

ウェブヘルパー Ver1.0

利用マニュアル



2003.8.1 版

<http://www.jwas.gr.jp>

目次

第1章 ウェブヘルパーを使うための準備 ____ 1

1	はじめに	2
1.1	ウェブアクセシビリティとは?	2
1.2	障害者・高齢者がウェブを利用するときの問題	2
2	ウェブヘルパーについて	6
2.1	ウェブヘルパーの機能構成	6
2.2	ウェブヘルパーの点検のしくみ	6
2.3	ウェブヘルパーの点検項目について	12
2.4	ウェブヘルパーの実行環境について	16
2.5	その他	17
3	ウェブヘルパーをインストールする	18
3.1	ウェブヘルパーのインストール【Windows】【Macintosh】共	18
3.2	Java 実行環境のインストール【Windows 環境の場合】	21
3.3	Java 実行環境のインストール【Macintosh 環境の場合】	23
4	ウェブヘルパーをアンインストールする	25
5	ウェブヘルパーを起動する	26
5.1	ウェブヘルパーの起動	26
5.2	環境設定をする	27

第2章 ウェブヘルパーを使うための準備 ____ 31

6	【アクセシビリティ点検】全体の手順	33
7	アクセシビリティ点検を開始するまで	34
8	アクセシビリティ点検結果を確認する	40

第3章 その他の機能を使う _____ 47

9	【フォルダ一括簡易チェック】機能の利用手順	48
10	【文章構造チェック】機能の利用手順	55
11	【タイトルエディタ】機能の利用手順	57
12	【ALTエディタ】機能の利用手順	60

-
- * Sun、Sun Microsystems、Java およびすべての Java 関連の商標は、米国およびその他の国における米国 Sun Microsystems, Inc.の商標または登録商標です。
 - * Microsoft、Windows およびすべての Windows 関連の商標は、米国 Microsoft Corporation の米国およびその他の国における登録商標です。
 - * Apple、Macintosh は、米国 Apple Computer, Inc.の商標です。
 - * その他、本マニュアルの各ページに記載されている会社名、システム名、製品名などは、一般に 各社の商標または登録商標です。
 - * なお、各ページ内では (TM)、(R) および (C) マークは省略しています。
-

第1章 ウェブヘルパーを使うための準備

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。

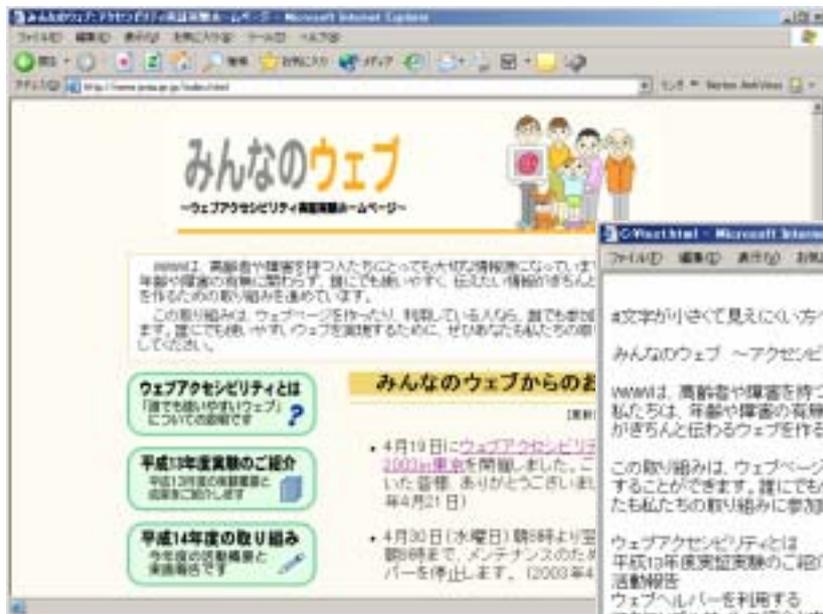
全盲の人の場合

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。

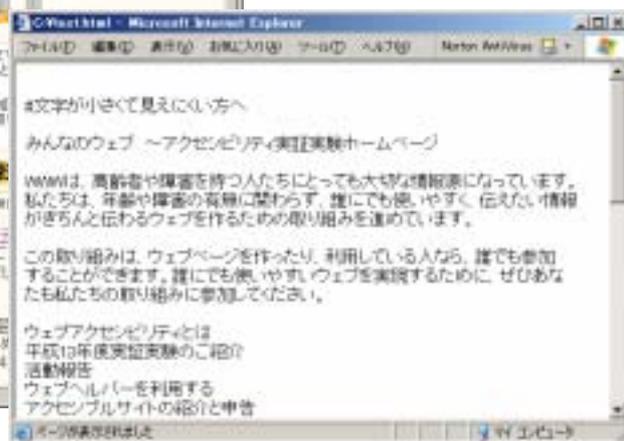
テキストで書いた情報であっても、テーブルでレイアウトする際に段組が適切でないと、情報の意味する順序と読み上げの順序とに矛盾が生じてしまい、理解が難しくなります。

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。全盲の利用者は、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。

また、読み上げソフトではテキスト情報を単線的に読み上げるため、データ表など二次元の配置のある情報は、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。



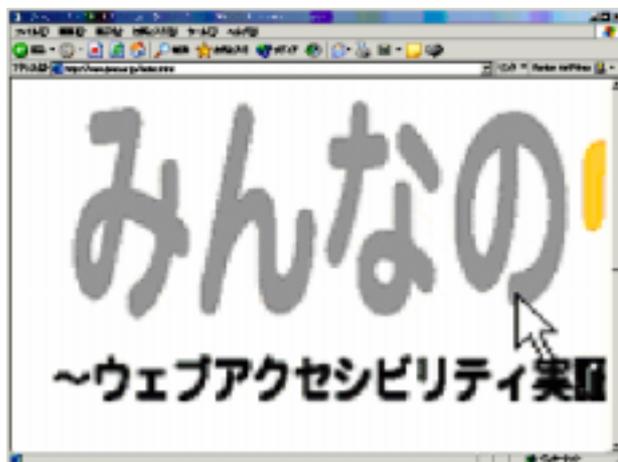
[一般的なブラウザでの表示例]



[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。

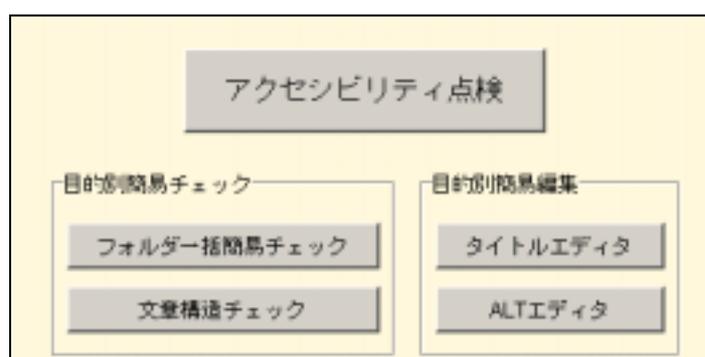
高齢者に起こりやすい問題としては、たくさんのサイトを調べているうちに現在の位置や履歴を見失ったり、ボタン名が英語になっていると操作を敬遠するという問題があります。また、直前に見ていた画面に戻る操作がページによって異なるため、戸惑う現象も多く見られます。適切な情報の整理、次が想像できるボタン名、わかりやすい操作手順への配慮がなされていないページは、高齢者にとってはバリアの高いページとなります。

2 ウェブヘルパーについて

2.1 ウェブヘルパーの機能構成

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易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。

ウェブヘルパー Ver1.0 には、以下の5つの機能があります。



【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。

【フォルダ一括簡易チェック】機能

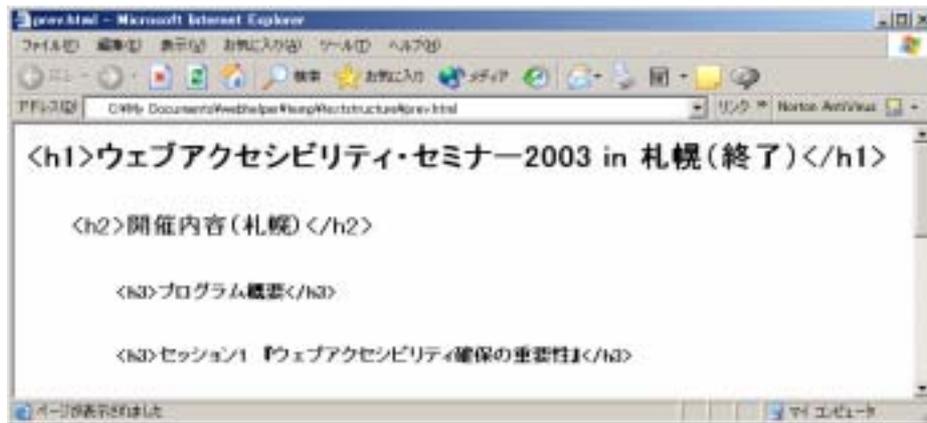
HTML のアクセシビリティの点検を行いますが、ここでの点検はローカルにあるフォルダを指定することで、そのフォルダ内にある複数の HTML ファイル（拡張子が「.HTML」または「.HTM」に限る）のアクセシビリティを一括して点検することができます（サブフォルダ以下は含まれません）。

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

ローカルにある HTML の文章構造 (h1 ~ h6 の見出しタグの構成) を確認するための機能です。

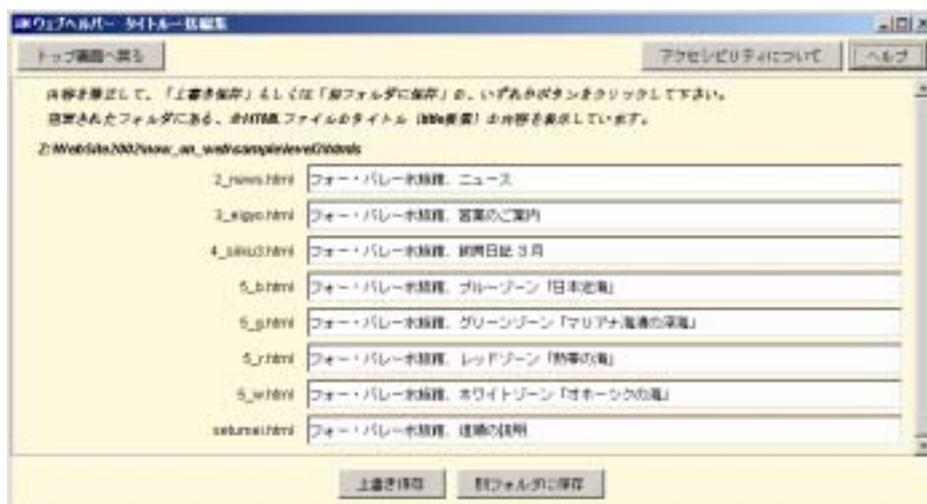
見出しタグがきちんと入れ子^{*1}になっているか、簡潔で分かりやすい見出しが付けられているかなどの点検を行うことができます。



【タイトルエディタ】機能

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。



^{*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。

【ALTエディタ】機能

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・修正するための機能です。

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。



^{*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： ``

2.2 ウェブヘルパーの点検のしくみ

ウェブヘルパーの点検機能は、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などを考慮して開発されています。

HTMLコンテンツのアクセシビリティを確保するため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 の点検レベルと優先度について】

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

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。

- ウェブヘルパーの点検機能は AERT を参考にしていますので、アクセシビリティを確保する視点は WCAG1.0 が拠りどころとなります。
- しかし、WCAG1.0 の 14 項目のガイドラインにしたがって、それを完全かつ自動的にシステムが点検することは不可能です。ウェブヘルパーが行う点検は、WCAG1.0 のガイドラインが意図するところの一部分を点検しているに過ぎません。かつ、利用者の判断に委ねられている点検項目も多数存在します。
- したがって、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。

2.3 ウェブヘルパーの点検項目について

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、それ自身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みを提供する					
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。

2.4 ウェブヘルパーの実行環境について

ウェブヘルパー Ver1.0 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 98SE / 2000sp2 / XP
- ・ Mac OS X 10.1 以上

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、「3.3 JAVA 実行環境のインストールについて」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)

音声読み上げにより操作するために必要なソフトウェア

ウェブヘルパー Ver1.0 は Java Accessibility API (以下、JAAPI と呼ぶ) に対応していますので、Java Access Bridge 経由で読み上げに必要な情報を取得することができます。このため、JAAPI 対応 (Java Access Bridge 対応) のスクリーンリーダーを使用することにより、画面の読み上げが可能となります。現在、JAWS for Windows Version 4.5 (日本アイ・ビー・エム社製) が近日中に JAAPI 対応予定との情報を得ています。

なお、次期 Ver2.0 では、より多くのスクリーンリーダーに対応する予定です。

2.5 その他

(1) 著作権など

- ・著作権について : ウェブヘルパー Ver1.0 の著作権は総務省が有しています。
- ・再配布について : ウェブヘルパー Ver1.0 は、再配布不可とします。

(2) 免責事項について

ウェブヘルパー Ver1.0 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

本ソフトウェアをご利用になるパソコンに、Java 実行環境がインストールされていない場合、Sun Microsystems 社 (Windows の場合) あるいは apple 社 (Macintosh の場合) のホームページより「ダウンロード」および「インストール」を行なっていただく必要があります (詳しいダウンロードおよびインストールの手順は次節を参照してください)。

ただし、勤務先のパソコンなどの場合、「ダウンロード」や「インストール」など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、この利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 Ver1.0 を利用する場合には、以下の手順に沿ってウェブヘルパー Ver1.0 をご自分のパソコンにインストールしてください。

なお、以下の手順をご覧いただき、必要な場合には Java の実行環境もインストールしていただく必要があります (Java の実行環境が整っていないと、ウェブヘルパーをお使いいただくことはできません)。

3.1 ウェブヘルパーのインストール【Windows】【Macintosh】共

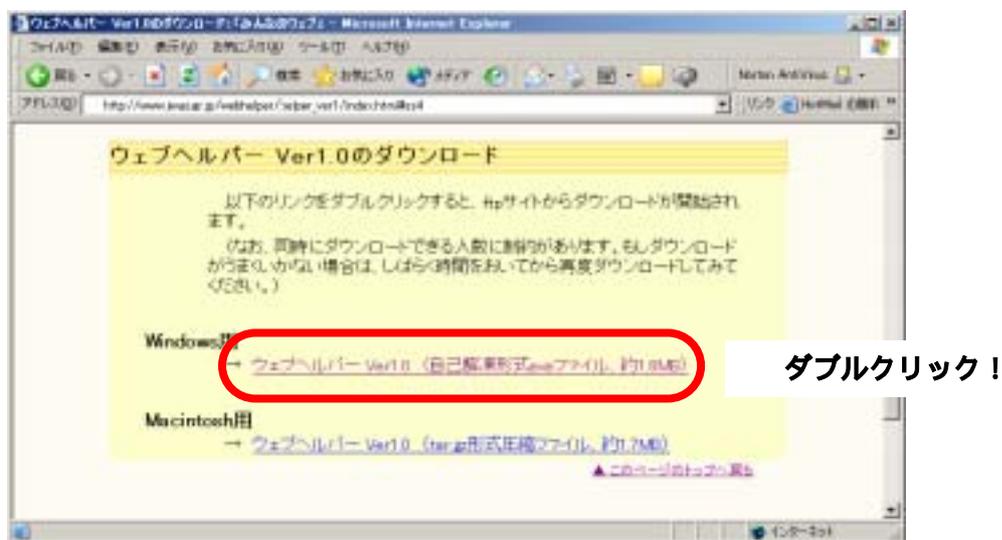
ウェブヘルパー Ver1.0 のインストールは、自己解凍形式に圧縮されたプログラムファイルをダウンロードし、それをパソコン内の任意の場所へ解凍 (ダウンロードファイルをダブルクリックするか、ファイルを選択した状態で Enter キーを押す) していただくだけです。

解凍すると「webhelper」というディレクトリと、そのディレクトリ内に各種ファイルが展開されます。

以下、Windows 環境の場合を例に手順を説明します。

ウェブヘルパー Ver1.0 (プログラムの圧縮ファイル) のダウンロード

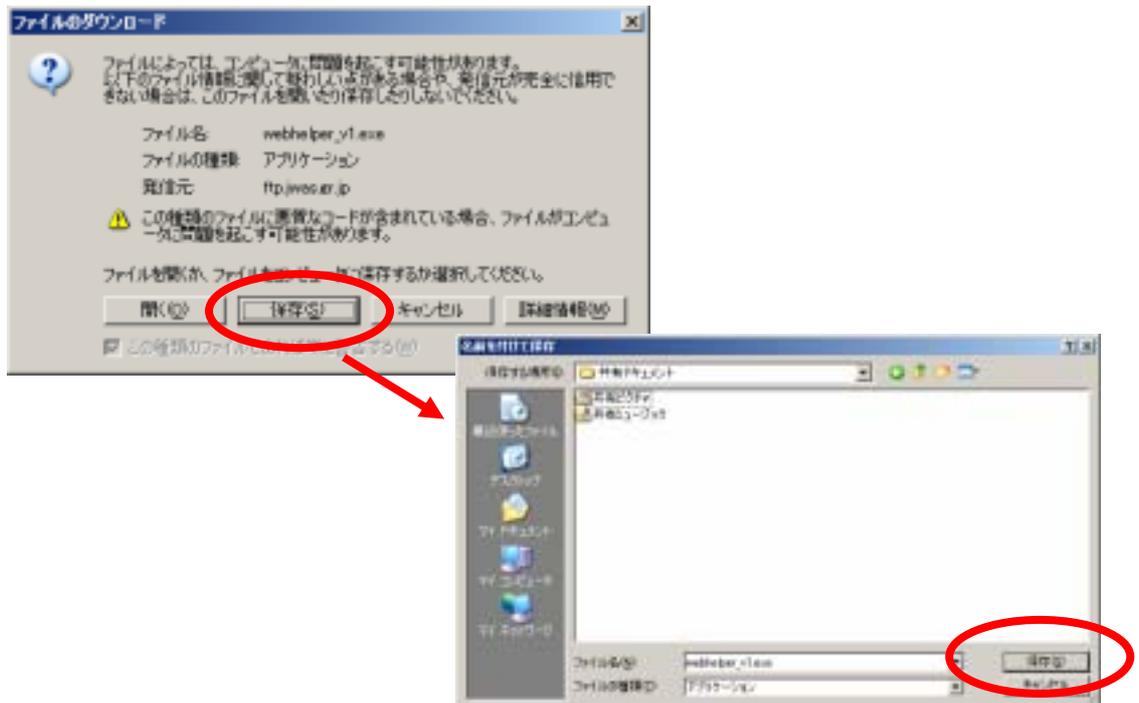
ホームページ「みんなのウェブ」(http://www.jwas.gr.jp/) からリンクをたどり、ウェブヘルパー Ver1.0 をダウンロードします。



ダウンロードファイルの保存

「ファイルのダウンロード」のダイアログが表示されたら、【保存(S)]】を選択してください。

「名前をつけて保存」のダイアログが表示されたら、ファイルを保存する先を指定して、再度【保存(S)]】を実行してください。

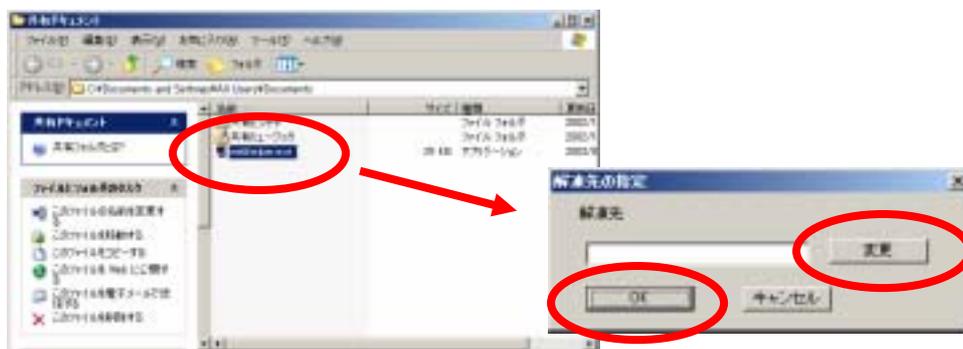


ダウンロードファイルの解凍

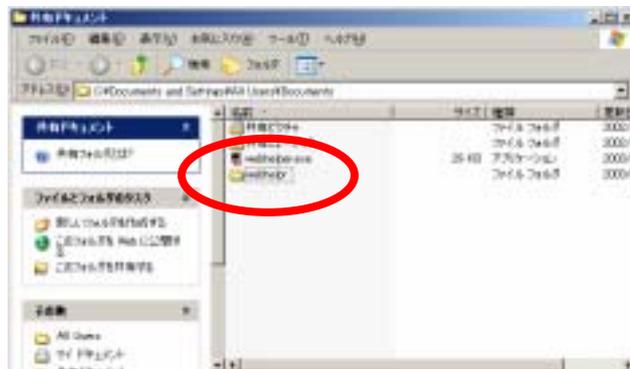
マイドキュメントやエクスプローラーなどで、上記ダウンロードファイルを保存したフォルダを開いてください。

ダウンロードしたファイル「webhelper_v1.exe」をダブルクリック（あるいはファイルを選択して Enter キー）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ら、解凍後のファイルを保存するフォルダを指定して【OK】を選択してください。



解凍が済みましたら、解凍先として指定したフォルダ内に「webhelper」というフォルダが作成されたことを確認してください。



webhelper フォルダ内には、以下のフォルダ（4つ）とファイル（1つ）が解凍されます。

- 1) [フォルダ]  conf
- 2) [フォルダ]  html
- 3) [フォルダ]  img
- 4) [フォルダ]  lib
- 5) [ファイル] webhelper.jar

以上で、ウェブヘルパー Ver1.0 のインストールは完了です。

引き続き、以下の手順をご覧いただき、必要な場合には Java 実行環境のインストールを実施してください。

注意！

ウェブヘルパー Ver1.0 を一度でも起動すると、webherler フォルダ内には、以下のフォルダが自動作成され追加されます。

- [フォルダ]  temp
- [フォルダ]  usr

3.2 Java 実行環境のインストール【Windows 環境の場合】

ウェブヘルパー Ver1.0 は、Java 実行環境 1.3.1 以上を推奨しています。

ご使用の Java 実行環境が Java 1.2 以下の場合、または Java 実行環境がインストールされていない場合、Java 1.3.1 のインストールを推奨しています。

まず、ご使用のパソコンにおける現在の Java 実行環境の確認を行ってください。

(1) ご使用のパソコンにおける Java 実行環境の確認

Java の実行環境が既にインストールされている場合には、コマンドプロンプトで Java のバージョンを確認することができます。

Windows 98 では「スタート」「プログラム」「MS-DOS プロンプト」、Windows 2000 / XP では「スタート」「プログラム」「アクセサリ」「コマンドプロンプト」を選択し、コマンドプロンプトを起動してください。

例えば「C:¥Documents and Settings¥user > 」のようにコマンドの入力が要求されますので、“>”に続いて“java△-version”(△はスペース、入力文字は半角)と入力し、Enter キーを押してください。

Java 実行環境が既にインストールされている場合には、「java version “1.3.1”」といったメッセージが表示されます。

(例)

```
C:¥Documents and Settings¥user> java-version
Java version "1.3.1"
Java(TM) 2 runtime Environment, Standard Edition (build 1.3.1-b24)
Java HotSpot(TM) Client VM (build 1.3.1-b24. mixe mode)

C:¥Documents and Settings¥user>
```

もし、Java 実行環境が以下の(ア)あるいは(イ)のような場合は、次頁のインストール方法を参照して、Java 実行環境の 1.3.1 をインストールしてください。

(ア) Java 実行環境がインストールされていない場合

(“ コマンドまたはファイル名が違います ” とか “ コマンドが認識されません ”
といった旨のメッセージが表示された場合)

(イ) Java 実行環境のバージョンが 1.3.1 よりも古い (1.3.0 など) 場合

(Java 実行環境が 1.3.1 以上の場合は、インストールせずそのままお使い下さい)

(2) Java 実行環境のインストール方法

重要！^{*1}

上記 のように他のバージョンの Java 実行環境がインストールされている場合、そのバージョンをアンインストールしてからインストールをおこなってください。
詳しいインストールガイドは、<http://java.sun.com/j2se/1.4/ja/jre/install-windows.html> を参照してください。

Sun Microsystems 社のホームページにある「Java 2 Platform, Standard Edition v 1.3 のページ」(<http://java.sun.com/j2se/1.3/ja/index.html>) へアクセスしてください。

このページにある【J2SETM v1.3.1_08 のダウンロード】(表示位置は右上方)をクリックし、J2SE v 1.3.1_08 のダウンロードページに移動してください。

1.3.1 の後ろのアンダーバー以降の数字 (_08) は、逐次更新により番号が増えている場合があります。

J2SE v 1.3.1_08 のダウンロードリストの中から、「Windows(各国語)」の「JRE」の【ダウンロード】をクリックします (JRE と SDK の 2 種類がありますが、JRE の方をダウンロードしてください)。

ここから英語のページに移動します。

ライセンス同意の確認文書(英語)が表示されたら、ページ下方にある【ACCEPT】(同意) ボタンをクリックしてください。

新しい「J2SE V1.3.1_08」のダウンロードページが表示されますので、「Download j2re-1_3_1_08-windows-i586-i.exe」をクリックして、インストーラをダウンロードしてください。

ダウンロード終了後、インストーラを実行 (ダウンロードしたファイルをダブルクリック) してインストールを行なって下さい。

^{*1} アンインストールは、OS が Windows の場合にはコントロールパネルから「アプリケーションの追加と削除」ユーティリティを使用して下さい。

Sun Microsystems では、このように他のバージョンの Java 実行環境をアンインストールすることをバージョン 1.3.x までは特に指定していませんでしたが、バージョン 1.4.0 からこのような条項が追加されています。

詳しくは、「<http://java.sun.com/j2se/1.4/ja/jre/install-windows.html>」をご参照下さい

3.3 Java 実行環境のインストール【Macintosh 環境の場合】

ウェブヘルパー Ver1.0 は、Java 実行環境 1.3.1 以上を推奨しています。

ご使用の Java 実行環境が Java 1.3.1 以下の場合、または Java 実行環境がインストールされていない場合、Java 1.3.1 のインストールを推奨しています。

まず、ご使用のパソコンにおける現在の Java 実行環境の確認を行ってください。

(1) ご使用のパソコンにおける Java 実行環境の確認

ご自身のパソコンの Java 実行環境をご確認ください。

Macintosh では Java 実行環境が OS 標準でインストールされており、JAVA WEB START のヘルプなどから確認することができます。

もし、Java の実行環境がバージョン 1.3.1 よりも古い (1.3.0 など) 場合は、次のインストール方法を参照して、Java 実行環境を 1.3.1 以上にバージョンアップして下さい。

(Java 実行環境が 1.3.1 以上の場合は、インストールせずそのままお使い下さい)

重要！

Macintosh 版の Java 実行環境をバージョンアップする場合、対応する OS (Mac OS X) のバージョンアップが必要な場合があります。ダウンロードページの情報をよくご確認の上、Java 実行環境のバージョンアップをおこなってください。

古い Java 実行環境をアンインストールする必要はありません。

ご使用中の Java アプリケーションは、新しいバージョンの Java 実行環境で動く事となります。

(2) Java 実行環境のインストール方法

Apple Computer 社のホームページ (<http://www.apple.co.jp/>) にアクセスしてください。

上段にあるメニューバーの中から、「サポート」をクリックします。

サブ・メニューバーの中から、「ダウンロード」をクリックし、ソフトウェアアップデートのページに進んでください。

表示位置では右上方に表示されるメニューリストの中から、「用途別リスト」をクリックしてください。

- 21 Mac OS X、Java カテゴリ内にある、「Java 1.3.1 アップデート 1」をクリックして、Java 1.3.1 アップデートのページに進んでください。

(http://www.apple.co.jp/ftp-info/reference/java_1.3.1_update_1.html)

- 22 以下、画面の指示にしたがって、「Java1.3.1Update1.dmg」をダウンロードしてインストール作業を行ってください。

以上で、Windows 環境および Macintosh 環境でのウェブヘルパー Ver1.0 のインストールは終了です。

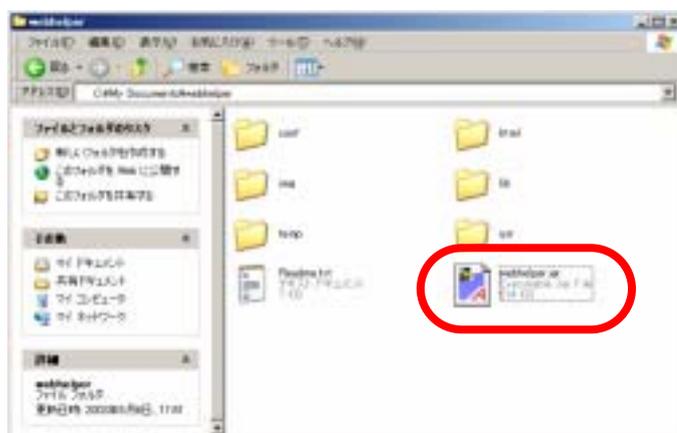
5 ウェブヘルパーの起動と環境設定

以上でウェブヘルパー Ver1.0 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 Ver1.0 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 Ver1.0 をお使いください。

5.1 ウェブヘルパーの起動

「3.1 ウェブヘルパーのインストール」で解凍し作成された webhelper 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（ダブルクリックするか選択した状態で Enter キーを押す）すると、ウェブヘルパー Ver1.0 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。

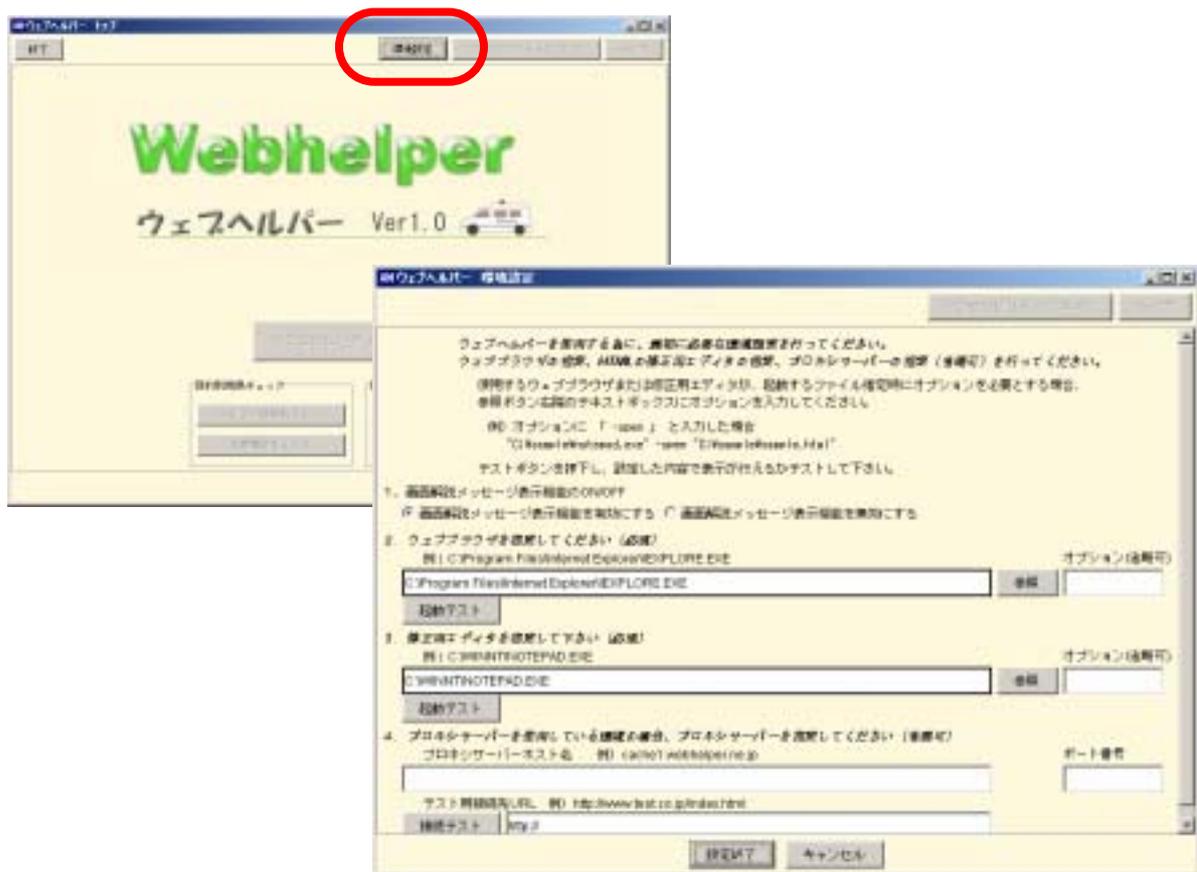
5 - 2 環境設定をする

ウェブヘルパー Ver1.0 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 Ver1.0 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。



23 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 Ver1.0 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。

スクリーンリーダーをお使いの場合には、この設定を「有効」にしてください。これにより、音声読み上げでウェブヘルパー Ver1.0 を操作していただくことが可能になります。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。

24 ウェブブラウザの設定（必須項目）

ウェブヘルパー Ver1.0 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。

25 修正用エディタの設定（必須項目）

ウェブヘルパー Ver1.0 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。

26 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。

27 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 Ver1.0 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。

重要！

ウェブヘルパー Ver1.0 は Java Accessibility API (以下、JAAPI と呼ぶ) に対応していますので、Java Access Bridge 経由で読み上げに必要な情報を取得することができます。

このため、JAAPI 対応 (Java Access Bridge 対応) のスクリーンリーダーを使用することにより、画面の読み上げが可能となります。平成 15 年 5 月 19 日現在、JAWS for Windows Version 4.5 (日本アイ・ビー・エム社製) の JAAPI 対応が確認できています。

第2章 アクセシビリティ点検を行う

ウェブヘルパー Ver1.0 には、以下の5つの機能があります。

ウェブヘルパー Ver1.0 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。

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。

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。

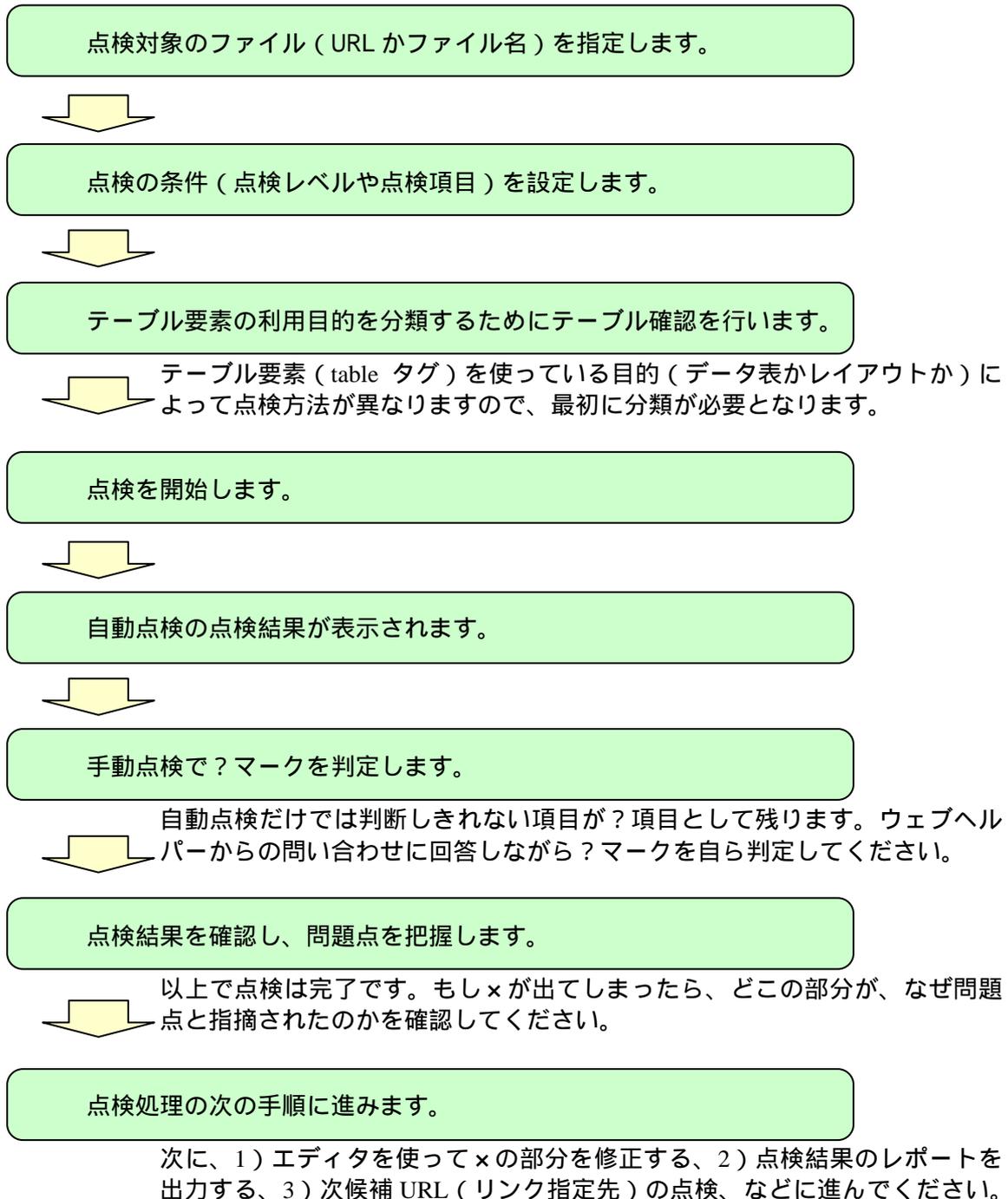
複数のファイルのタイトルが適切に付けられているかを確認しながら必要に応じて修正できる【タイトルエディタ】機能と、一つのファイル内の画像(IMG 要素)に付けられている ALT 属性を確認しながら必要に応じて修正できる【ALT エディタ】機能があります。

以下では、それぞれの機能の利用方法について順番に説明します。

まず最初に、一番基本的な機能である【アクセシビリティ点検】機能を使って、HTML コンテンツのアクセシビリティを点検してみましょう。

6 【アクセシビリティ点検】全体の手順

ウェブヘルパー Ver1.0 を使った HTML コンテンツの「アクセシビリティ点検」は、以下のような手順で行います。1つのファイル（1ページ）毎にこの手順を繰り返します。



7 アクセシビリティ点検を開始するまで

点検対象のファイル（URL かファイル名）を指定します。

アクセシビリティを点検したい HTML ファイルを指定します。

ウェブヘルパー Ver1.0 では、インターネット上の HTML コンテンツのほか、自分のパソコンのハードディスク内などローカルネットワーク環境でアクセスできる HTML ファイルも点検することができるようになりました。

点検対象を指定する枠内に、URL（<http://www.jwas.gr.jp/index.html> 等）か、もしくはファイル名（C:/WINDOWS/My Documents/index.html など）を入力してください。ファイル名を指定する場合には、【参照】ボタンを使ってファイルを探し出すこともできます。



重要！

フレームを使っている場合は、フレームセットを定義した HTML ページとは別に、各フレームを構成する全てのページを個別に点検してください。

注意！

ウェブヘルパー Ver1.0 は、HTML と compact-HTML に対応しています。
XML は、一部の点検項目を除くと点検することは可能です。
携帯電話用は i-mode 向けのみで、ez-web や J-sky など向けコンテンツには対応していません。
URL が https から始まるページの点検は行えない場合があります。
cgi で動的に生成されるページを点検することはできません。

点検対象のファイルを指定したら、次にそのファイルの HTML 形式を選択してください。
HTML か compact-HTML (i-mode 向けコンテンツの場合) を選ぶことができます。
通常のホームページを点検する場合には、HTML を選んでください。

点検の条件 (点検レベルや点検項目) を設定します。

点検レベル	<input type="radio"/> A	〈満たさなければならないレベル〉
	<input type="radio"/> ダブルA	〈満たすべきレベル〉
	<input checked="" type="radio"/> トリプルA	〈満たすべき事を望まれるレベル〉
	<input type="radio"/> 個別指定	〈点検項目を個別に指定可能〉

点検の条件として、点検レベルを選ぶか、点検項目を個別に指定します。

点検レベルは、シングルAからトリプルA (AAA) までの3段階に分かれています。この点検レベルは、ウェブヘルパーが点検の仕組みの拠りどころにしている WCAG1.0 の優先度に準じています。目標とする点検レベルを3段階の中から選んでください。

【参考】WCAG1.0 の点検レベルと優先度について

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

点検レベルを選択する以外に、個別に点検項目を指定して点検することもできます。例えば、「全ての画像（img 要素）に代替テキスト（ALT 属性）が指定されているかだけを確認したい」といった場合に便利です。

点検レベルが選択されている場合、【個別指定】ボタンは使えない状態になっていますが、個別指定の をチェックすると【個別指定】ボタンを押すことができるようになり、それをクリックすると点検項目を選択する次のページに移動します。



ここでは、点検したい項目を個別にチェックします。一つだけでなく、複数の点検項目を選ぶことができます。

上に並んだ【テーブル】【画像】【マルチメディア】【文書構造】をそれぞれ選択（クリック）することで、それぞれのテーマに関連した点検項目をまとめて選択することができます。複数のテーマを併せて選択することも可能です。

【全選択】をクリックすると、全ての点検項目が選択されます（トリプルAの点検と同じ）。【全削除】をクリックすると、既に選択した点検項目がクリアされます。

設定が終わりましたら、最下段にある【設定終了】をクリックしてください。もし何も選択せず、前の画面に戻りたい場合は、【キャンセル】をクリックしてください。

テーブル要素の利用目的を分類するためにテーブル確認を行います。

テーブル関連の点検をおこなう場合（点検レベル A、ダブル A、トリプル A、および個別指定でテーブル関連の点検項目が選択されている場合）には、点検を開始する前に、テーブル要素（table タグ）を使っている目的が「データ表」か「レイアウト」かによって点検方法が異なるため、分類しておく必要があります。



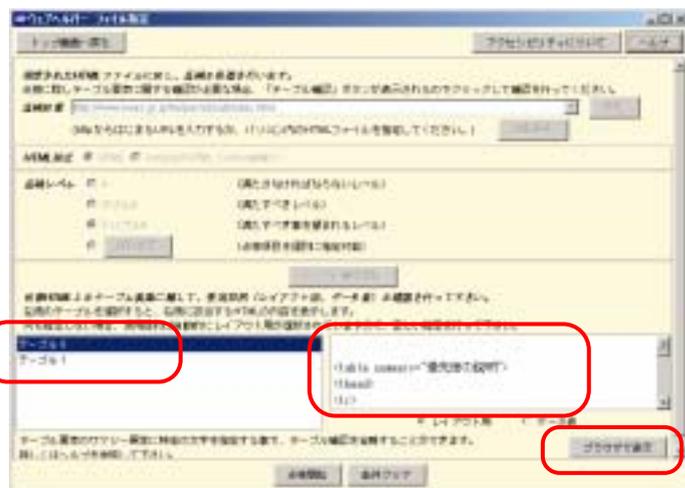
「点検対象」「HTML 形式」「点検レベル」の 3 項目の入力と選択が済んだら、【テーブル確認開始】ボタンをクリックしてください。

注意！

点検レベルを個別指定して、テーブル関連の点検項目を選択しなかった場合は、【テーブル確認開始】ボタンは選択できません。点検開始に進んでください。

テーブルの点検が終了すると、テーブル要素（table タグ）が使われていた場合、テーブル要素ごとに一覧が左側の枠内にリストで表示されます。もしテーブル要素が使われていなければ、何も表示されません。

左枠内にテーブル確認用の一覧が表示された場合、表示されているテーブル番号を選択すると、右側に



は該当テーブルの HTML ソースが表示されます。また HTML ソースを見ただけではどのテーブルが分からない場合は、【ブラウザで表示】ボタンをクリックするとブラウザが起動し、該当箇所のテーブルに[?]マークが付いて表示されます。

左側の枠内に表示されたテーブル要素の全リストを順番に選択しながら、そのテーブル要素の利用目的が「レイアウト用」か「データ表」かを指定してください。

重要！

最初は、全てレイアウト用に指定されます。

もし、データ表が一つも使われていない場合（全部がレイアウト用の場合）は、そのまま何も設定せず、先に進むことができます。データ表がある場合には、必ずどれがデータ表かを探し出して、それをデータ表に指定し直してください。

【参考】テーブル確認の省略方法について

データ表を使っているコンテンツを点検する場合には、このテーブル確認の作業がとてみたいへんになります。そこで、テーブル要素のサマリー属性に以下のような工夫をしていただくことで、この確認作業を簡便にすますことができます。

1) テーブルをデータ表として指定する場合

テーブル要素 (table タグ) のサマリー属性 (summary) に、「データ」(全角カタカナ) か「data」(大文字・小文字どちらでも可) の文字列を入れるようにしてください。そうすると、ウェブヘルパー Ver1.0 はその表がデータ表だと判断します。

例： <table summary="アクセス回数のデータ表です">

2) テーブルをレイアウト用として指定する場合

テーブル要素のサマリー属性に、「レイアウト」(全角カタカナ) か「layout」(大文字・小文字どちらでも可) の文字列を入れるようにしてください。そうすると、ウェブヘルパー Ver1.0 はその表がレイアウト用だと判断します。

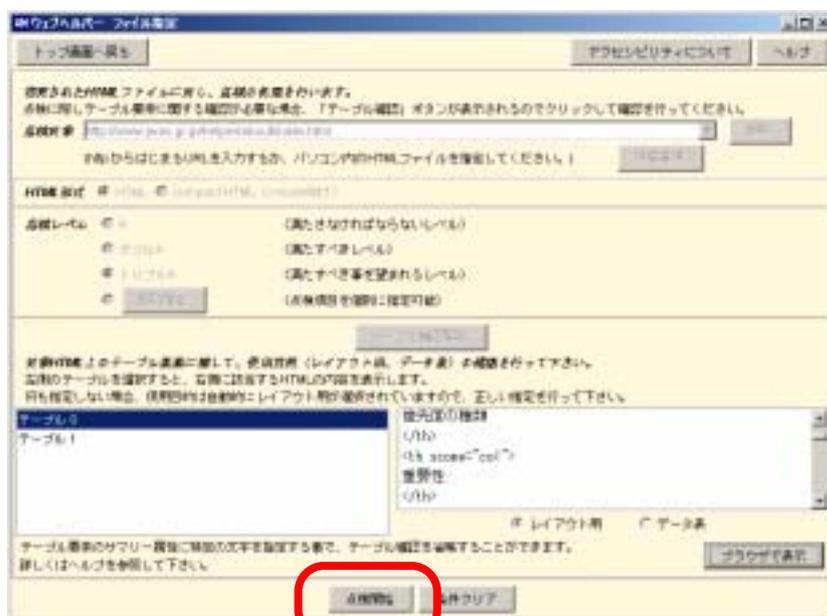
例： <table summary="この表はレイアウトを整える表です">

ただし、サマリー属性にデータ表だと判断する文字列（「データ」「data」）とレイアウト用だと判断する文字列（「レイアウト」「layout」）が混在している場合は、自動的に判断できず確認の問い合わせが発生します。

点検を開始します。

以上で、点検の準備が完了です。

最下段に表示された【点検開始】ボタンをクリックしてください。ウェブヘルパー Ver1.0
が指定された HTML コンテンツのアクセシビリティを点検します。



↑ 点検開始！

8 アクセシビリティ点検結果を確認する

自動点検の点検結果が表示されます。

点検開始を実行すると、ウェブヘルパー Ver1.0 は自動的に HTML のソースを解析し、WCAG1.0 に沿ってアクセシビリティを点検した後、その結果をレポートします。

重要！

ウェブヘルパーは、全てのアクセシビリティ点検を自動的に行うことはできません。最初の点検結果は自動点検のみの結果です。必ず手動点検が必要です。

✕マークは、自動点検で発見された問題点を指摘しています。

?マークは、自動点検では疑わしいとしか判断できず、別途手動点検が必要な箇所を指摘しています。

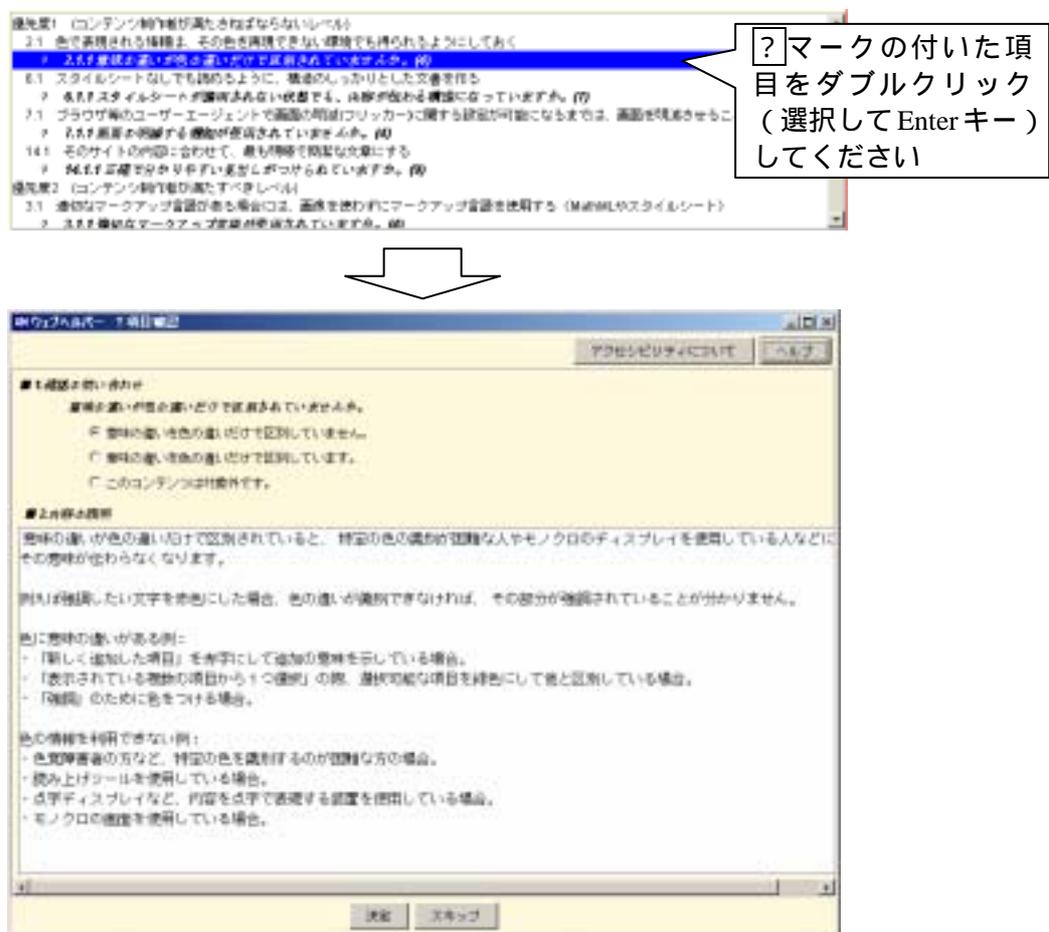
同じ点検項目でも複数個の問題箇所が見つかった場合には、「項目・箇所の問題点」といった形でレポートされます。



手動点検で「?」マークを判定します。

自動点検結果のレポートで「?」マークが表示された項目全てについて、ウェブヘルパー Ver1.0 の問い合わせに回答しながら、手動点検を実施します。

「?」マークが付いた項目をダブルクリックすると(選択して Enter キーを押す)、「?項目確認画面」が表示されます。



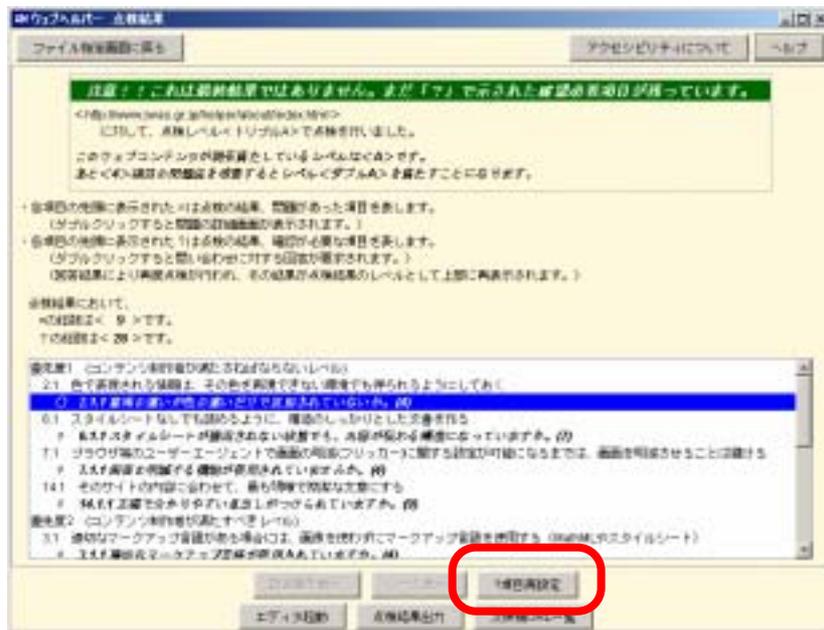
「内容の説明」をよく読み、問題には繋がらないか、あるいは問題に繋がる可能性があるかどうかをご自身で判定して、当てはまる選択肢にチェックを入れてください。

ウェブヘルパー Ver1.0 の自動点検では、疑わしいと判定された箇所に「?」マークが付きますので、問題無しか問題ありという二者択一の答えだけでなく、そもそも点検対象外という場合もあり得ます。その場合は「対象外」を選択してください。

選択肢にチェックしたら【決定】ボタンをクリックしてください。もしここで判定しない場合は、【スキップ】ボタンをクリックしてください。なお、決定した選択肢も後から変更することは可能です。

この手動点検で問題ありと判断した？は、マークに変わります。反対に問題なしと判断した？はマークに変わります。

もし先の手動点検の判断を変更したい場合は、再度その点検項目を選択した上で、下段に並んだ実行ボタンの中にある【?項目再設定】ボタンをクリックしてください。



ウェブヘルパーが指摘している問題点の場所が分からない場合には、以下の二つの手段で確認することができます。

確認したい項目を選択した上で、以下のいずれかのボタンをクリックしてください。



a. 表示画面上で該当箇所を確認



b. ソースで該当箇所を確認



点検結果を確認し、問題点を把握します。

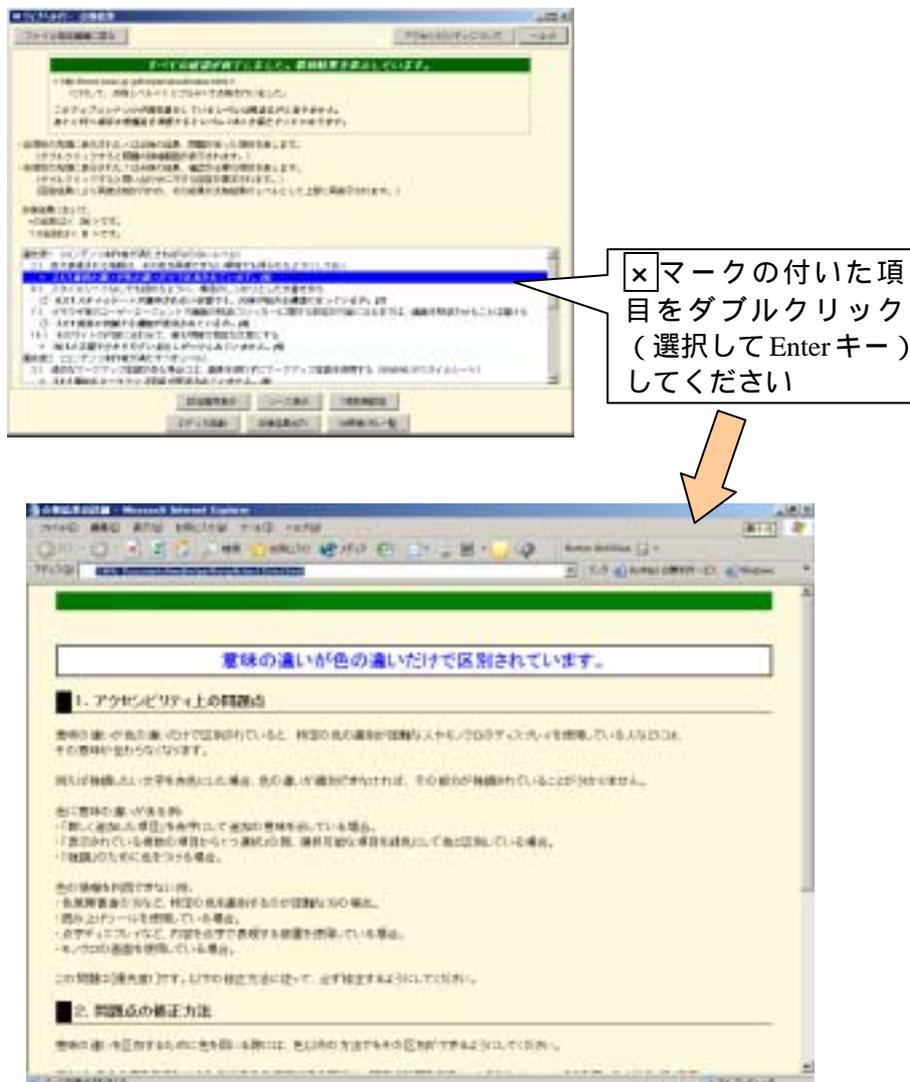
以上の自動点検と手動点検で、ウェブヘルパー Ver1.0 を使ったアクセシビリティ点検はひとまず完了です。

点検結果を確認しながら、問題点を把握しましょう。

(ア) 問題点の内容や修正方法についての把握

☒マークの付いた項目をダブルクリック（選択して Enter キーを押す）してください。

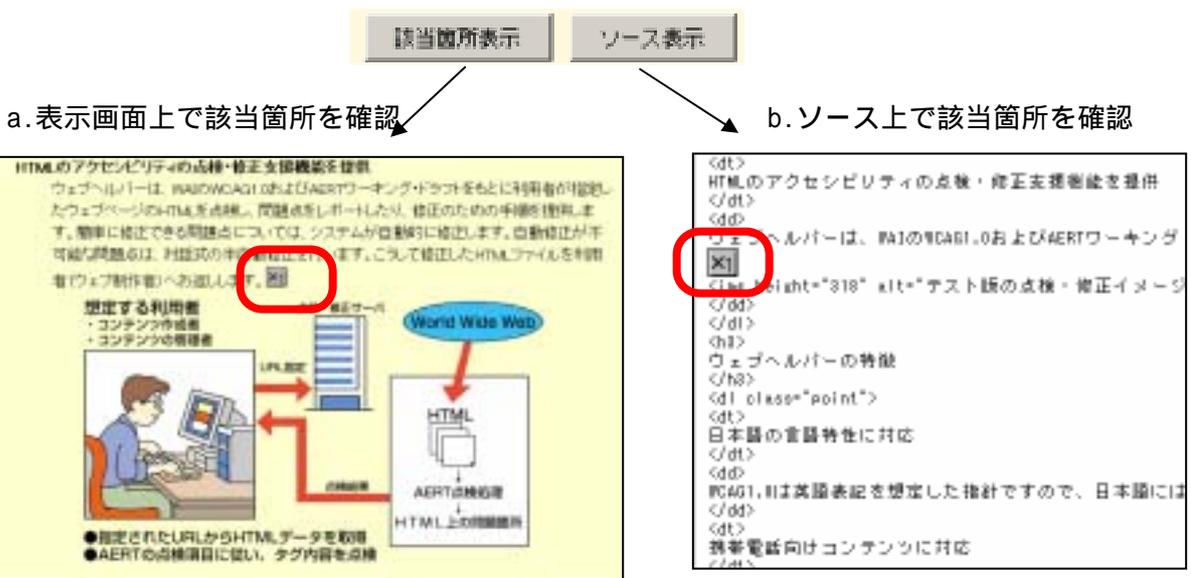
ブラウザが立ち上がり、その項目についての「1 アクセシビリティ上の問題点」と「2 問題点の修正方法」についての解説を読むことができます。



(イ) 問題箇所の把握

問題点として指摘されている箇所は、手動点検の時と同様に、表示画面上で該当箇所に **x** マークを表示させて確認するか、ソース上で該当箇所に **x** マークを表示させて確認することができます。

確認したい項目を選択した上で、以下のいずれかのボタンをクリックしてください。



注意！

HTML 全体にわたる問題点や外部 CSS に関する問題点などの場合は、表示画面上やソース上でその位置を特定することができませんので、一番先頭に **x** マークを並べて表示します。

x マーク内に表示されている数字は、優先度の数字です。

- x**1 … 優先度 1 の問題点
- x**2 … 優先度 2 の問題点
- x**3 … 優先度 3 の問題点

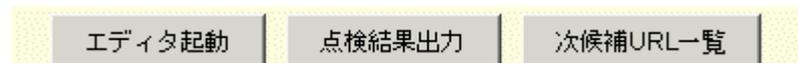
点検処理の次の手順に進みます。

点検結果を確認し、問題点が把握できたら、ウェブヘルパー Ver1.0 を使ったアクセシビリティ点検の作業は一段落です。しかし、問題点をそのままにしておいたら点検をした意味がありません。

次に、問題点を修正する作業に取り掛かりましょう。

(ア) 修正作業

ウェブヘルパー Ver1.0 の点検結果表示画面の最下段に、【エディタ起動】【点検結果出力】【次候補 URL 一覧】の3つのボタンが並んでいます。



【エディタ起動】を実行すると、環境設定で指定したテキストエディタのプログラムが起動し、点検を実施した HTML ソースを直接修正する作業に移ることができます。

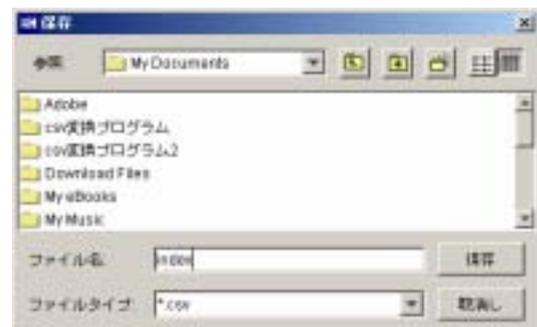
重要！

修正を行ったら、再度ウェブヘルパーを使ってアクセシビリティ点検を実施し、目標のレベルが達成されているかどうかを確認しましょう。

(イ) 点検結果の出力

【点検結果出力】を実行すると、ウェブヘルパー Ver1.0 を使ったアクセシビリティ点検結果を、レポートとして CSV 形式のファイルに出力することができます。

【点検結果出力】ボタンをクリック（選択して Enter キーを押す）すると、ファイルの保存先を指定するダイアログが表示されるので、「保存先のフォルダ」と「保存するファイル名」を指定して、保存を実行してください。



点検結果出力ファイル (CSV 形式)

(ウ) 次候補URLの点検

修正が済んで、必要に応じて点検結果レポートの出力を終えたら、次の HTML コンテンツの点検に移りましょう。

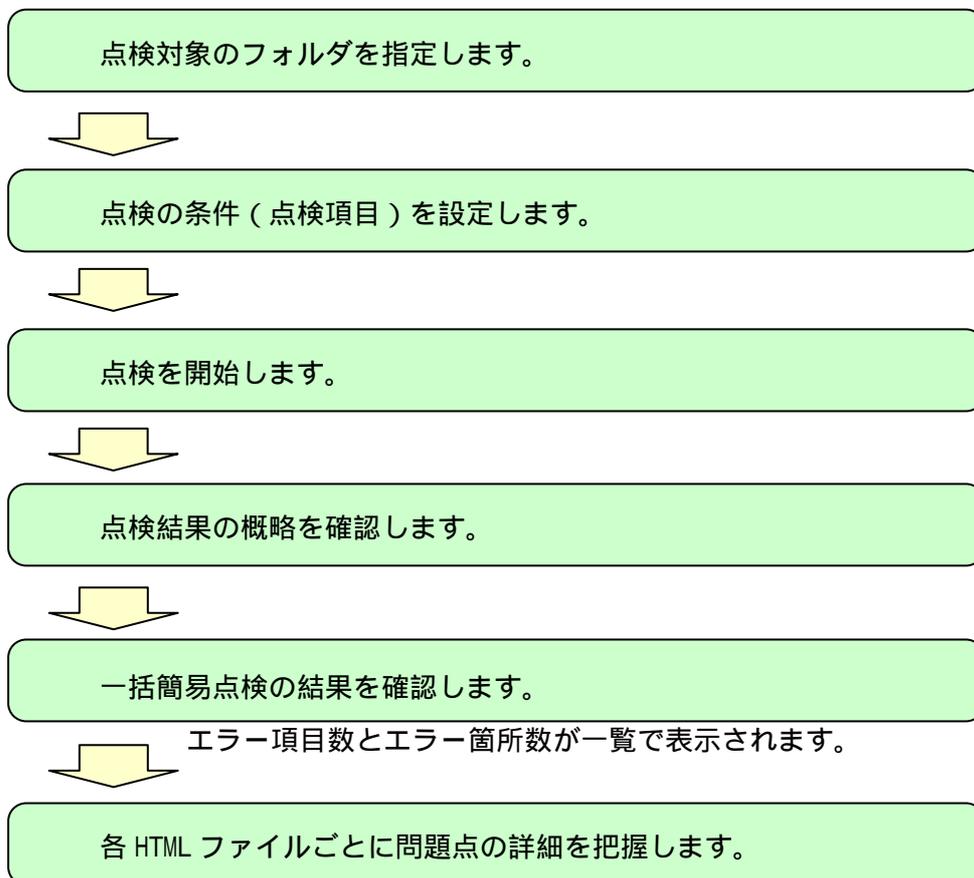
【次候補 URL 一覧】ボタンをクリックすると、先に点検した HTML コンテンツの中でリンク先が指定されていた URL やファイル名の一覧が表示されます。その中から次に点検したい HTML コンテンツを選んで、引き続き点検作業を行うことができます。



第 3 章 その他の機能を使う

9 【フォルダ一括簡易チェック】機能の利用手順

フォルダ内の複数ファイルを一度に簡易点検する手順は、以下のとおりです。



以下に、各手順の詳細を説明します。

点検対象のフォルダを指定します。

アクセシビリティを点検したいフォルダを指定します。

指定したフォルダ内にある HTML ファイルを一括点検することができます(サブフォルダ内のファイルは対象になりません)。

点検対象を指定する枠内にフォルダ名 (C:/My Documents/www/sample/ など)を入力してください。フォルダ名を指定する場合には、【参照】ボタンを使ってフォルダを探し出すこともできます。



重要！

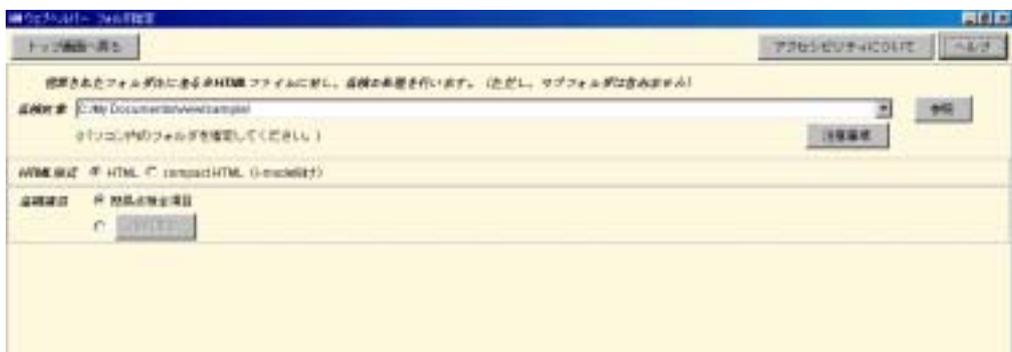
この機能での点検対象はローカルネットワーク環境でアクセスできるフォルダ内の HTML ファイルです。

- インターネット上に公開されているコンテンツは、一括点検することは出来ません。
- 指定したフォルダ直下の HTML ファイルが対象になります。サブフォルダ内のファイルは対象になりません。

点検対象のフォルダを指定したら、次に HTML 形式を選択してください。HTML か compact-HTML (i-mode 向けコンテンツの場合) を選ぶことができます。

通常のホームページを点検する場合には、HTML を選んでください。

点検の条件（点検項目）を設定します。



自動的に点検できる全項目を対象にする場合は、【簡易点検全項目】を選択します。

個別に点検項目を指定して点検する場合は、【個別指定】ボタンをクリックし、点検項目を選択する次のページに移動します。【簡易点検全項目】が選択されている場合、【個別指定】ボタンは使えない状態になっていますが、個別指定の をチェックすると【個別指定】ボタンを押すことができるようになり、それをクリックすると点検項目を選択する次のページに移動します。



ここでは、点検したい項目を個別にチェックします。一つだけでなく、複数の点検項目を選ぶことができます。チェックボックスが利用可能になっている項目が簡易点検の対象項目です。項目名がグレーで薄く表示されていてチェックボックスが選択できない状態に

なっている項目は、簡易点検の対象外です。自動的には点検が行えない項目ですので、「アクセシビリティ点検」機能を使って個別に手動点検する必要があります。

上に並んだ【テーブル】【画像】【マルチメディア】【文書構造】をそれぞれ選択（クリック）することで、それぞれのテーマに関連した点検項目をまとめて選択することができます。複数のテーマを併せて選択することも可能です。

【全選択】をクリックすると、全ての点検項目が選択されます（トリプルAの点検と同じ）。【全削除】をクリックすると、既に選択した点検項目がクリアされます。

設定が終わりましたら、最下段にある【設定終了】をクリックしてください。もし何も選択せず、前の画面に戻りたい場合は、【キャンセル】をクリックしてください。

重要！

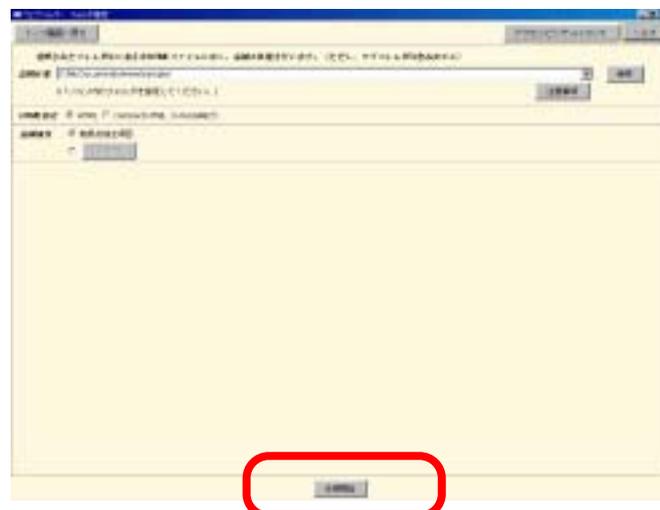
点検項目が限られます。

- 自動点検可能な項目についてのみ一括で点検する簡易機能です
- 全ての項目を対象に点検をおこなう場合は、「アクセシビリティ点検」機能を利用してください。

点検を開始します。

以上で、点検の準備が完了です。

最下段に表示された【点検開始】ボタンをクリックしてください。ウェブヘルパー Ver1.0 が指定されたフォルダ内の HTML コンテンツのアクセシビリティを点検します。



自動点検の点検結果が表示されます。

点検処理が終了すると、画面が切り替わり「フォルダ毎エラー数表示」画面が表示されます。



点検結果の概略を確認します。

「フォルダ毎エラー数表示」画面では、フォルダ内の HTML ファイルについて、簡易点検の結果問題のあった点検項目数と、HTML 内の問題箇所の数を確認することが出来ます。各ファイルのアクセシビリティの状況を大まかに把握してください。

一覧では、“HTML ファイル名 (YY 項目 / ZZ 箇所)” という形式で表示されます。このうち“YY 項目”とは、HTML ファイル内でアクセシビリティ上問題があった点検項目の数、ZZ 箇所とは、HTML 内の問題箇所の総数です。

例えば、

13.1.1 リンクを示すテキストが明確になっていません

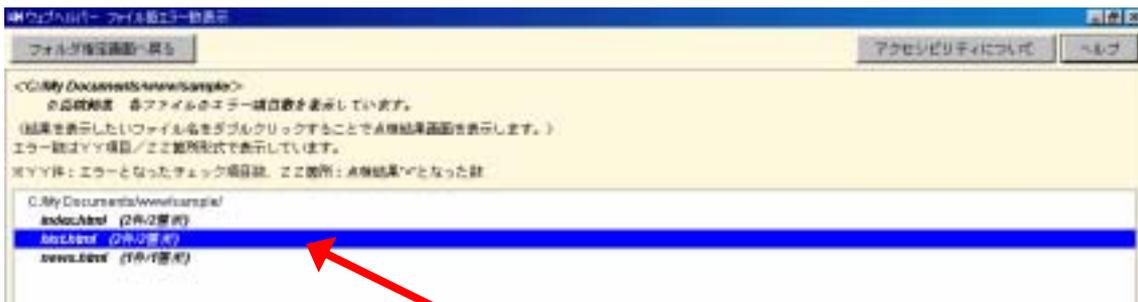
10.5.1 隣り合うリンクが明確になっていません

の2つの点検項目に問題があり、それぞれの項目について HTML 内に2箇所ずつ（合計4箇所）問題があった場合は、“HTML ファイル名 (2項目 / 4箇所)” という表示になります。

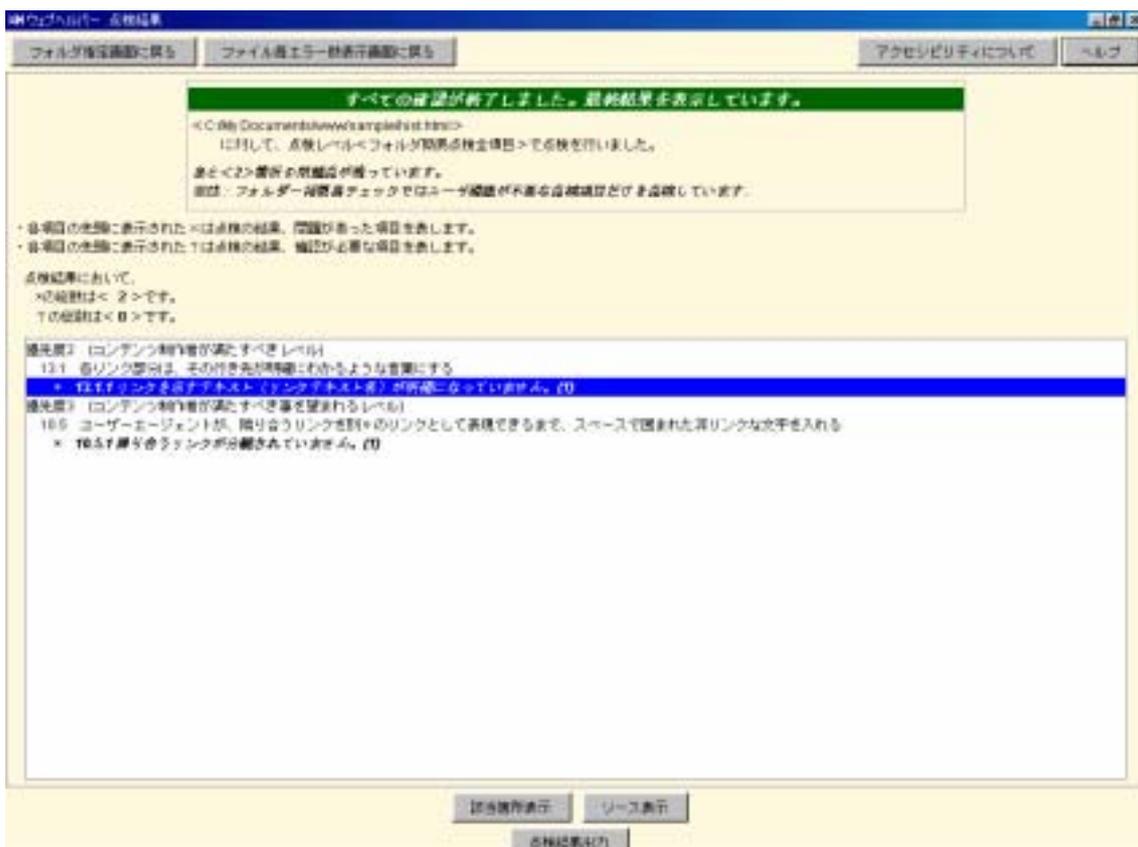
各 HTML ファイルごとに問題点の詳細を把握します。

さらに HTML ファイル毎に問題点の内容を詳細に確認することも出来ます。

一覧から確認したい HTML ファイル名を選択してください。「点検結果」画面に表示が切り替わります。



詳細を確認したい HTML ファイル名をクリック



「点検結果」画面では、選択した HTML ファイルの点検結果をレポートしています。この画面の確認方法、利用方法の詳細は 2 章 40 ページを参照してください。

点検結果の概略を示した「フォルダー毎エラー数表示」画面に戻るには、【フォルダー毎エラー数表示に戻る】ボタンを選択してください。ほかの HTML ファイルの点検結果の詳細を確認したい場合には、このボタンを選択し、「フォルダー毎エラー数表示」画面の一覧から確認したい HTML ファイル名を選択し直してください。

フォルダ名を指定する「フォルダ内一括簡易チェック」機能の先頭のページへ戻るには、【フォルダ指定画面に戻る】ボタンを選択してください。

注意！

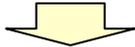
点検結果のレポートは、限られた点検項目を対象にしたものです

- 正確なアクセシビリティ点検を行うには、「アクセシビリティ点検」機能を利用して点検を行ってください。

10 【文章構造チェック】機能の利用手順

HTML の文章構造（h1～h6 の見出しタグの構成）を確認は、以下の手順で行います。

点検対象の HTML ファイルを選択します。



文章構造を確認します。

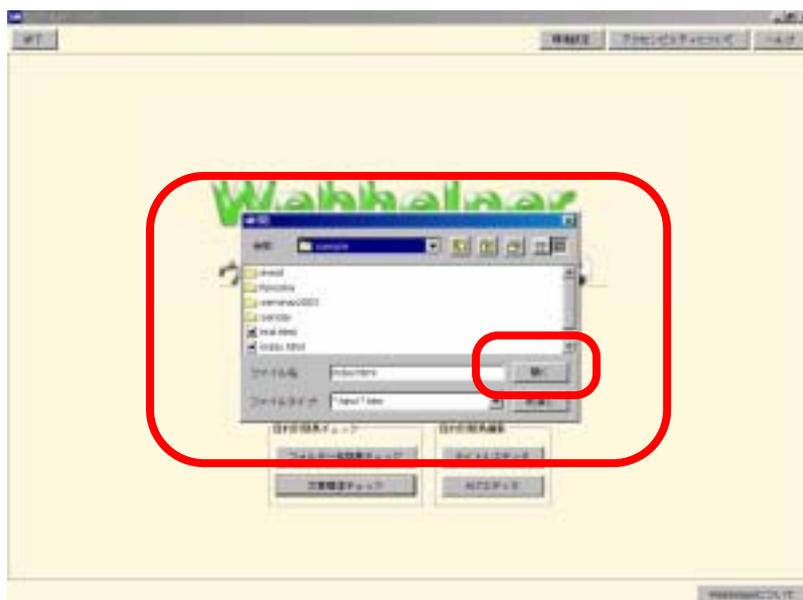
以下に手順の詳細を説明します。

点検対象の HTML ファイルを選択します。

まず、ウェブヘルパーver1.0 のトップ画面メニューから、【文章構造チェック】ボタンを選択します。



ファイル選択のためのウィンドウが開かれますので、点検を行いたい HTML ファイルのある階層へ移動し、ファイルを選択して【開く】ボタンを選択します。



注意！

点検対象はローカル環境でアクセスできる HTML ファイルです。
複数の HTML ファイルについて点検を行いたい場合は、一連の手順を繰り返して行ってください。

文章構造を確認します。

自動的にブラウザが立ち上がり、見出しタグが指定されているテキストが表示されます。

見出しタグがきちんと入れ子^{*1}になっているかどうか、簡潔で分かりやすい見出しが付けられているかどうかを確認してください。

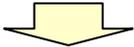


*1 入れ子：上位の見出しタグの中に順番に下位の見出しタグが位置づけられている状態。

1 1 【タイトルエディタ】機能の利用手順

「タイトルエディタ」機能の利用手順は以下のとおりです。

点検対象のフォルダを選択します。



各ページのタイトルを確認し、必要に応じて修正します。



修正結果を保存します。(で修正を行った場合)

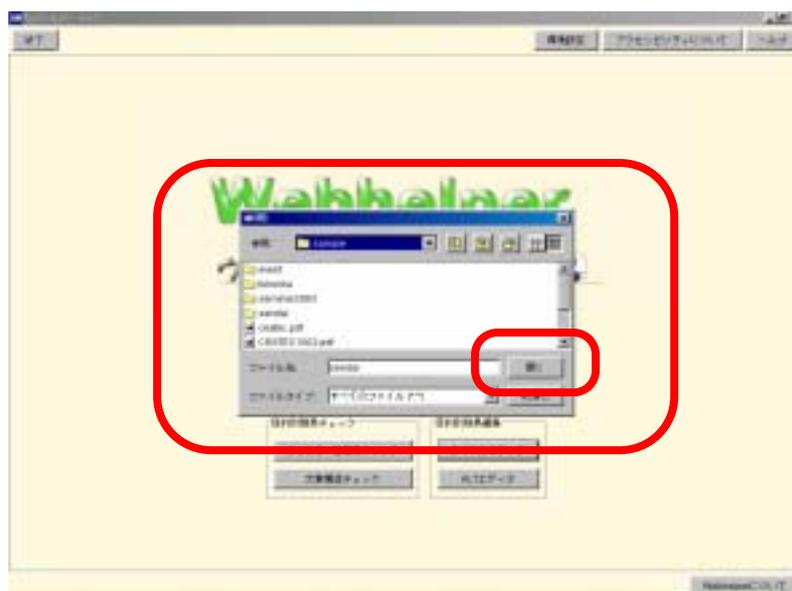
以下に手順の詳細を説明します。

点検対象のフォルダを選択します。

まず、ウェブヘルパー-ver1.0 のトップ画面メニューから、【タイトルエディタ】ボタンを選択します。



フォルダ選択のための
ウィンドウが開かれます
ので、点検を行いたいフォル
ダ（点検を行いたい
HTML ファイルの入った
フォルダ）のある階層へ移
動し、そのフォルダを選択
して【開く】ボタンを選択
します。



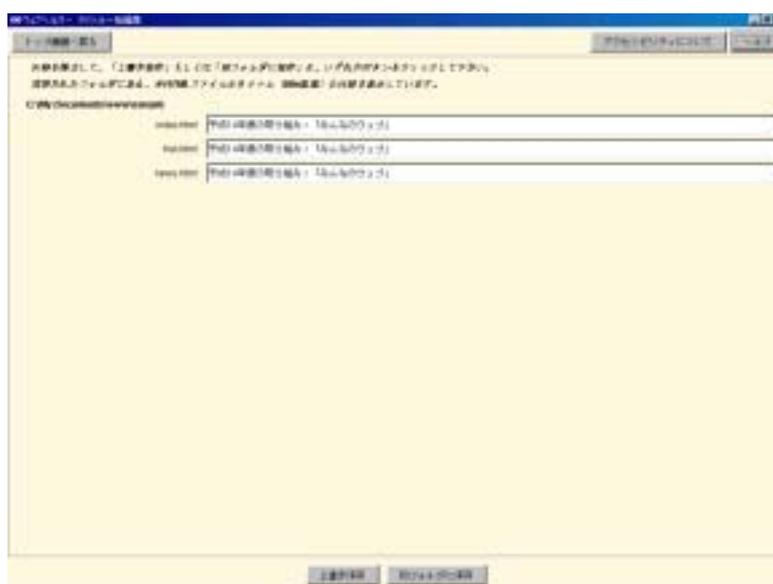
注意！

点検対象はローカル環境でアクセスできるフォルダ内の HTML ファイルです。
点検対象に、サブフォルダ以下の HTML ファイルは含まれません。複数のフォルダにつ
いて点検を行いたい場合は、一連の作業を繰り返して行ってください。

各ページのタイトルを確認し、必要に応じて修正します。

「タイトル一括編集」画面
に表示が切り替わり、対象フ
ォルダ内の全 HTML ファイル
のファイル名とタイトル
(title 要素に指定されたテキ
スト)が表示されます。

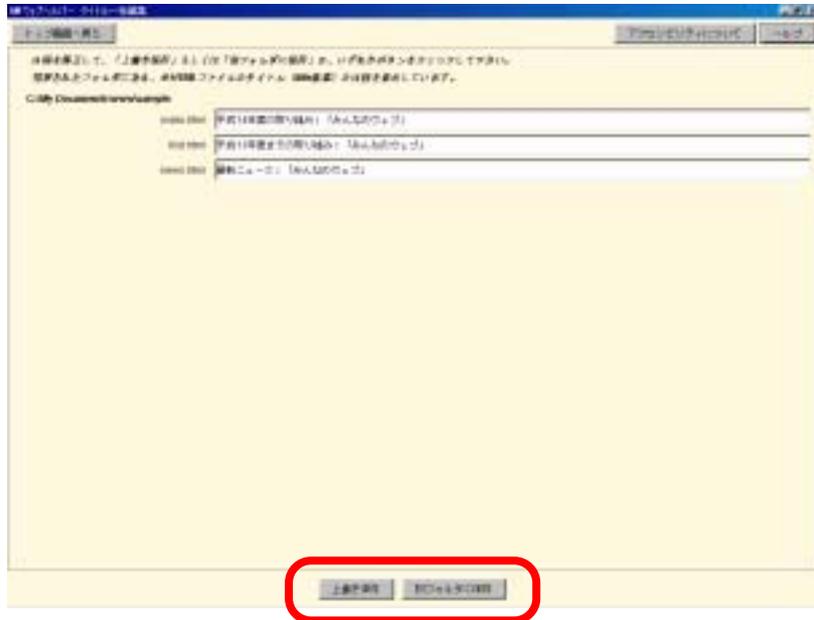
それぞれ分かりやすいペ
ージタイトルが付けられて
いるか、同じタイトルのペ
ージが無いかなどを確認し、必
要に応じて修正してくださ
い。



修正結果を保存します。(で修正を行った場合)

でタイトルの修正を行った場合は、修正結果を保存します。

対象 HTML ファイルに修正結果を上書き保存する場合は、ページ下部の【上書き保存】ボタンを選択します。上書き保存は対象となっている全 HTML ファイルに対して一括で行われますのでご注意ください。なお、修正



結果を反映した別ファイルを作成する場合は、ページ下部の【別ファイルに保存】ボタンを選択します。

修正を行わず、現状のままでよい場合は、ページ上部の【トップ画面へ戻る】ボタンを選択します。

1 2 【ALT エディタ】機能の利用手順

「タイトルエディタ」機能の利用手順は以下のとおりです。

点検対象の HTML ファイルを選択します。



代替テキストの指定有無や内容を確認し、必要に応じて修正します。



修正結果を保存します。(で修正を行った場合)

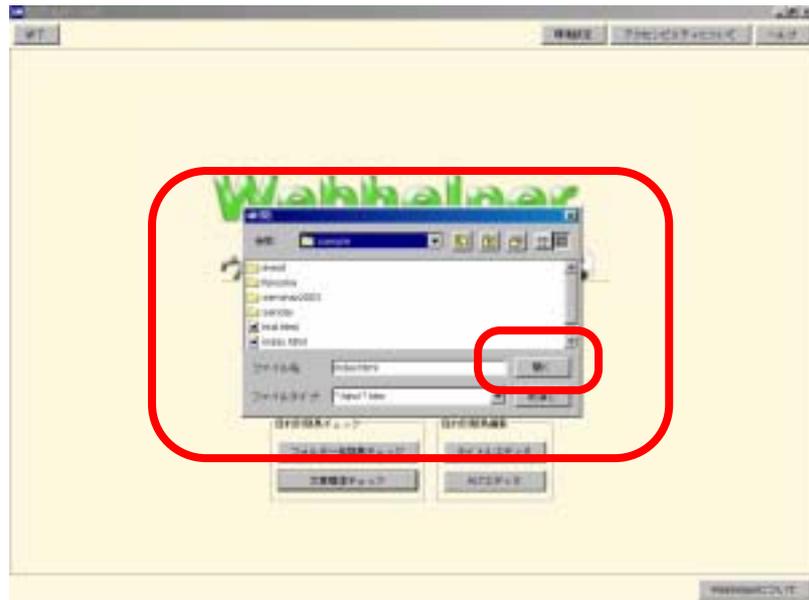
以下に手順の詳細を説明します。

点検対象の HTML ファイルを選択します。

まず、ウェブヘルパー-ver1.0 のトップ画面メニューから、【ALT エディタ】ボタンを選択します。



ファイル選択のためのウィンドウが開かれますので、点検を行いたい HTML ファイルのある階層へ移動し、ファイルを選択して【開く】ボタンを選択します。



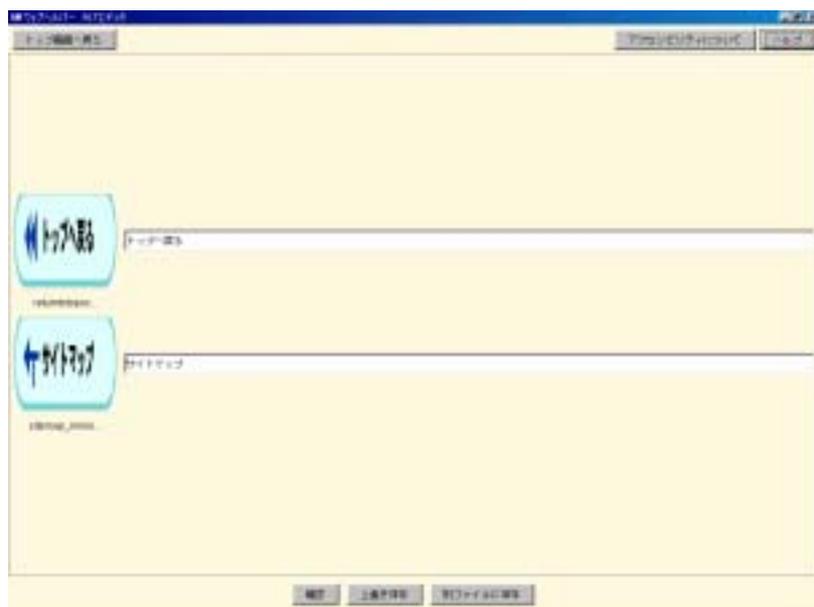
注意！

点検対象はローカル環境でアクセスできる HTML ファイルです。
複数の HTML ファイルについて点検を行いたい場合は、一連の作業を繰り返して行ってください。

代替テキストの指定有無や内容を確認し、必要に応じて修正します。

「ALT エディタ」画面に表示が切り替わり、対象 HTML ファイル内で使われている全画像（img 要素）について、画像イメージと指定されている代替テキスト（alt 属性）の内容が一覧で表示されます。

画像イメージを参照しながら、代替テキストに洩れがないかどうか、



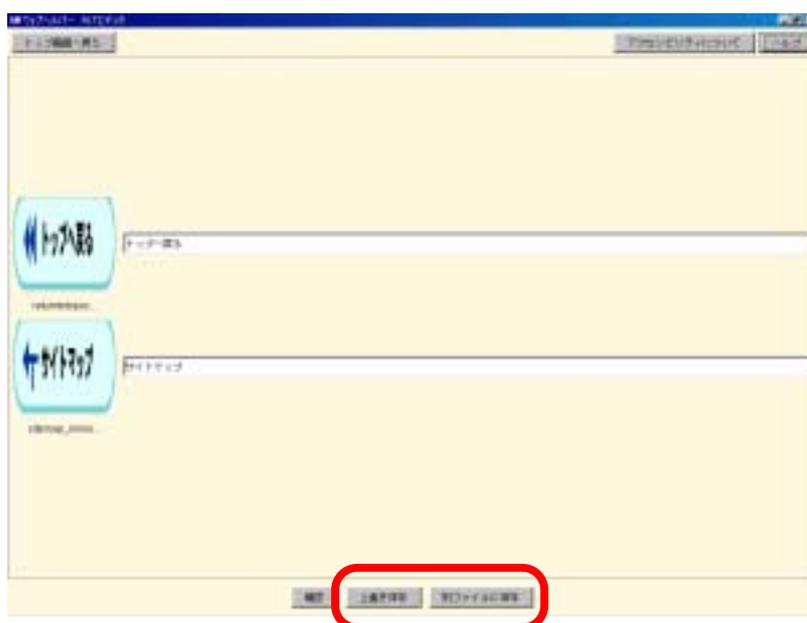
指定されている代替テキストが適切であるかを確認します。

必要に応じて、代替テキストを修正します。ページ下部の【確認】ボタンを選択すると自動的にブラウザが立ち上がり、修正結果を確認することができます。

修正結果を保存します。(で修正を行った場合)

で代替テキストの修正を行った場合は、修正結果を保存します。

対象 HTML ファイルに修正結果を上書き保存する場合は、ページ下部の【上書き保存】ボタンを、修正結果を反映した別ファイルを作成する場合は、ページ下部の【別ファイルに保存】ボタンを選択します。



修正を行わず、現状のままでよい場合は、ページ下部の【トップ画面へ戻る】ボタンを選択します。