表示例]



[テキストブラウザや
音声ブラウザでの表現例]

弱視の人の場合

ひと口に弱視と言っても、見え方や見えやすい条件に多様な個人差があります。視覚でのウェブ利用がある程度可能な場合、画面上の細かい文字の判読は困難なため、画面に顔を近づけたり、パソコンやブラウザ等の設定で文字の大きさや配色を変更して、情報を読みとります。HTML や CSS で文字が固定サイズで指定されていたり、文字情報が画像で提供されていると、ブラウザ等の設定変更で文字を拡大することができず、問題になります。



[拡大ブラウザでの表示例]

文字を拡大するとページが縦に長くなり、画面全体のレイアウト構成が把握しづらくなったり、ページの下の方にある情報や機能に気づくのが遅くなってしまいます。

画像等の特定の情報を詳しく確認したい場合等には、拡大レンズや専用の文字拡大ソフトを併用する人もいます。この時も、一度に確認できる範囲がかなり狭くなるため、全体

のレイアウト等を把握しにくくなります。また、大量の長文を読み流したい時や視覚での利用が困難な場合は、全盲の人と同様に音声読み上げソフトを利用することもあります。この場合は、全盲の人と同様、読み上げソフトを利用する際の問題が発生します。

色覚障害の人の場合

色覚障害とは、ある色とある色の組み合わせが判別できない、または判別しにくいことで、日本人の1割以上の人何らかの色覚障害を持っていると言われています。

最も多く見られるのが赤と緑に関係する色で混同が起きる色覚障害です。また、高齢になると白内障などによって水晶体に変化が起こり、黄色と青に関係する色の区別がつきにくくなる場合があります。

ウェブページ上でこれらの色の組み合わせが不適切に用いられていると、文字や図の判別ができなかったり、色を使ったナビゲーションが理解できないといった問題が発生します。

聴覚障害の人の場合

難聴あるいはまったく耳が聞こえない人の場合、ウェブで提供される音の情報が問題になります。例えば、画面と一緒にBGMや効果音が流れていても気がつかないといった問題です。音声付きのムービーでは、字幕やテキストによる解説がないと、内容が十分把握できません。また、手話を使って会話をしている人の場合、難しい文字表現が分かりにくいケースがあります。

肢体不自由の人の場合

病状や障害の状況によって身体の動かせる機能や程度に個人差があり、キーボードの設定変更や、特殊な入力装置・支援ソフトを利用するなど、様々な状態・利用環境でウェブが利用されています。

細かい手の動きができない人の場合、トラックボール等を使ってウェブを利用するケースがあります。この場合、スクロールが必要な縦長のページや、ページごとにボタンの位置が統一されていないサイトは、マウスポインタの大きな移動が負担となります。また、リンク箇所が小さかったり、隣同士のリンクが近づきすぎている場合や、マウスを乗せた時だけ表示されるメニューを選択する場合など、マウスポインタの細かな操作も困難です。

マウスやトラックボールを使うことが困難な場合でも、1つ、または2つ以上のスイッチやセンサー等を使って、ウェブや画面上のソフトキーボードを操作することができます。スイッチ等が1つの場合は、ウェブページのリンクや画面上のソフトキーボードを順にス

キャンさせ、スイッチ等で選択します。スイッチ等が2つ以上の場合は、1つのスイッチでマウスポインタを移動させ、もう1つのスイッチで選択するなど、それぞれのスイッチに機能を割り当てて利用します。

これらの場合、マウス等での操作と異なり、リンクを1クリックで選択することができません。TAB キーでひとつずつリンクを移動して選択するのと同じ状態で、1ページに含まれるリンク数が多いと、目的のリンクに辿りつくまでに大変な労力と時間が必要になります。

知的障害の人の場合

知的障害には様々なタイプ、程度がありますが、全般に文字で提供された情報が十分に理解できないケースが多いと言えます。ウェブで重要な情報を提供する場合、文字だけでなく、その内容をイラストにしたり、音声でも提供できるようにすることが望まれます。

高齢者の場合

加齢に伴い、視覚・聴覚が変化するため、高齢者がウェブを利用する際には視覚障害者・聴覚障害者双方と共通の問題が発生します。

また、個人差はありますが、全体的な傾向としてはIT関係の専門用語や新語は苦手な人が多く、複雑な操作の習得も負担となります。

高齢者に起こりやすい問題としては、たくさんのサイトを調べているうちに現在の位置や履歴を見失ったり、ボタン名が英語になっていると操作を敬遠するという問題があります。また、直前に見ていた画面に戻る操作がページによって異なるため、戸惑う現象も多く見られます。適切な情報の整理、次が想像できるボタン名、わかりやすい操作手順への配慮がなされていないページは、高齢者にとってはバリアの高いページとなります。

1.3 ウェブに関する技術開発と国際的な標準化

HTML コンテンツのアクセシビリティを確保するための手段や方策については、ウェブに関する技術開発と標準化に取り組んでいる国際的学術団体のW3C (World Wide Web Consortium)^{*1}が、「ウェブコンテンツ・アクセシビリティガイドライン 1.0」(Web Content Accessibility Guidelines 1.0^{*2}、以下 WCAG1.0 と略す)を 1999 年 5 月に勧告しています。これは、W3C の中に設置された WAI (Web Accessibility Initiative)^{*3} というウェブアクセシビリティに関する検討グループによりまとめられたもので、参考にすべき指針として広く知られています。

日本国内でも、郵政省(現総務省)と厚生省(現厚生労働省)が共同で開催した「情報バリアフリー環境の整備の在り方に関する研究会」において、「インターネットにおけるアクセシブルなウェブコンテンツの作成方法に関する指針」(平成 11 年 5 月)が発表されていますが、これも WCAG1.0 の内容にしたがって策定されたものです。

WCAG1.0 では、以下に示す 14 項目のガイドラインと、ガイドラインごとに一般的なコンテンツ制作の過程でどのように適用されるかを解説したいくつかのチェックポイントが提示されています。

WAI では、WCAG1.0 の検討に併せて、チェックポイントごとに一般のウェブコンテンツのアクセシビリティ確保状況をどの様に評価し、また必要に応じて修正する方法についての検討にも取り組んでおり、その内容はワーキングドラフト^{*4}として公開されています。それが「Accessibility Evaluation and Repair Tool」(以下、AERT と略す)^{*5}です。

^{*1} <http://www.w3.org> (英語)

^{*2} <http://www.w3.org/TR/WCAG10/> (英語)

大藤氏による日本語訳が <http://www.zspc.com/> に掲載されている。

^{*3} <http://www.w3.org/WAI/> (英語)

^{*4} ワーキングドラフト: W3C の標準化作業の一環として、勧告に向けた技術文書を作成する議論の途上において、暫定的に公開された技術文書のこと。

^{*5} <http://www.w3.org/TR/AERT> (英語)

【WCAG1.0 における 14 のガイドライン】

1. 聞くための内容や見るための内容には、同等の役割を果たす代替りのものを提供する
2. 色だけに依存しない
3. 正しくタグ付けし、適切にスタイルシートを使う
4. 自然言語の取り扱い方に関する情報を明確に示す
5. うまく変換されるテーブルを作る
6. 新しい技術を利用したページは、うまく変換されるようにしておく
7. 時間とともに変化する内容については、ユーザーが制御できるようにする
8. ページ中に組み込まれたもののユーザーインターフェイスは、それ自体がアクセシブルなものにする
9. 装置に依存しないように設計する
10. 暫定的な解決策をとる
11. W3C のテクノロジーとガイドラインを使用する
12. 前後関係や位置を表す情報を提供する
13. はっきりとわかるナビゲーションのための仕組みを提供する
14. 文書は明瞭で簡潔なものにする

(出所) 大藤氏による WCAG1.0 の日本語訳 (<http://www.zspc.com/>) より

【WCAG1.0 チェックポイントの優先度と適合度について】

WCAG1.0 では、14 個のガイドラインごとに複数のチェックポイントが挙げられており、合計 65 個のチェックポイントがあります。それぞれのチェックポイントは、アクセシビリティに与える影響に基づく「優先度」が指定してあり、[優先度 1] から [優先度 3] までの 3 種類があります。

優先度	重要性	WCAG1.0 での位置づけ
優先度 1	高い	コンテンツ制作者が満たさねばならない優先度
優先度 2	やや高い	コンテンツ制作者が満たすべき優先度
優先度 3	やや低い	コンテンツ制作者が満たすべき事を望まれる優先度

そのうえで WCAG1.0 では、それぞれの優先度を満たしているコンテンツに対して、3段階の「適合度」を規定しています。

適合度	WCAG1.0 での位置づけ
A (A)	優先度 1 のすべてのチェックポイントに適合
ダブルA (AA)	優先度 1 と 2 のすべてのチェックポイントに適合
トリプルA (AAA)	優先度 1 ~ 3 のすべてのチェックポイントに適合

2 ウェブヘルパーについて

ウェブヘルパーは、インターネット上のコンテンツが、全ての人にとってアクセスしやすいものになることを目指して開発されたシステムです。ウェブヘルパーを使うことで、日本語ウェブコンテンツのアクセシビリティ状況を容易に点検することができます。

2.1 点検のしくみ

ウェブヘルパーは、W3C/WAIによるWCAG1.0およびAERTワーキングドラフトを参考に、日本語環境での利用条件を考慮して開発されました。

一般的に、WCAG1.0の14項目のガイドライン、65項目のチェックポイントにしたがって一つずつ手作業で点検しながらアクセシブルなホームページを作るには、相当な専門知識や時間が必要となります。ウェブヘルパーは、ホームページのアクセシビリティを自動的に点検し、制作者が簡単にアクセシビリティ上の問題点を修正できるよう、支援します。

ウェブヘルパーは、WCAG1.0の優先度と適合度に準じて、シングルAからトリプルA(AAA)までの3段階の点検レベルを用意しています。希望する適合度に合わせて、チェックポイントをまとめて点検することができます。

また、点検項目の中には、例えば「意味の違いが色の違いだけで区別されていないか」等、制作者自身がコンテンツの内容や意味を確認した上で、アクセシブルかどうか判断する必要がある項目もあります。その場合も、確認が必要な項目や箇所をウェブヘルパーがおおよそ自動的に検出しますので、点検作業を効率的に、もれなく行なうことができます。

重要！

ウェブヘルパーは、あくまでアクセシビリティ点検のための“補助ツール”です。最終的にアクセシビリティの確保が実現されているかは、WCAG1.0に照らし合わせて、ご自身で判定していただく必要があります。

2.2 点検項目

ウェブヘルパーの点検項目は、AERT に沿ってリストアップしました。したがって項目番号の1桁目は、該当する WCAG1.0 のガイドライン番号に、2桁目はそのチェックポイントの番号になっています。

「項目番号」の後ろに、*(アスタリスク)のついているものは、AERT には無くウェブヘルパーで新たに付け加えた独自の点検項目です。

プログラム化できないものは検査方法欄で「未実装」としましたが、点検まで自動実行するものを「自動」、点検要素の検索を自動実行するものを「半自動」としてプログラム化しています。

真の判定には、WCAG1.0 および AERT ワーキング・ドラフトを参照し、「未実装」の点検と「半自動」の判定を各自で実施していただく必要があります。

フォルダー括簡易チェックでは、「検査方法」が自動となっている項目のみ点検可能です。

項目番号	優先度	検査方法	項目内容	HTML点検	C-HTML点検
ガイドライン1. 聞くための内容や見るための内容には、同等の役割を果たす代替のものを提供する					
1.1.1	1	自動	すべての画像(IMG)に代替テキスト(alt)が用意されているか		-
1.1.2	1	自動	画像が重要な情報を伝達し、その代替テキストがある場合は longdesc属性やDリンクが加えられているか		-
1.1.3	1	自動	imageタイプのINPUTで代替テキスト(alt)が正しく用意されているか		-
1.1.4	1	自動	アプレット(APPLET)に代替テキスト(alt)が正しく用意されているか		-
1.1.5	1	自動	画像表示目的のOBJECT要素に代替テキストが正しく使われているか		-
1.1.6	1	半自動	オーディオファイルには代替テキストがあるか		-
1.1.7	1	自動	埋め込みオーディオファイルにテキスト等価物が提供されているか		-
1.1.8	1	自動	FRAME中のlongdesc属性は正しく使われているか		-
1.1.9	1	自動	AREA中に代替テキスト(alt)があるか		-
1.1.10	1	自動	SCRIPTはNOSCRIPT等価とあわせ正しく使われているか		-
1.1.11	1	自動	a要素内のテキスト内容は適切か(13.1.1と同じ:AERT参照)		-
1.1.12	1	未実装	ASCIIアートに代替テキストが用意されているか	-	-
1.1.13(*)	2	自動	絵文字(携帯端末用コードでよく使われるもの)が使われていないか		-
1.1.14(*)	2	自動	機種依存文字が使われていないか		-
1.1.15(*)	1	半自動	画像(IMG)の代替テキスト(alt)が場所取りテキストになっていないか		-

1.2.1	1	半自動	サーバー側イメージマップの各動作領域にテキストのリンクを作っているかイメージマップのすべてのホットスポットには代替のテキストリンクが用意されているか		-
1.3.1	1	半自動	マルチメディア情報はオーディオ説明を持っているか		-
1.4.1	1	半自動	マルチメディア情報には同期した代替情報があるか		-
1.4.2	1	半自動	SMIL中ではsystem - captionフラグがあるか(1.4.1に含む)		-
1.5.1	3	自動	クライアント側イメージマップの各動作領域にテキストリンクを作っているか		-
ガイドライン2 . 色だけに依存しない					
2.1.1	1	半自動	色彩を伴うすべての情報は、色彩指定しなくても入手可能か		-
2.2.1	3	自動	テキスト等と背景は誰にでも見やすい組み合わせになっているか		-
ガイドライン3 . 正しくタグ付けし、適切にスタイルシートを使う					
3.1.1	2	半自動	適切なマークアップ言語があれば、マークアップを用いて情報を伝えているか		-
3.2.1	2	自動	公開されている正式な文法でドキュメントが作られているか		-
3.3.1	2	自動	レイアウトや体裁の制御にはスタイルシートを用いているか		-
3.4.1	2	自動	絶対画素数ではなく、相対サイズと相対座標を使っているか		-
3.5.1	2	自動	見出し項目は正しく入れ子化(nesting)されているか		-
3.5.2	2	自動	テキストは見出しに間違えられないか		-
3.5.3	2	自動	見出し要素(H1 ~ H6)に長いテキストを使っていないか		-
3.6.1	2	自動	リスト要素はリストのみに使われ、正しく入れ子化(nesting)されているか		-
3.7.1	2	半自動	引用はQ要素やBLOCKQUOTE要素でマークアップしているか		-
3.7.2	2	半自動	Q要素やBLOCKQUOTE要素は正しく使われているか		-
3.7.3	2	自動	BLOCKQUOTE要素をフォーマットに使用していないか		-
ガイドライン4 . 自然言語の取り扱い方に関する情報を明確に示す					
4.1.1	1	未実装	テキストの自然言語やその切り替わりを識別しているか	-	-
4.2.1	3	自動	略語が使われていないか		-
4.3.1	3	自動	文章の基本の(primary)自然言語を定義しているか		-
ガイドライン5 . うまく変換されるテーブルを作る					
5.1.1	1	半自動	表の目的を決定しているか(データ表か、レイアウト表かを自動点検の前に指定します)		-
5.1.2	1	自動	データ表では行と列の見出し語が付いているか		-
5.2.1	1	半自動	データ表ではscope、headers、axisでラベルをつけているか		-
5.3.1	2	半自動	レイアウト表はリニアライズされて意味があるか		-
5.4.1	2	自動	表をレイアウトのため使用する場合、見かけ上の書式化のために、構造化マークアップを使用していないか		-
5.5.1	3	自動	表中の行や列のデータを表示する場合、表の要約が用意されているか		-
5.5.2	2	自動	表にキャプションが用意されているか		-
5.6.1	3	自動	行や列の長い見出しラベルに対しての省略語が用意されているか		-
ガイドライン6 . 新しい技術を利用したページは、うまく変換されるようにしておく					
6.1.1	1	半自動	スタイルシートがなくても、ページは読むことが可能になっているか		-
6.2.1	1	半自動	FRAMEおよびIFRAME中で、正しいソースが使われているか		-
6.2.2	1	半自動	ダイナミックなコンテンツと連動してその代替がアップデートされる		-

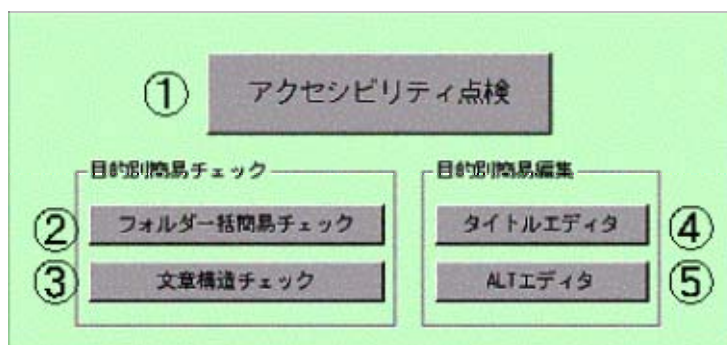
			か		
6.3.1	1	自動	重要な情報を伝えたり、機能したりするスクリプトの各々に代替のコンテンツを用意しているか(OBJECT、EMBED、APPLET)		-
6.4.1	2	半自動	プログラムのオブジェクトにおいては、入力装置に依存しないか		-
6.5.1	2	自動	各FRAMESETにはNOFRAMES要素が含まれているか		-
ガイドライン7. 時間とともに変化する内容については、ユーザーが制御できるようにする					
7.1.1	1	半自動	ページがフリッカーの原因になっていないか		-
7.2.1	1	自動	BLINK要素によりプリンキングが使われていないか(W3C削除推奨)		-
7.3.1	2	自動	MARQUEE要素によりスクロールが使われていないか		-
7.3.2	2	半自動	アプレットやスクリプトで動くコンテンツを使っていないか		-
7.4.1	2	自動	META要素に自動リフレッシュが入っていないか		-
7.5.1	2	自動	META要素に自動リダイレクトが入っていないか		-
ガイドライン8. ページ中に組み込まれたもののユーザーインターフェイスは、それ自体がアクセシブルなものにする					
8.1.1	2	半自動	スクリプトやアプレットを補助的技術を用いて可能な限りアクセシブルにしているか		-
ガイドライン9. 装置に依存しないように設計する					
9.1.1	1	自動	サーバー側イメージマップを使っていないか		-
9.2.1	2	半自動	固有のインターフェイスを持つ要素は、すべてデバイスに依存していないか		-
9.3.1	2	半自動	スクリプトは論理的イベント処理型になっているか		-
9.4.1	3	自動	論理的なタブ属性(tabindex)を、フォー - ム制御、オブジェクトから定義しているか		-
9.5.1	3	半自動	accesskey属性が使われているか		-
ガイドライン10. 暫定的な解決策をとる					
10.1.1	2	自動	AおよびAREAのtarget属性において、blankやnewが含まれていないか		-
10.1.2	2	半自動	スクリプト類が新しいウィンドウを開かないか		-
10.2.1	2	半自動	LABEL要素は正しく配置されているか		-
10.3.1	3	未実装	リニアライズされた表が用意されているか	-	-
10.4.1	3	自動	INPUT、TEXTAREA、SELECT要素でデフォルトの値があるか		-
10.5.1	3	自動	隣接するリンクは分離されているか		-
ガイドライン11. W3Cのテクノロジーとガイドラインを使用する					
11.1.1	2	未実装	可能な最新の技術仕様を使っているか	-	-
11.2.1	2	自動	いずれ廃棄される予定のW3C技術を使用していないか		-
11.3.1	-	未実装	利用者の設定に合ったドキュメントを提供しているか(解析方法が未解決のため未実装:AERT参照)	-	-
11.4.1	1	未実装	ページは必要なレベルのすべてのチェックポイントに合格したか	-	-
ガイドライン12. 前後関係や位置を表す情報を提供する					
12.1.1	1	自動	各フレームにタイトルがつけられているか		-
12.2.1	2	未実装	フレームタイトルだけでは明確でない場合、フレームの目的や、フレームの相互関係を説明する	-	-
12.3.1	2	半自動	8つ以上OPTION要素がある場合にOPTGROUP要素が使用されているか		-

12.3.2	2	半自動	FORM要素に8つ以上コントロールがある場合に、FIELDSET要素が使用されているか		-
12.3.3	2	半自動	リスト要素が正しく使用されているか		-
12.3.4	2	未実装	長い文が使われていないか	-	-
12.3.5	2	未実装	長い段落が使われていないか	-	-
12.3.6	2	未実装	長いテキストが使われていないか	-	-
12.4.1	2	自動	LABEL要素やINPUT要素は正しい属性をもっているか		-
ガイドライン13. はっきりとわかるナビゲーションのための仕組を提供する					
13.1.1	2	自動	各リンク部分の行き先は明確になっているか		
13.2.1	2	自動	ページにタイトルは付けられているか		-
13.2.2	2	未実装	RDFは正しいか	-	-
13.3.1	2	未実装	サイト上の全体的なレイアウトに関しての情報を用意する	-	-
13.4.1	2	未実装	一定のナビゲーション機能を用いる	-	-
13.5.1	3	未実装	ナビゲーション構造へ入りやすくするため、ナビゲーションバーを用意する	-	-
13.6.1	3	未実装	関連するリンクはグルーピング化されているか	-	-
13.7.1	3	未実装	いろいろな熟練度と設定条件での検索タイプを可能にしているか	-	-
13.8.1	3	未実装	見出し、パラグラフ、リスト等のはじめに、区分のための情報を用意する	-	-
13.9.1	-	未実装	文書が複数ページから成る場合は、それらを集めるための情報が提供されているか(解析方法が未解決のため未実装:AERT参照)	-	-
13.10.1	3	未実装	多行にわたるASCIIアートをスキップする手段があるか	-	-
ガイドライン14. 文書は明瞭で簡潔なものにする					
14.1.1	1	半自動	明瞭で正確な見出し化がされているか		-
14.1.2	1	未実装	文や段落のトピックは、文や段落の始めに位置しているか	-	-
14.1.3	1	未実装	それぞれの段落では1つの主要な考えが書かれているか	-	-
14.1.4	1	未実装	俗語や専門用語が使われていないか	-	-
14.1.5	1	未実装	共通語が使われているか	-	-
14.1.6	1	未実装	受動的な動詞よりも能動的な動詞が使われているか	-	-
14.1.7	1	未実装	サイトの内容にふさわしい言語が使われているか	-	-
14.1.8	1	未実装	よく使われる言葉が特殊な意味合いで使われていないか	-	-
14.1.9 (*)	2	自動	追加項目 同音異義語が含まれていないか		
14.1.10 (*)	2	自動	追加項目 難漢字(第2水準)が含まれていないか		
14.1.11 (*)	2	自動	追加項目 難しいカタカナ語が含まれていないか		
14.2.1	3	未実装	画像やオーディオ表現を用いてテキスト内容を補足し、ページの把握を容易にする	-	-
14.3.1	3	未実装	ページ間で一定の表現形式になっているか確認してください	-	-

compact-HTML は、HTML から不要なもの(携帯電話等の狭いディスプレイでは表示しきれないもの等)を省いて作られているため、対応する点検項目数も少なくなっています。

2.3 機能構成

ウェブヘルパー Ver2.0 には、以下の5つの機能があります。



ウェブヘルパー Ver2.0 のメイン機能

【アクセシビリティ点検】機能

HTML のアクセシビリティの点検を行うための、一番基本的な機能です。

点検対象となる HTML を1つ指定することで、指定した HTML がどれくらいアクセシビリティを考慮して作られているかを点検し、その結果を表示します。点検対象の HTML の指定では、インターネット上の URL かローカル（自分のパソコン内かローカルなネットワーク上）にあるファイルを指定し、点検することができます。

目的別簡易チェック機能

【フォルダ一括簡易チェック】機能

【文章構造チェック】機能

アクセシビリティ点検機能のような網羅的な点検はできませんが、ある特定の点検項目に関してのみ点検する場合に使うと便利な機能です。

フォルダ内の複数のファイルを一度に点検できる【フォルダ一括簡易チェック】機能と、見出し要素（タグ）について点検する【文書構造チェック】機能があります。

- 【フォルダ一括簡易チェック】機能

HTML のアクセシビリティの点検を行います。ここでの点検はローカルにあるフォルダを指定することで、そのフォルダ内にある複数の HTML ファイル（拡張子が「.HTML」または「.HTM」に限る）のアクセシビリティを一括して点検することができます（サブフォルダ以下は含まれません）。

ただし、点検項目は利用者の確認を必要しないものだけに限定されますので、あくまで簡易的な点検になります。特定の項目だけを点検したい場合に便利です。

- 【文章構造チェック】機能

ウェブヘルパーの点検項目 3.5.1 や 3.5.3 に関連して、ローカルにある HTML の文章構造（h1～h6 の見出しタグの構成）を確認することができる便利機能です。

見出しタグがきちんと入れ子^{*1}になっているか、簡潔で分かりやすい見出しが付けられているかなどを視覚的に点検することができます。



目的別簡易編集機能

【タイトルエディタ】機能

【ALT エディタ】機能

アクセシビリティ確保の観点から、HTML ファイルの編集に使うと便利な機能です。

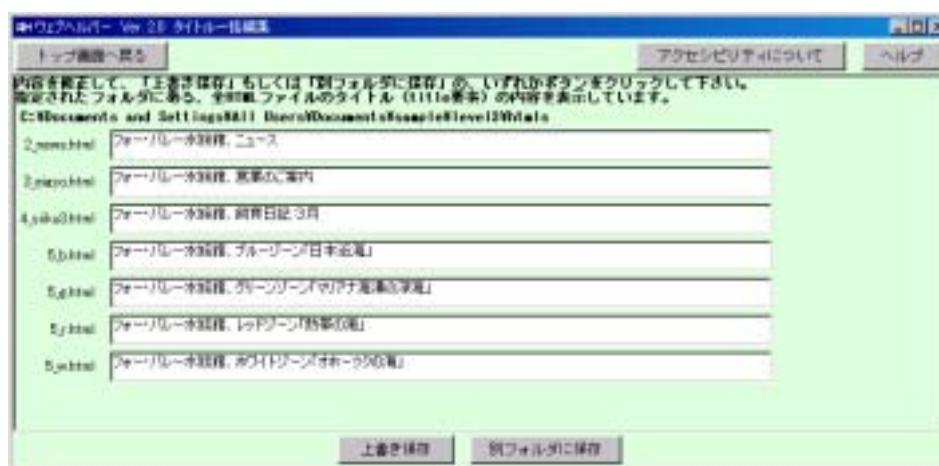
複数のファイルのタイトルが適切に付けられているかを確認しながら必要に応じて修正できる【タイトルエディタ】機能と、一つのファイル内の画像（IMG 要素）に付けられている ALT 属性を確認しながら必要に応じて修正できる【ALT エディタ】機能があります。

*1 入れ子：上位の見出しタグの中に順番に下位の見出しタグが位置づけられている状態。

- 【タイトルエディタ】機能

ウェブヘルパーの点検項目 13.2.1 に関連して、ローカルのフォルダ内にある複数の HTML ファイルに対して、そのページタイトル (title タグ) を一括して確認・修正するための便利機能です (サブフォルダ以下は含まれません)。

それぞれ分かりやすいページタイトルが付けられているか、同じタイトルのページが無いかなどを点検し、必要に応じて修正することができます。



- 【ALTエディタ】機能

ウェブヘルパーの点検項目 1.1.1 に関連して、ローカルにある HTML で使われている画像 (img タグ) の代替テキスト (ALT 属性^{*1}) を、一括して確認・修正するための便利機能です。

^{*1} ALT 属性： 音声読み上げソフト (音声ブラウザなど) では、HTML で img タグ (要素) 使って画像が貼り付けられていると、そこで指定されている ALT 属性を読み上げるため、音声読み上げソフトを使っている視覚障害者はどのような画像がそこに存在するかを理解することができます。

もし ALT 属性が指定されていないと、多くの音声ブラウザはリンクされているファイル名を読み上げてしまうため、視覚障害者はそこに存在する画像が何か理解できず不安になります。img タグを使う場合には、必ず ALT 属性を記述するようにします。

特に意味を持たない画像 (リストを示すポイント画像など) の場合でも、必ず ALT 属性を記述し、属性値に空欄や空白 ("" や " ")、"*" や "-" を指定しておくことで、音声ブラウザは逆に何も読み上げなくなります。

ALT 属性は、端的に分かりやすく記述すべきです。もし詳細な説明が必要となる場合には、別の手法を用いるべきです。詳しくは、後述する WCAG1.0 に解説されていますので、そちらを参照してください。

例： ``

使われている画像を参照しながら、その画像の代替テキストが適切であるかを点検し、必要に応じて修正することができます。



2.4 実行環境

ウェブヘルパー Ver2.0 は、以下の実行環境でお使いいただくことができます。

その他の利用環境での動作は確認できていませんが、寄せられた情報はホームページ「みんなのウェブ」(<http://www.jwas.gr.jp/>)に掲載していく予定です。

OS

- ・ Microsoft Windows 2000sp3 / XP
- ・ Mac OS X 10.2

ウェブブラウザ

- ・ Windows 版 Internet Explorer 6.x 以上推奨
- ・ Macintosh 版 Internet Explorer 5.2 以上推奨

修正用エディタ (テキストエディタ)

- ・ Windows 版 notepad.exe を推奨
- ・ Macintosh 版 TextEdit を推奨

Java 実行環境

- ・ 「Java™ 2 Runtime Environment (J2RE)」のバージョン 1.3.1 以上

本ソフトウェアは Java で開発されているため、Java の実行環境が必要です。詳細については、「3 ウェブヘルパーをインストールする」を参照してください。

モニター

- ・ 800 × 600 ピクセル以上

インストールに必要なハードディスク空き容量

- ・ 10 メガバイト以上 (Java 実行環境を除く)

注意!

ウェブヘルパー Ver 2.0R は、上記 Windows 環境でスクリーンリーダー (音声読上げソフト) をお使いの場合にご利用いただくことが出来ません。95Reader Version 4.5 (愛称: XP Reader) および PC-Talker XP で動作確認を行っています。2.0 と 2.0R では、機能および動作に関しての違いはありません。詳しくは、次ページを参照してください。

2.5 音声読上げによる利用

スクリーンリーダーをお使いの方は、ウェブヘルパー Ver.2.0R をご利用ください。

ここでは、特にスクリーンリーダーでご利用いただく際の注意点を記述していますが、他の操作方法は 2.0 と全く同じです。

(1) 2.0 と 2.0R の違いについて

ウェブヘルパー Ver.2.0R は、スクリーンリーダー（音声読上げソフト）に対応したバージョンです。

ウェブヘルパーは Java で開発していますが、Java に対応したスクリーンリーダーに限られるため、より多くのスクリーンリーダーでも読上げられるよう、画面表示に関する部分に変更が加えられています。機能および動作に関しては、2.0 と 2.0R に違いはありませんが、2.0R は読上げに対応した反面、表示の制御にいくつか制約が生じています。

スクリーンリーダーをお使いでない方は、2.0 をご利用ください。

(2) ウェブヘルパー 2.0R の動作環境について

ウェブヘルパー Ver.2.0R は、「95Reader Version 4.5（愛称：XP Reader）」（株式会社システムソリューションセンターとちぎ）および「PC-Talker XP」（株式会社高知システム開発）で動作確認をしています。

JAWS for Windows（IBM Version）Version 3.7 および Version 4.5（いずれも、日本アイ・ビー・エム株式会社）でも動作はしますが、環境設定内のウェブブラウザやエディタのオプションを指定するエディットボックスが、「参照のエディット」と読み上げられますので、ご注意ください。

なお、ウェブヘルパー Ver.1.0 では、Java Accessibility API（以下、JAAPI と呼ぶ）に対応していますので、JAAPI 対応のスクリーンリーダーを使用することにより、Java Access Bridge 経由で読上げに必要な情報を取得することができます。

JAAPI に対応している JAWS for Windows Version 4.5 をお使いの方は、日本アイ・ビー・エムのホームページから「ウェブヘルパー Ver1.0 用スクリプトの公開について」（http://www-6.ibm.com/jp/accessibility/soft/download_v2_wh.html#navskip）を参照し、ウェブヘルパー Ver1.0 をご利用ください。2.0 は 1.0 に比べ、操作性の面で向上を図っていますが、機能面ではほぼ同等のものです。ウェブヘルパー Ver1.0 は、ホームページ「みんなのウェブ」（<http://www.jwas.gr.jp>）からダウンロードしてください。

(3) スクリーンリーダーで使用する場合の注意点

共通項目

ウェブヘルパー Ver2.0R では、メッセージを表示するためのエディットボックスと、実際に文字列が入力できるエディットボックスがあります。メッセージが入っているエディットボックスには、文字の入力はできません。タブキーで項目間を移動する場合は、読み上げられる内容をよく確認してから、操作してください。

アクセシビリティ点検、フォルダー括簡易チェック、文章構造チェック、タイトルエディタ、ALT エディタ内の、最初に「点検対象」と読み上げられるエディットボックスは、メッセージ表示のためのものです。実際にファイルやフォルダ名を入力する場所は、タブキーを一度押して移動したエディットボックスやコンボボックスですので注意してください。

フォルダー括簡易チェックで、表示された点検結果のエラー項目を確認するには、リストボックスでエラーのあるファイルにカーソルを合わせ、エンターキーを押してください。

XP Reader (95 Reader Ver.4.5) をお使いの場合

XP Reader をお使いの場合は、XP Reader の設定から「全体」タブにある「クリップボード」を有効にして、お使いください。

PC-Talker XP をお使いの場合

PC-Talker XP をお使いの場合は、「画面の読み方」設定にある「クリップボードを自動的に読み上げる」を有効にして、お使いください。

トップ画面において項目間を移動する際、最初に「メッセージ」と読み上げられることがあります。その後に項目名が読み上げられますので、内容を確認してから、操作してください。

トップ画面において、タブキーで移動していると「目的別簡易チェックのメッセージ、エンターはフォルダー括簡易チェック、文章構造チェック」と読み上げられる場所があります。その場合、エンターキーを押すと「フォルダー括簡易チェック」が起動します。「目的別簡易編集」と読み上げられる場所も同様です。

項目間をタブキーで移動しただけでは、項目の内容を確認できない部分があります。「～の文字入力」と聞こえたら、上か下矢印を押して内容を確認してください。

2.6 その他

(1) 著作権など

- ・ 著作権について： ウェブヘルパー Ver2.0 の著作権は総務省が有しています。
本活用マニュアルの著作権は通信・放送機構が有しています。
- ・ 再配布について： ウェブヘルパー Ver2.0 のオリジナルプログラムの再配布は禁止します。詳しくは、「みんなのウェブ：著作権および再配布について」
(<http://www.jwas.gr.jp/helper/index.html>) を参照してください。

(2) 免責事項について

ウェブヘルパー Ver2.0 および動作に必要な Java の実行環境を、インストールまたは使用したことにより生じた、直接的または間接的な損失、損害等について、いかなる場合においても一切の責任を負わないものとします。

全て各利用者の責任において、本ソフトウェアのインストールまたは使用を行ってください。

(3) インストール時の補足注意事項

本ソフトウェアを勤務先のパソコンなどにインストールし、また利用のための環境設定を行うにあたり、セキュリティの設定状況によっては管理者権限を持った方でないと行えない場合があります。

このような場合は、システム管理者の方にご相談ください。

重要！

バージョンアップに従い、この活用マニュアルの内容を更新する場合があります。
最新版はホームページ「みんなのウェブ」(<http://www.jwas.gr.jp/>) で公開していきます。

3 ウェブヘルパーをインストールする

ウェブヘルパー Ver2.0 を利用するには、ウェブヘルパー Ver2.0 のプログラムをご自分のパソコンにインストールしてお使いください。

重要！

ウェブヘルパーVer2.0 をインストールする方法は、以下の二つの方法があります。

- 1) ホームページ「みんなのウェブ」からダウンロードしてインストールする。
- 2) CD-ROM 版を入手し、CD-ROM からインストールする。

プログラムに更新が発生した場合は、みんなのウェブでお知らせします。その場合最新版をインストールするには、ホームページよりダウンロードしてインストールを行ってください。

3.1 ウェブヘルパーのインストール

注意！

CD-ROM からウェブヘルパー Ver2.0 をインストールする場合は、CD-ROM を挿入すると自動的にブラウザが起動しインストールのためのガイダンスが表示されます。よく読んでから手順に沿ってインストール作業を行なってください。もし自動的に表示されない場合は、CD-ROM 内にある index.html ファイルをダブルクリックしてください。

ここでは、Windows 環境の場合を例に、ホームページ「みんなのウェブ」からダウンロードしてインストールする、一般的な方法を説明します。

OS やウェブヘルパーのバージョン等によって、インストールの方法は異なります。詳しくは、必ずホームページに記載の手順を参照してください。

- ・ ホームページ「みんなのウェブ」(<http://www.jwas.gr.jp>) にアクセスしてください。
- ・ 「ウェブヘルパーを利用する」のページから、「ウェブヘルパーのダウンロード」ページに移動してください。
- ・ 記載の説明に沿って、インストールするウェブヘルパーのプログラムを選択し、圧縮されたプログラムファイルをダウンロードしてください。

- ・ ダウンロードしたファイルをダブルクリックすると、自動的に解凍が開始されます。解凍先のフォルダを指定するダイアログが表示された、ウェブヘルパーのプログラムを保存するフォルダを指定して【OK】を選択してください。
- ・ 解凍が済むと、新しいフォルダが作成され、その中には [conf][html][img][lib] の4つのフォルダと、webhelper.jar という1つのファイルが解凍されます。(設定により拡張子は表示されない場合があります。また、一度でもウェブヘルパーを起動すると [temp][usr] という2つのフォルダが追加作成されます。)

3.2 Java 実行環境のインストール

ウェブヘルパーは Java で開発しているため、Java の実行環境が必要になります。

ご使用されるパソコンの Java 実行環境がバージョン 1.3.1 より古い場合、または Java 実行環境がインストールされていない場合、Java 実行環境をインストールしていただく必要があります。

注意！

CD-ROM からウェブヘルパー Ver2.0 をインストールした場合、Windows 環境では自動的に Java 実行環境もインストールされますので、この手順を行う必要はありません。次のステップに進んでください。

CD-ROM からウェブヘルパー Ver2.0 をインストールした場合でも、Macintosh 環境では現在の Java 実行環境を確認し、場合によっては新しい Java 実行環境をインストールしていただく必要があります。CD-ROM に入っている「お読みください (Readme.txt)」を参照してください。

ご使用パソコンにおける現在の「Java 実行環境の確認方法」および「Java 実行環境のインストール方法」は、ご使用の OS により異なります。

それぞれの環境における手順は、ホームページ「みんなのウェブ」の「ウェブヘルパーのダウンロード」ページに記載していますので、その内容を参照してください。

4 ウェブヘルパーをアンインストールする

ウェブヘルパー Ver2.0 をアンインストール(削除)する方法は、OS やインストール方法によって異なります。

注意！

Windows 環境で CD-ROM からウェブヘルパー Ver2.0 をインストールした場合は、コントロールパネルの「プログラムの追加と削除」から、【ウェブヘルパー2.0】の削除を実行してください。

この場合は、ウェブヘルパー Ver2.0 をインストールする際に一緒に自動的にインストールされた Java 実行環境も、同時にアンインストールされます。

ウェブヘルパー Ver2.0 を、ホームページ「みんなのウェブ」からダウンロードしてインストールした場合、ならびに Macintosh 環境の場合は、インストール時に作成されたウェブヘルパー Ver2.0 のプログラムの入っているフォルダごと削除してください(レジストリへの登録、DLL の更新等は行っていません)。

5 ウェブヘルパーの起動と環境設定

以上でウェブヘルパー Ver2.0 を利用していただく準備が整いました。

ただし、初めてウェブヘルパー Ver2.0 を起動した時に、まず「環境設定」をしていただく必要があります。以下の手順を参考に、環境設定を済ませてからウェブヘルパー Ver2.0 をお使いください。

5.1 ウェブヘルパーの起動

ウェブヘルパー Ver2.0 の起動方法は、インストール方法によって異なります。

注意！

Windows 環境で CD-ROM からウェブヘルパー Ver2.0 をインストールした場合は、デスクトップ上に自動作成された【ウェブヘルパー-2.0】のアイコンをダブルクリックしてください。あるいは、スタートメニューからプログラム ウェブヘルパー-2.0 を選択し、ウェブヘルパーを起動してください。

ウェブヘルパー Ver2.0 を、ホームページ「みんなのウェブ」からダウンロードしてインストールした場合、ならびに Macintosh 環境の場合は、インストール時に作成されたウェブヘルパー Ver2.0 のプログラムが入っているフォルダ内にある「webhelper.jar」もしくは「webhelper」（拡張子を表示しない設定になっている場合）というファイルを実行（ダブルクリック）してください。ウェブヘルパー Ver2.0 が起動します。

必要に応じて、上記実行ファイルのショートカットを作成し、そこから起動していただいても結構です。

5.2 環境設定をする

ウェブヘルパー Ver2.0 を初めて起動した時に、まず「環境設定」をしていただく必要があります。また、環境設定の変更が必要になった場合も、ここを参照して環境設定の変更を行ってください。

注意！

ウェブヘルパー Ver2.0 を初めて起動した時に表示されるトップ画面では、【環境設定】のボタン以外は操作できません。
初めて起動した時は、まず環境設定を行ってください。



画面解説メッセージ表示機能の設定（必須項目）

ウェブヘルパー Ver2.0 は、この設定を「有効」にしておくことで、画面が切り替わる毎にその画面を説明するポップアップメッセージを表示します。

特にウェブヘルパー Ver2.0R をスクリーンリーダーでお使いの場合には、この設定を「有効」にしてください。ポップアップメッセージを読上げることで、音声読み上げでの操作に役立つ情報が得られるようになります。

このメッセージが不要な場合は、「無効」を選択することで画面解説メッセージは表示されなくなります。

ウェブブラウザの設定（必須項目）

ウェブヘルパー Ver2.0 は、点検結果に対する「詳細内容の表示」や「エラー該当箇所の表示」などに、ウェブブラウザを使用します。お使いのウェブブラウザを起動するための実行ファイルを、フルパスで設定して下さい。

修正用エディタの設定（必須項目）

ウェブヘルパー Ver2.0 は、点検を行なった HTML をその場で修正する場合にテキストエディタを使用します。お使いのテキストエディタを起動するための実行ファイルを、フルパスで設定して下さい。

プロキシサーバの設定（任意項目）

インターネット接続時にプロキシサーバを使用している場合のみ、「プロキシサーバホスト名」および「ポート番号」を設定して下さい。プロキシサーバを使用していない場合や、ウェブ上のコンテンツ（HTML）を点検しない場合には設定は不要です。

分からない場合には、ネットワーク管理者にご相談ください。

重要！

Macintosh 環境でこのプロキシサーバの設定を行った場合、ウェブヘルパーからウェブサイトにアクセスできなくなる現象が発生する場合があります。
Macintosh 環境でプロキシサーバを使用している場合は、次ページの手順にしたがって設定してください。

設定の終了

以上の設定が完了したら、【設定終了ボタン】を選択実行(クリック)してください。

【Macintosh 環境でのプロキシサーバの設定方法について】

この内容は、ウェブヘルパー Ver2.0 を Macintosh 環境でお使いになり、インターネット接続時にプロキシサーバを使用する方のみお読みください。

- 1) 環境設定画面にて、「ウェブブラウザ」および「修正用エディタ」を設定する。
- 2) 同画面にて、ご使用になっている「プロキシサーバホスト名」および「ポート番号」を設定する。
- 3) 「テスト用接続先 URL」入力欄に普段ご覧になるウェブサイトの URL を入力する。
- 4) テストボタンを押下して、ウェブサイトへアクセスできるかを確認してください。
- 5) ブラウザが起動し、入力した URL のページが表示されれば問題ありません。
- 6) ブラウザは起動するが、入力した URL のページが表示されない場合には、「プロキシサーバホスト名」および「ポート番号」を削除し、再度「テストボタン」を押下してテストを行なって下さい(これで接続されるはずです)。
- 7) 接続を確認後、【設定終了ボタン】を押下して環境設定を保存終了してください。