

# みんなのアクセシビリティ評価ツール miChecker 利用ガイド

2024年 3月 31日

## 本書の目的

本書は「みんなのアクセシビリティ評価ツール miChecker」を有効に活用するための、利用方法を説明するものです。

## 目次

1.	はじめに.....	2
1.1	みんなのアクセシビリティ評価ツール miChecker（エムアイチェッカー）とは .....	2
1.2	ウェブアクセシビリティとは .....	2
1.3	ウェブアクセシビリティの重要性.....	2
1.3	ウェブアクセシビリティを推進する法律と規格の概要 .....	3
2.	miChecker について .....	4
3.	miChecker を利用する.....	7
3.1	miChecker の実行 .....	7
3.2	評価ページの指定.....	7
3.3	アクセシビリティ検証の実行 .....	7
3.4	ロービジョンシミュレーションの実行.....	8
3.5	詳細レポートのメニューを活用して達成基準や達成方法を確認する .....	9
3.6	ソースを開いて問題箇所を確認する .....	9
3.7	検証結果を保存する .....	9
4.	miChecker の出力結果を理解する.....	10
4.1	音声での利用をイメージしながら確認する.....	10
4.2	弱視者・高齢者の利用をイメージしながら確認する .....	11
4.3	問題箇所を確認する .....	11
4.4	よく問題となる点を個別に確認する .....	12
4.4.1	代替テキストを確認する .....	12
4.4.2	見出しを確認する.....	12
4.4.3	ランドマークを確認する.....	13
4.4.4	ページ内リンクを確認する .....	14
4.4.5	入力フォームとラベルを確認する.....	14
4.4.6	音声での読み上げの様子を確認する.....	15
4.5	埋め込みコンテンツを確認する .....	15
4.6	評価対象を切り替える（HTML ファイルとブラウザ内の DOM） .....	16
4.7	ブラウザのサイズを指定する .....	17
4.8	オンラインヘルプを活用する .....	17
5.	JIS X 8341-3:2016 における達成基準チェックリストの作り方 .....	18
5.1	作業の流れ.....	18
5.2	詳細レポートの確認とワークシートへの記入例 .....	19
5.3	実装チェックリストへの記入例 .....	20
5.4	達成基準チェックリストの完成例 .....	21
6.	miChecker のアクセシビリティ評価機能と CMS 等との連携.....	21
(参考)	miChecker Ver. 2 から Ver.3 への変更に関する注意点 .....	22
(参考)	HTML 文書の堅ろう(牢)(Robust)性の確保 .....	23

## 1. はじめに

### 1.1 みんなのアクセシビリティ評価ツール miChecker（エムアイチェッカー）とは

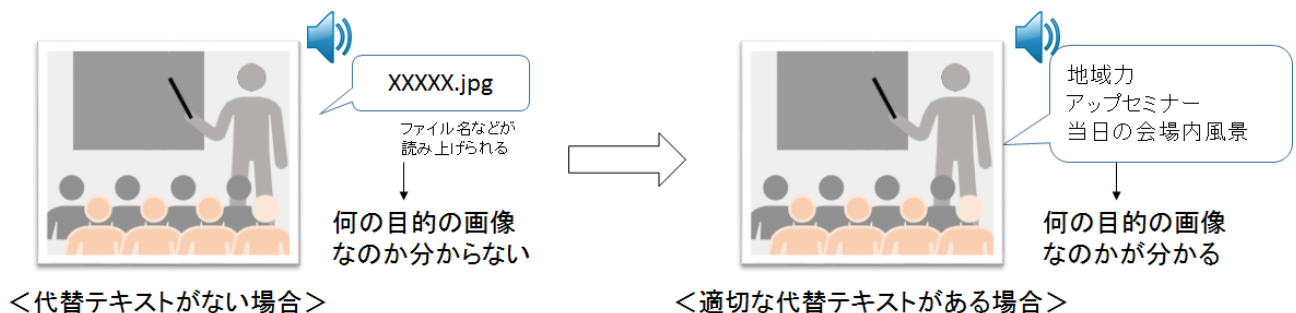
「みんなのアクセシビリティ評価ツール miChecker」（以下、miChecker）は、JIS X 8341-3:2016（高齢者・障害者等配慮設計指針—情報通信における機器、ソフトウェア及びサービス—第3部：ウェブコンテンツ）に基づくウェブアクセシビリティ対応の取組を支援するために、総務省が開発し、提供するアクセシビリティ評価ツールです。本書は、miChecker を有効に活用して頂くための情報と手順を紹介します。

### 1.2 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は、高齢者や障害者を含む誰もが、ウェブで提供されている情報や機能を支障なく利用できることを意味します。

例えば、視覚障害者の方は、ウェブの内容を音声で読み上げる支援技術などを用いてウェブを利用しています。この際、ウェブで利用されている画像は、そのままでは音声で読み上げることはできないため、視覚障害者がウェブの情報や機能を利用するための障壁（バリアー）となってしまいます。一方で、画像に、画像が何を表しているものか（何の画像なのか）分かるテキスト情報（代替テキスト）が付けられていれば、音声での読み上げが可能となり、ウェブアクセシビリティが確保されることになります。

【代替テキストの有無によるアクセシビリティへの影響の例】



### 1.3 ウェブアクセシビリティの重要性

ウェブは情報の発信・取得にとどまらず、様々な手続きやコミュニケーションの手段として、私たちの生活に欠かせないものとなっています。一方で、情報を提供する側がウェブアクセシビリティへの配慮を怠ると、高齢者や障害者が目的の情報に辿り着けない、必要な手続きが行えないなどの問題が発生し、社会生活を営む上で多大な不利益を与えることになります。場合によっては、災害時や緊急時等に必要な情報が届かず、生命の危機に直面する可能性もあります。

このような問題の発生を防ぐためにも、日頃からウェブアクセシビリティの確保に取り組むことが重要になります。障害者差別解消法（平成28年4月1日施行、令和3年5月改正）において、ウェブアクセシビリティを含む情報アクセシビリティは、国および地方公共団体などの行政機関等だけでなく事業者についても合理的配慮が求められています（改正前は、行政機関等は義務、事業者は努力義務とされています）。また、障害者から申出があった場合は、必要かつ合理的な配慮を行う必要があります。

ウェブアクセシビリティ対応には、音声での読み上げ対応に加え、字が小さい、配色が見分けづらいなど、情報利用における様々な障壁（バリアー）への対応も含まれます。これらのバリアーを取り除くことは、高齢者や障害者だけでなく、すべてのユーザに使いやすさ、快適さ、安心・安全をもたらすことにもつながります。

### 1.3 ウェブアクセシビリティを推進する法律と規格の概要

障害者差別解消法が平成 25 年 6 月に制定され、平成 28 年 4 月 1 日から施行されました。同法においては、ウェブアクセシビリティを含む情報アクセシビリティは、合理的配慮を的確に行うための環境の整備と位置づけられており、事前的改善措置として計画的に推進することが求められています。また、行政機関等は、障害者から個別の申出があった場合は、必要かつ合理的な配慮を行う必要があります。

「みんなの公共サイト運用ガイドライン（2016 年版）」より

下記に挙げた法、規格、指針等により、公的機関はウェブアクセシビリティへの対応が求められています。

- 障害者の権利に関する条約（略称：障害者権利条約）
- 障害者基本法（昭和 45 年法律第 84 号）（情報の利用におけるバリアフリー化）
- 障害を理由とする差別の解消の推進に関する法律（平成 25 年法律第 65 号。略称：障害者差別解消法）
- 産業標準化法（昭和 24 年法律第 185 号）
- 日本産業規格（JIS）：JIS X 8341-3:2016「高齢者・障害者等配慮設計指針—情報通信における機器、ソフトウェア及びサービス—第 3 部：ウェブコンテンツ」

**参考：** 国、地方公共団体などにおけるウェブアクセシビリティの取り組みに関する詳細については、総務省が提供する「みんなの公共サイト運用ガイドライン」、およびデジタル庁が提供する「デジタル・ガバメント推進標準ガイドライン」とその関連文書も参照ください。

みんなの公共サイト運用ガイドライン

[https://www.soumu.go.jp/main\\_sosiki/joho\\_tsusin/b\\_free/guideline.html](https://www.soumu.go.jp/main_sosiki/joho_tsusin/b_free/guideline.html)

デジタル・ガバメント推進標準ガイドラインおよび関連文書

[https://www.digital.go.jp/resources/standard\\_guidelines/](https://www.digital.go.jp/resources/standard_guidelines/)

（非機能要件の定義を中心に、アクセシビリティに関する事項の記載があります）

## 2. miChecker について

miChecker（エムアイチェッカー）」は、JIS X 8341-3:2016 に基づくウェブアクセシビリティ対応の取組を支援するために、総務省が開発し、提供するアクセシビリティ評価ツールです。

### 【活用例】

- 新しいページの公開前に、問題の有無を確認する
- 既に公開されているページの問題の有無を確認する
- リニューアルなどの際、業者の作成した HTML 雛形やページの検証を行う
- JIS X 8341-3:2016 に基づく適合性評価や試験に活用する
- 情報アクセシビリティ自己評価様式作成時の技術基準の作成に活用する
- ウェブアクセシビリティに関する知識の習得に活用する

### 【特徴】

- アクセシビリティの検証は、機械的に評価できることと、人が判断しなければならないことがあります。miChecker は、機械的に検証可能な項目を自動的に評価するとともに、人による判断の支援を行います。
  - ◇ 明らかな問題がある箇所を特定します。
  - ◇ 問題の可能性が高い箇所、問題であるかについて人が判断すべき箇所を特定します。
  - ◇ 問題箇所や検証項目に該当する JIS X 8341-3:2016 の関連情報へのリンクを提供し、理解を深められるよう支援します。
- JIS X 8341-3:2016 に基づく適合性評価や試験の実施を支援する付属資料を提供します。
- 音声読み上げソフトによる読み上げ順や、高齢者・弱視者の見え方などを視覚的にシミュレーションし、ウェブアクセシビリティに関する問題の把握や目視確認時の判断を支援します。

### 【注意点】

- miChecker は JIS X 8341-3:2016 に基づく検証を全て自動的に行えるものではありません。
- miChecker の付属資料等を参考に、人の判断により検証すべき項目が多数あります。

### 【miChecker の画面例】

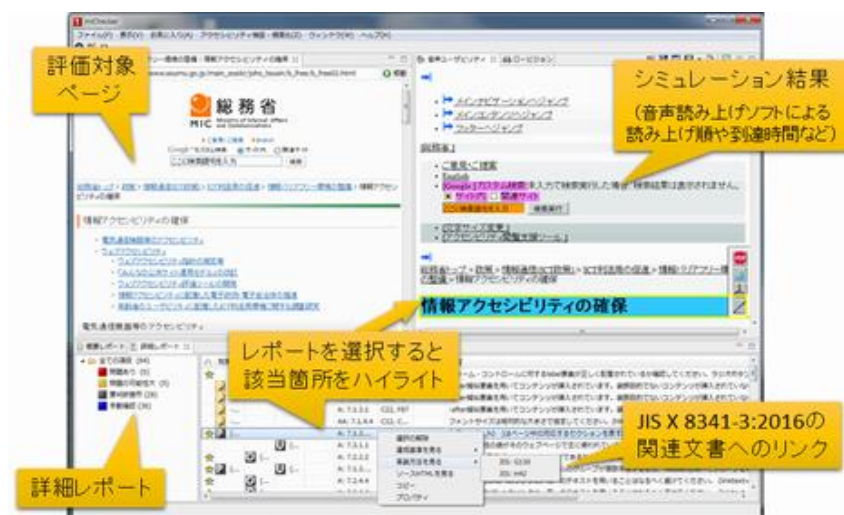


図 1. miChecker 画面の例

## 【miChecker の付属文書】

miChecker を効果的にお使い頂くために、そして適切なアクセシビリティ対応のためには、miChecker 本体による評価と併せて、miChecker 利用ガイドやワークシートなどの付属文書の利用を推奨します。miChecker の起動画面から重要な付属文書を参照することができますので、必要に応じて適宜参照してください。



図 2 miChecker 起動画面と付属文書

各付属文書のファイル名、概要、利用場面は以下の通りです。

みんなの アクセシビリティ評価ツール miChecker		<利用場面>			
<ファイル名>	<概要>	導入・ 利用時	検証作業	試験実施	ウェブ アクセシ ビリティの 知識習得
1_miChecker紹介.pdf	miCheckerの概要、利用場面、ユーザーインタフェースが紹介された概要紹介資料です。	<input checked="" type="checkbox"/>			
miChecker利用ガイド.pdf	miChecker利用における <b>必読のガイド</b> です。ウェブアクセシビリティの必要性やmiCheckerの利用方法が図入りで解説されています。	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
miChecker 達成基準別活用法.pdf	JIS X 8341-3:2016の達成基準とmiCheckerを活用した対応方法を事例を交えて解説しています。		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
miCheckerを用いた 試験手順書.pdf	JISに基づく試験を、miCheckerを用いて実施する際に参照する手順書です。			<input checked="" type="checkbox"/>	
ワークシート.xlsx ワークシート.ods	このワークシートを用いることで、検証作業を適切に実施することができます。miCheckerの評価結果を取り込むことが可能です。		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
CMS等連携手順書.pdf	miCheckerの評価機能をCMS等と連携するための開発手順を紹介した資料です。		<input checked="" type="checkbox"/>		

図 3 各付属文書のファイル名、概要、利用場面

(付属文書はmiChecker の ZIP ファイル展開後の「2\_手順書」フォルダーにも同梱されています。)



## 【miChecker のユーザインターフェースの概要】

miChecker は、大きく分けて5つパーツからできています(図 4)。上部に表示されている2つの項目のうち、左側が「ブラウザ」、右側が「視覚化エリア」(「音声ユーザビリティ」ビュー/「ロービジョン」ビュー)です。下部に表示されているのは、「レポートエリア」です。「レポートエリア」には「概要レポート」ビューと「詳細レポート」ビューの2つのビューがあります。

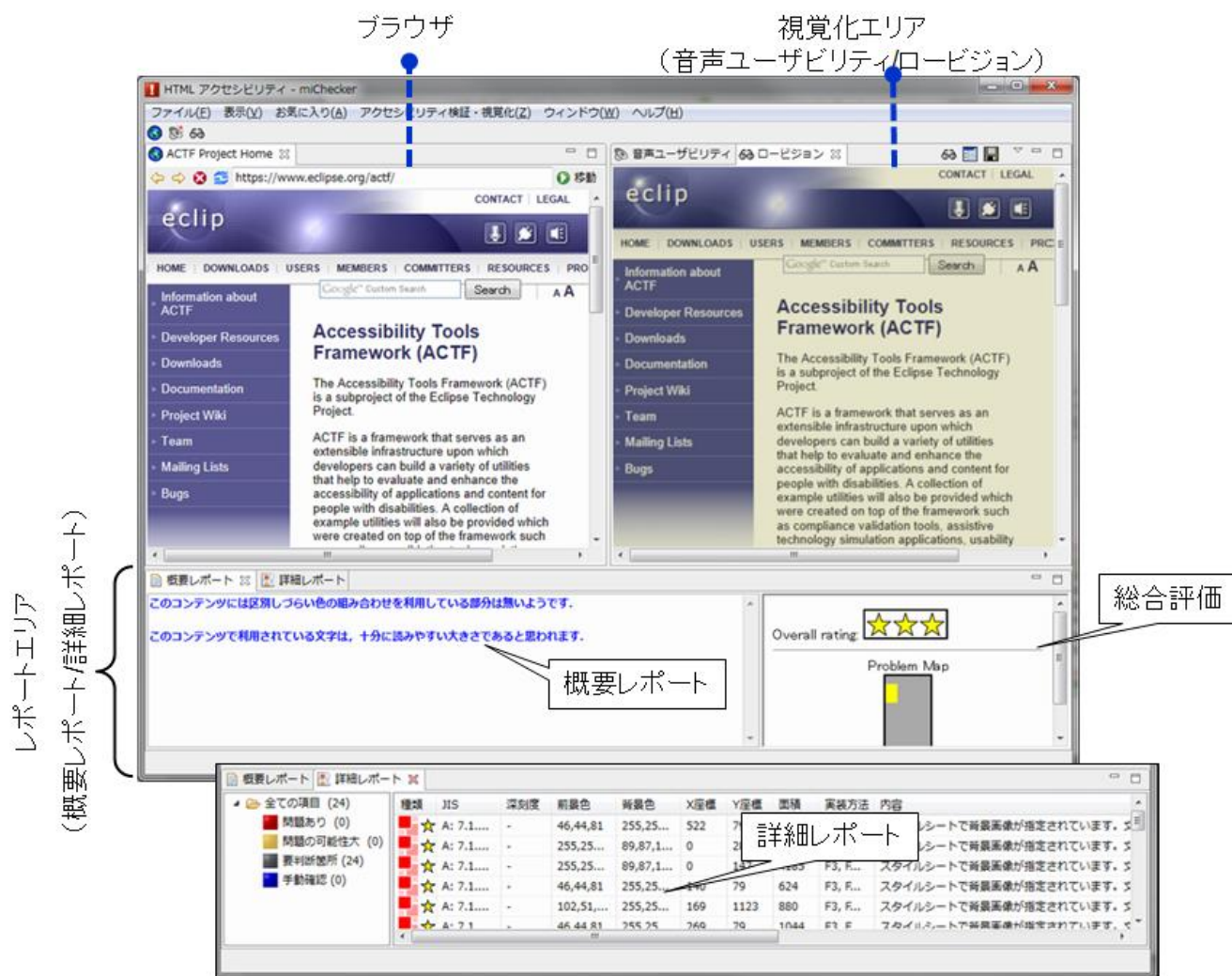


図 4 miChecker のユーザインターフェース

miChecker では、まず画面左側の「ブラウザ」に評価対象となるウェブページを表示した上で、「アクセシビリティ検証・音声ユーザービリティ視覚化」機能や「ロービジョンシミュレーション」機能を用いて、アクセシビリティの検証を行います。miChecker 右上にある視覚化エリアには、スクリーンリーダーなどの支援技術を用いて音声でウェブページにアクセスしているユーザにとってのウェブページの使い勝手や、一部のロービジョンユーザ（弱視、色覚異常、白内障、あるいはそれらが組み合わせられた症状を持つユーザ）にとってのウェブページの見え方のシミュレーション結果が表示されます。画面下部のレポートエリアには、miChecker によるアクセシビリティ評価の結果が表示されます。それぞれの機能やその使い方については、次章以降で紹介します。

### 3. miChecker を利用する

#### 3.1 miChecker の実行

miChecker を「miChecker 導入手順書」に従い導入します。導入が完了したら、導入先のフォルダーの中にある「miChecker」を実行します。

(補足：OS の設定によっては、miChecker.exe と拡張子と共に表示されています。)



図 5. miChecker を起動

#### 3.2 評価ページの指定

miChecker の「ブラウザ」内に表示されているアドレスバー (図 6-①) に、評価対象ページの URL を入力し[移動]ボタン (図 6-②) をクリックします。(もしくは、Alt+d でアドレスバーに移動し、URL の入力後 Enter キーを押す。)

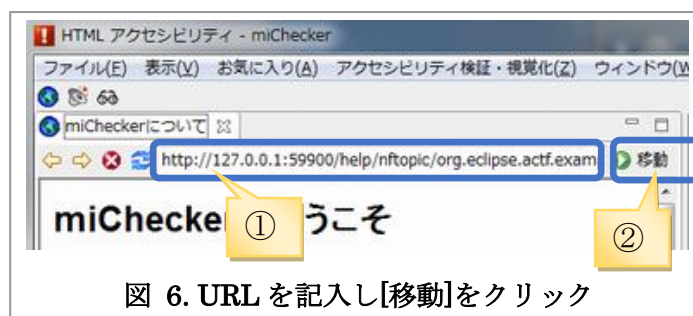


図 6. URL を記入し[移動]をクリック

#### 3.3 アクセシビリティ検証の実行

対象となるページが表示されたら、[アクセシビリティ検証・音声ユーザービリティ視覚化]ボタン(スピーカーのアイコン)をクリックしてアクセシビリティを検証します(図 7)。

メニューバーの「アクセシビリティ検証・視覚化」から「アクセシビリティ検証・音声ユーザービリティ視覚化」(もしくはCtrl+Shift+B)を選択しても検証を実行することが出来ます。

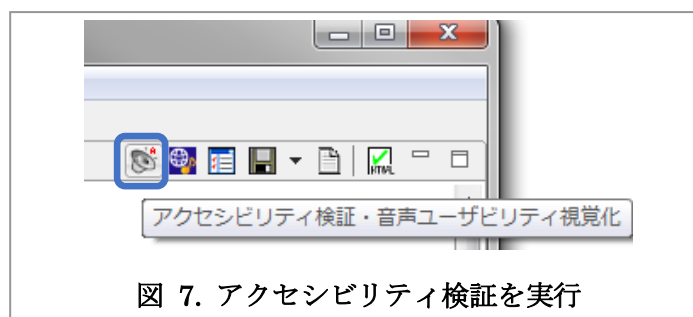


図 7. アクセシビリティ検証を実行

検証が完了すると、検証結果が下側の「詳細レポート」に、音声ユーザービリティの視覚化結果が右側の「音声ユーザービリティ」に表示されます(図 8)。検証結果の確認については、3.5 節および 4 章で解説します。

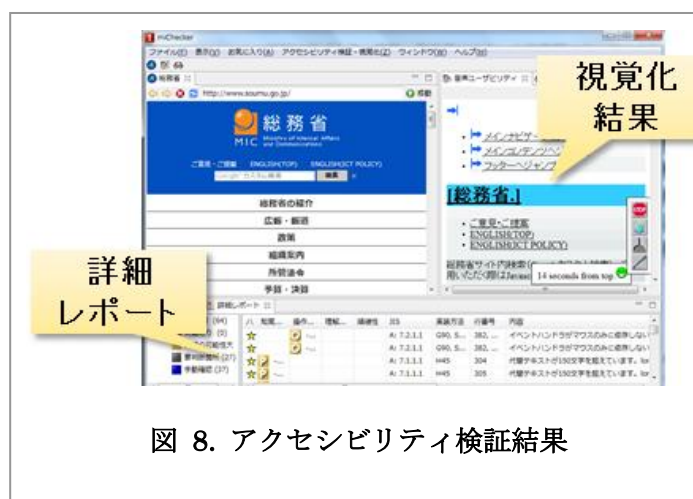


図 8. アクセシビリティ検証結果

補足:各ビューをキーボード操作で移動するには、Ctrl+Shift+F7 を順に押してください。

miChecker には、下側のレポートエリアに「詳細レポート」、「概要レポート」の 2 種類、右側の視覚化エリアに「音声ユーザービリティ」、「ロービジョン」の 2 種類、合計 4 種類のビューが配置されています。



### 3.4 ロービジョンシミュレーションの実行

miChecker では HTML ファイルやウェブページを開いてロービジョンシミュレーションを実行することにより、一部のロービジョンユーザ（弱視、色覚異常、白内障、あるいはそれらが組み合わされた症状を持つユーザ）にとってのアクセシビリティおよびユーザビリティをチェックすることができます。

評価対象となるページが表示されたら、[ロービジョン シミュレーション]ボタン（メガネのアイコン）をクリックしてシミュレーションを実行します(図 9)。メニューバーの「アクセシビリティ検証・視覚化」から「ロービジョン シミュレーション」（もしくはCtrl+Shift+0）を選択してもシミュレーションができます。

シミュレーションが完了すると、表示上の問題に関する検証結果が下側の「詳細レポート」に、ロービジョンユーザの見え方のシミュレーション結果が右側の「ロービジョン」に表示されます(図 10)。

設定パネル（ロービジョン シミュレーション 設定）(図 12. ロービジョンシミュレーション設定図 12)では、シミュレーションするロービジョンの症状や程度を設定することができます。設定では、チェックボックスを用いることで、下記の 3 項目を自由に組み合わせることができます。また、これらの症状はそれぞれ固有のパラメーターを持っています。

・**視力**：視力の悪い方の見え方をシミュレートします。ルーラーをスライドさせて、視力 0.1 から 1.0 までを設定することができます。

・**色覚異常**：色覚異常の方の見え方をシミュレートします。一般的な色覚異常の 3 タイプ（第一色覚異常、第二色覚異常、第三色覚異常）から選択できます。

・**水晶体透過率（年齢）**：加齢とともに水晶体が光（特に短い波長）を通しづらくなる現象をシミュレートします。ルーラーをスライドさせて、20 才から 60 才までのユーザの平均的な見え方をシミュレーションすることができます。

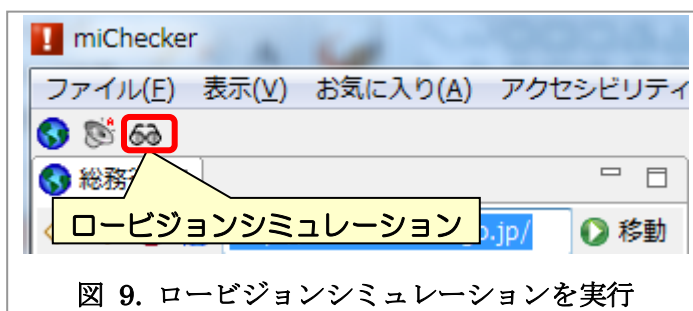


図 9. ロービジョンシミュレーションを実行



図 10. ロービジョンシミュレーション実行結果

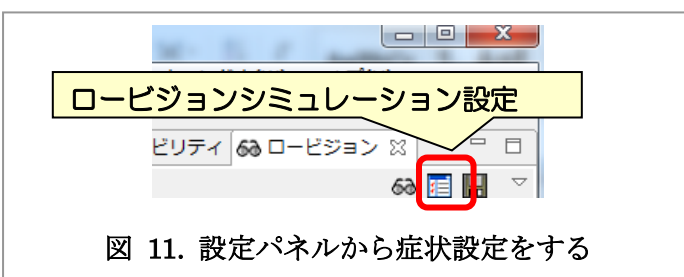


図 11. 設定パネルから症状設定をする



図 12. ロービジョンシミュレーション設定

### 3.5 詳細レポートのメニューを活用して達成基準や達成方法を確認する

miChecker の下部にある「詳細レポート」のビューにおいて、確認したい問題点の行を選択し、右クリック（もしくは Shift+F10）すると、問題点の詳細を知るためのメニューが表示されます（図 13）。

問題に関連する「JIS X 8341-3:2016」の達成基準の内容を確認したい場合には、[達成基準を見る]を選択し、次に表示される達成基準番号を選択してください（図 13）。新規にブラウザが開き、ウェブアクセシビリティ基盤委員会（WAIC）が提供している達成基準の解説ページが表示されます。同様に、達成方法の解説ページを開くには[達成方法を見る]、HTML のソースを確認する場合には [ソース HTML を見る]、該当箇所について詳細情報をクリップボードにコピーするには [コピー]、詳細情報を一覧するには [プロパティ]（図 14）を選択してください。



図 13. 問題点の詳細を知るためのメニューの例



図 14. プロパティの表示例

### 3.6 ソースを開いて問題箇所を確認する

3.5 のメニューから [ソース HTML を見る] を選択した場合は、新たに「HTML ソース」ウィンドウが開きます。問題の場所が明確な場合は、該当箇所が反転表示されます（図 15）。

「HTML ソース」ウィンドウでは、該当箇所の HTML ソースをクリップボードにコピーすることや、文字列の検索も可能です。

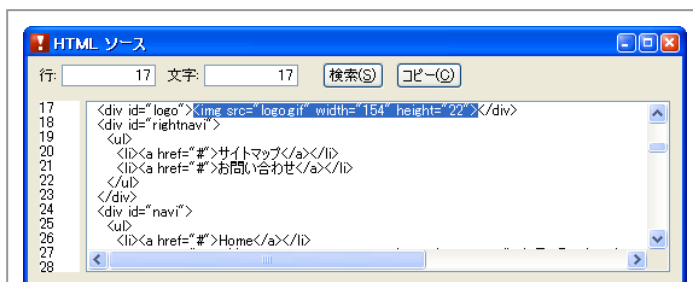


図 15. ソース上で問題箇所を確認

### 3.7 検証結果を保存する

miChecker の検証結果は csv 形式で保存することができます。検証の実行後、「結果を保存」ボタンをクリックして保存用のダイアログボックスを開き保存してください（図 16）。

ここで保存した検証結果はワークシートと連携させて活用することも可能です。（参照 5.2 詳細レポートの確認とワークシートへの記入例）

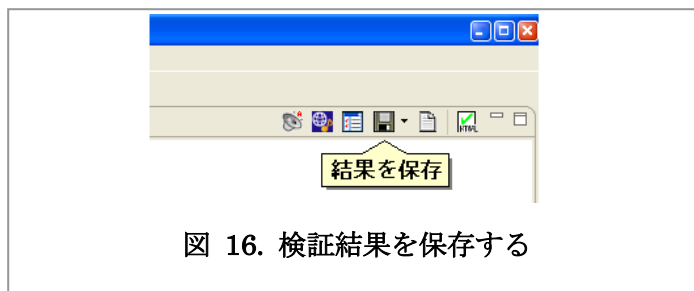


図 16. 検証結果を保存する

補足：キーボード操作で検証結果を保存する場合には、Ctrl+Shift+F7 で「音声ユーザビリティ」のビューに移動し、そこから tab キーを押してツールバーに移り、カーソルキーで「結果を保存」メニューを選択してください。

## 4. miChecker の出力結果を理解する

### 4.1 音声での利用をイメージしながら確認する

視覚に障害をもつユーザやロービジョンユーザの中には、スクリーンリーダーなどの支援技術を用いてウェブページを利用している人がいます。まず、このような支援技術を用いて音声でウェブページを利用した場合の状況を「音声ユーザビリティ」ビューで確認しましょう。

3.3 節で紹介した手順と同様に、miChecker の[アクセシビリティ検証・音声ユーザービリティ視覚化実行]ボタン(スピーカーのアイコン)をクリックして、アクセシビリティ検証・音声ユーザービリティ視覚化を実行すると、ウェブページを音声で利用した場合に読み上げられる情報をシミュレーションした結果が右側の「音声ユーザービリティ」ビューに表示されます(図 17)。ここでは、ページ内で用いられている画像などの非テキストコンテンツは音声読み上げのための代替テキスト(例: alt 属性の値など)に置き換えられると共に、テーブル構造以外の視覚的な表現(配置や形状、文字のスタイルなどの情報)は失われた状態となっています。ウェブページのアクセシビリティを確認する際は、視覚化結果として表示されているテキストだけを讀んで、ウェブページの内容が正しく伝わっているか確認することが重要です。

また、「音声ユーザービリティ」ビューでは、ユーザがページ内の情報に到達するまでに必要となる時間の予測結果を、画面の明るさを用いて表現しています。従って、視覚化結果が暗ければ暗い箇所ほど、ユーザがその部分に到達するまでに時間が掛かってしまう可能性が高いことを表しています。一方、ウェブページの途中から再び明るくなる箇所がある場合は、そこにアクセシビリティの配慮として見出しやページ内リンク、ナビゲーションのためのランドマーク(目印)が設定されて、ユーザが支援技術の機能を用いて容易にアクセスできることを意味しています。ウェブページが全体に暗くなっている場合には、ページ内の重要な項目や、記載内容が切り替わる部分などに見出しやランドマークを用いるなどして、ウェブページ全体が明るくなる、すなわちウェブページ全体にアクセスしやすくなる様に HTML を修正しましょう。

次に、「音声ユーザービリティ」ビュー内に「赤いビックリマーク」が表示されている場合は、その場所に何らかのアクセシビリティの問題があることを示しています(図 18-①)。該当箇所にカーソルを当てると、問題の内容を知ることができます(画面下側の「詳細レポート」にも問題点として報告されています)。

他に、「音声ユーザービリティ」ビュー内でピンクで示されている部分(図 17-①)はユーザに入力操作などを行わせる入力フォームのためのラベル、水色で示されている部分(図 17-②)は見出しを示しています。この他にも、miChecker は、テーブルヘッダ(黄緑色)、入力フォーム(オレンジ色)なども色付きで示します。ウェブページのアクセシビリティを検証する場合には、視覚化結果を用いてこれらの情報を概観した上で、「詳細レポート」に報告されている問題一つ一つについて検証を進めていくと良いでしょう。

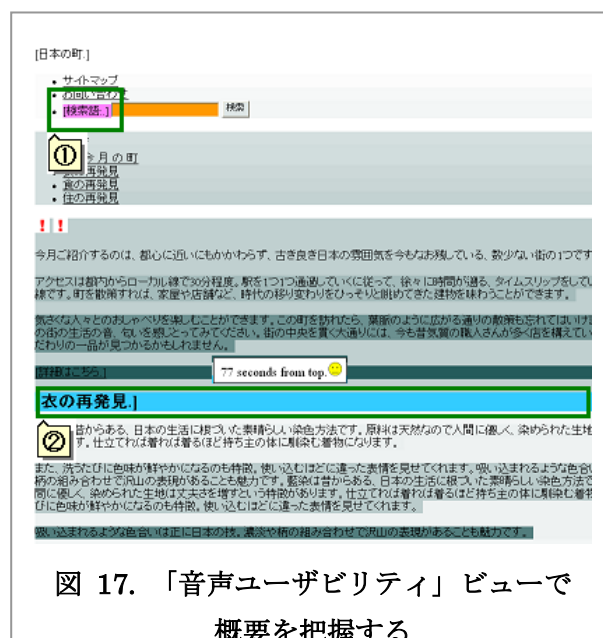


図 17. 「音声ユーザービリティ」ビューで概要を把握する

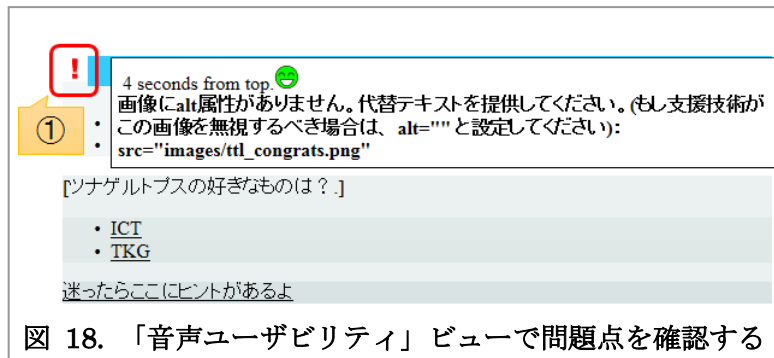





図 18. 「音声ユーザービリティ」ビューで問題点を確認する

## 4.2 弱視者・高齢者の利用をイメージしながら確認する

次に、3.4 節で紹介した手順と同様に、[ロービジョン シミュレーション実行]ボタン(メガネのアイコン)をクリックしてロービジョンシミュレーションを実行してみましょう。右側の「ロービジョン」ビューに、設定で指定した症状のロービジョンユーザにとっての対象ページの見え方のシミュレーション結果が表示されます。図 19 の例では、晴眼者の見え方(左側のブラウザの画面)と比べるとコントラストが低くなってしまうことがわかります。

ウェブページのリニューアルや新規ページの追加に向けてデザインを検討する場合などには、この機能を用いて、弱視者・高齢者などに見えづらい配色となっていないかを確認すると良いでしょう。

また、ロービジョンシミュレーション時に検出された問題点などは、画面下側の「詳細レポート」に一覧表示されます。「詳細レポート」の表示アイコンは以下のような区分となっていますので、これらを把握した上で、「詳細レポート」に報告されている問題一つひとつを検証していくと良いでしょう。

	このアイコンは他のアイコンと併用されます。このアイコンが表示されている行を選択すると、その問題点に対応する要素がシミュレーションビュー上で矩形枠に囲まれて表示されます。
	色に関する問題点を表します。
	視力(フォントの大きさ)に関する問題点を表します。

## 4.3 問題箇所を確認する

miChecker でアクセシビリティ検証やロービジョンシミュレーションを実行すると、発見された問題点などが画面下部の「詳細レポート」に表示されます(図 20)。「詳細レポート」内の左側の列に☆印が付いている場合、その行を選択状態にすると、画面右側の視覚化エリア(シミュレーションの結果)の該当箇所が黄色の枠(図 21-①)でハイライトされます。複数行を選択し、まとめてハイライトさせることも可能です。

この機能を用いることで、報告された問題が、どの場所におけるものなのかを視覚的に確認することが出来ます。3.6 節で紹介した HTML のソースを開く方法も利用しながら検証作業を進めてください。

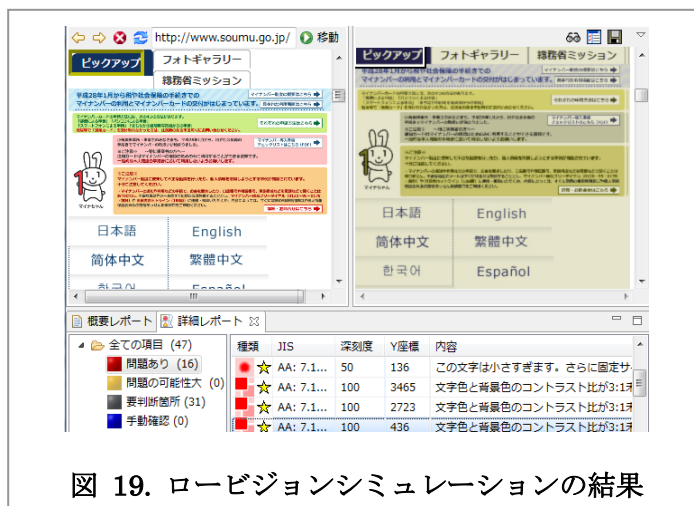


図 19. ロービジョンシミュレーションの結果

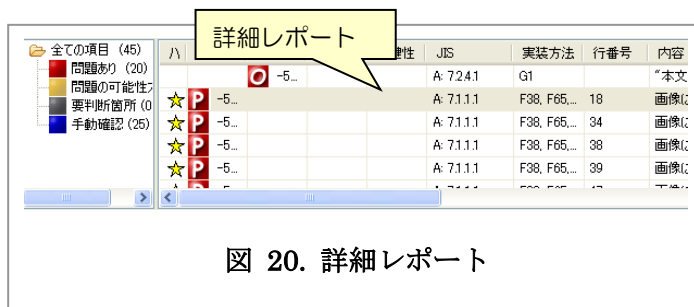


図 20. 詳細レポート

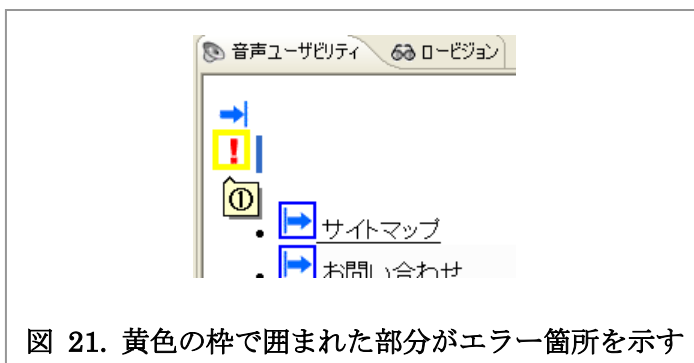


図 21. 黄色の枠で囲まれた部分がエラー箇所を示す



## 4.4 よく問題となる点を個別に確認する

### 4.4.1 代替テキストを確認する

この項目では、スクリーンリーダーなどの支援技術を用いてウェブページに音声でアクセスしている利用者が、画像などの非テキストコンテンツの内容を理解できるようになっているか否かを確認します。miCheckerの「音声ユーザビリティ」ビューでは、ページ内で用いられている画像など

の非テキストコンテンツは代替テキスト（例：画像の alt 属性の値など）に置き換えられて表示されます。評価対象のウェブページと視覚化の結果を見比べながら、非テキストコンテンツが代替テキストに置き換わった場合でも、同じ機能及び情報が提供出来ているか確認してください。

また、miChecker には、非テキストコンテンツに代替テキストが付与されていない場合や、不適切と思われる代替テキストが付与されている場合に指摘する機能があります。問題箇所や問題と思われる箇所は、視覚化結果内で「赤いビックリマーク」（図 22-①）で表示されていますので、該当箇所にカーソルを当てて問題の内容を確認し、適切な代替テキストを付与するようにしてください。また、画面下側の「詳細レポート」にも問題点として報告されていますので、全ての問題箇所をまとめて確認したい場合には、「詳細レポート」の一覧も確認するようにしてください。

（より詳細な確認方法については、「miChecker 達成基準別活用法」の【適合レベル A】達成基準 1.1.1 に関する説明を参照してください。）

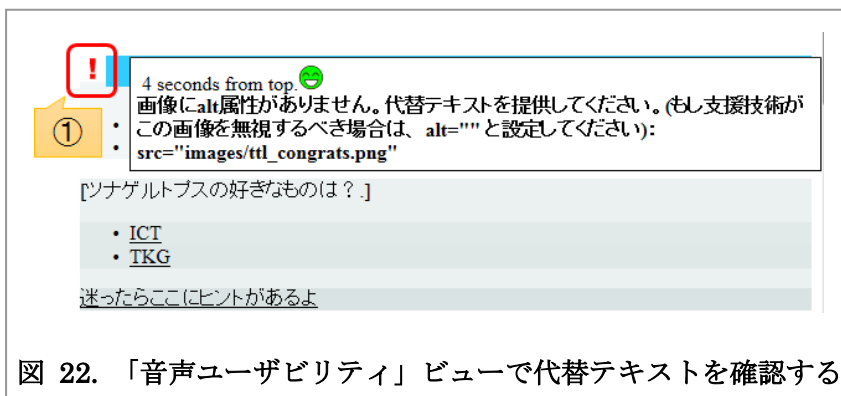


図 22. 「音声ユーザビリティ」ビューで代替テキストを確認する

### 4.4.2 見出しを確認する

スクリーンリーダーなどの支援技術は、ウェブページ内に設定された見出し(h1～h6の要素)を順に読み上げる機能を持っています。見出しが正しく提供されていれば、支援技術の利用者は、ウェブページで提供されている情報の概観を本の目次を読むような形で理解できるようになります。

miCheckerの「音声ユーザビリティ」ビューでは、見出し部分が予め設定してある背景色(初期値は水色)で表示されます(図 23-①)。見出し部分にマウスを当てると、見出しのレベルも表示されます。これらの機能を用いて、ウェブページに見出しが正しく付与されているか確認しましょう。具体的には、見出しだけを読み上げたときやページ内から見出しから見出しへ移動したときに、本の目次のように利用できるようになっているか否かを確認してください。

（より詳細な確認方法については、「miChecker 達成基準別活用法」の【適合レベル A】達成基準 1.3.1 および【適合レベル AA】2.4.6 に関する説明を参照してください。）

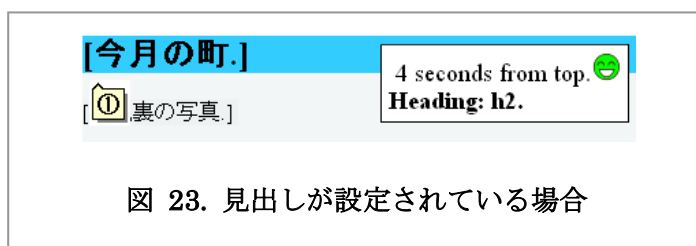


図 23. 見出しが設定されている場合



### 4.4.3 ランドマークを確認する

前節で紹介した見出しに加えて、ウェブページ内をナビゲーションするためのランドマーク（目印）を指定することも可能になっています。miChecker の「音声ユーザビリティ」ビューでは、ウェブページ内をナビゲーションするためのランドマーク（目印）となる要素が設定されている箇所に、山形のアイコン（**main** など）が表示されます。ランドマークが指定されていると、スクリーンリーダーなどの支援技術を利用している場合に、メインコンテンツや、ヘッダー・フッター部分、それぞれの記事などへ簡単に移動することが可能になります。それぞれのランドマークで指定された区画は、青い線で周囲を囲まれますので、下の表を参考に、指定された内容と区画が一致しているか確認するようにしてください。

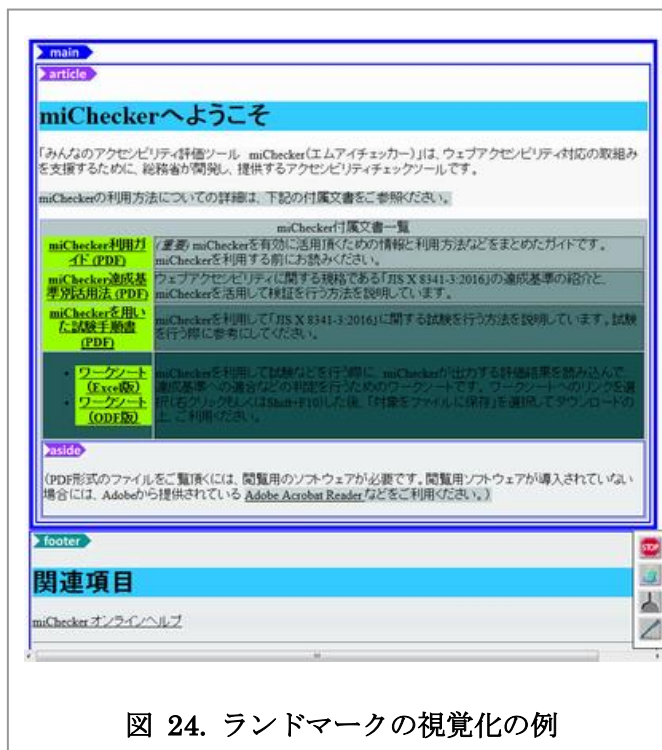


図 24. ランドマークの視覚化の例

ランドマークの種類とアイコン	役割と説明	注意点
<b>main</b> main 要素	ウェブページのメインコンテンツを指定します	<ul style="list-style-type: none"> <li>・ ページ内に 1 つだけ指定します</li> <li>・ 他のランドマークの中には作成できません</li> <li>・ 互換性確保のため、当面は role 属性 (role="main") と併記してください</li> </ul>
<b>header</b> header 要素	ウェブページのヘッダー部分(サイトのロゴや共通のナビゲーション部分など)を指定します。	<ul style="list-style-type: none"> <li>・ header 要素と body 要素の間に他のランドマークとなる要素が無い場合に、ウェブページのヘッダーを指定する役割となります</li> <li>・ 互換性確保のため、当面は role 属性 (role="banner") と併記してください</li> </ul>
<b>footer</b> footer 要素	ウェブページのフッター部分(著作権表示や連絡先など)を指定します。	<ul style="list-style-type: none"> <li>・ 通常はページの最後に配置されます</li> <li>・ 互換性確保のため、当面は role 属性 (role="contentinfo") と併記してください</li> </ul>
<b>nav</b> nav 要素	ナビゲーションメニュー、ナビゲーションリンク集などを指定します	<ul style="list-style-type: none"> <li>・ 複数箇所に指定することが可能です</li> </ul>
<b>article</b> article 要素	ウェブページ内の一つの記事のかたまりを指定します	<ul style="list-style-type: none"> <li>・ 指定した部分を単独で抜き出しても利用者に情報が伝わることを指定の目安になります</li> </ul>
<b>aside</b> aside 要素	前後の情報に関連しているが、補足的な情報である場合に指定します	<ul style="list-style-type: none"> <li>・ 補足や備考などを記載する場合などに指定します</li> </ul>

#### 4.4.4 ページ内リンクを確認する

見出しやランドマークなどの要素を用いる方法に加えて、ページ内の各部に移動することができるページ内リンクを用いて、利用者がページ内をナビゲートするのを支援することもできます。miChecker の「音声ユーザビリティ」ビューでは、ページ内リンクのリンク元が[ | → ]アイコン、リンク先が[ → | ]アイコンとして表示されます。これらのアイコンをクリックすると、

参照関係を示す赤い矢印が表示されます。この機能を用いることで、適切な位置にページ内リンクがあるか、ページ内リンクの参照先が誤っていないかなどを視覚的に確認することができます。参照関係が誤っている場合には、ページ内リンクのリンク元、リンク先を修正するようにしてください。

(ページ内リンクのアイコンをもう一度クリックすることで、表示された矢印を消すことができます。)

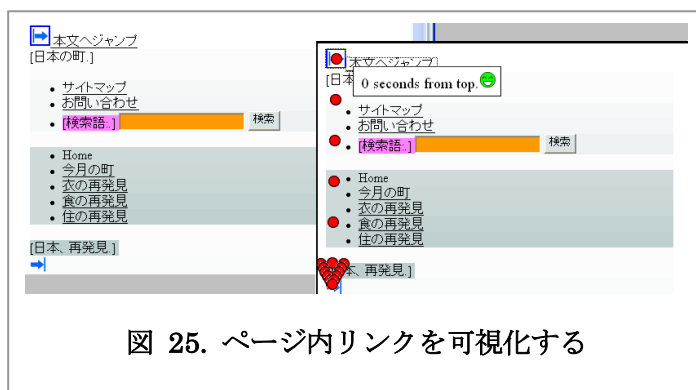


図 25. ページ内リンクを可視化する

#### 4.4.5 入力フォームとラベルを確認する

利用者に入力を促す入力フォームと、その説明を記載したラベルを関連付けておくことで、スクリーンリーダーなどの支援技術の利用者が入力フォームの内容を理解しやすくなったり、細かな操作が苦手な利用者がフォーム要素を選択するのが容易になったりします。miChecker の「音声ユーザビリティ」ビューで入力フォームのある部分を表示すると、ラベルが付いている場合には色付きで表示されます。図 26 の例では、「検索語」がピンク色で表示されており、ラベルが付与されていることが分かります。この場合は、ラベルが入力フォームの内容を正しく表現しているか確認を行うようにしてください。一方、ラベルが付いていない場合は、何も色が付かずに表示されます(図 27)。ラベルがついていない場合には、ラベルの付与を行います。

次に、入力フォームとラベルが正しく関連付けされているか確認します。ラベルと入力フォームの組み合わせ毎に、「詳細レポート」ビューの「要



図 26. ラベルの指定がある場合  
(検索語の背景がピンク色になっている)



図 27. ラベルが付いていない場合  
(検索語の背景は元と変わらない)

判断箇所」の一覧の中に「label 要素が対応するフォームコントロールの目的を表していること、および label 要素が画面上に表示されていることを確認してください」という内容の項目が表示されますので、順に選択してください。「音声ユーザビリティ」ビューでラベル部分と対応するフォームコントロール部分(テキスト入力やラジオボタン、チェックボックスなど)がハイライト表示されますので、その関連付けおよびラベルの内容が適切か否かを確認します(図 28)。



図 28. ラベルと入力フォームの関係を確認する

より詳細な確認方法については、「miChecker 達成基準別活用法」の [適合レベル A] 達成基準 3.3.2 および [適合レベル AA] 2.4.6 に関する説明を参照してください。)

#### 4.4.6 音声での読み上げの様子を確認する

「音声ユーザビリティ」ビューには、ページレイアウトを排除し、音声ブラウザなどで聞き進めた場合の状況と同様に内容を表示する**音声ブラウザモード**が用意されています。このモードは「アクセシビリティ検証・音声ユーザビリティ視覚化実行(音声ブラウザモード)」ボタンを選択して実行します(図 29)。

この機能を活用することで、ユーザが音声でウェブページを利用する際の様子を、テーブルの読み上げ順序や入力フォーム読み上げの様子なども含めて再現し、確認することができます。ウェブページのアクセシビリティを確認する際の最後のステップとして、評価対象のウェブページと視覚化の結果を見比べながら、同じ機能及び情報が提供出来ているか確認してください。

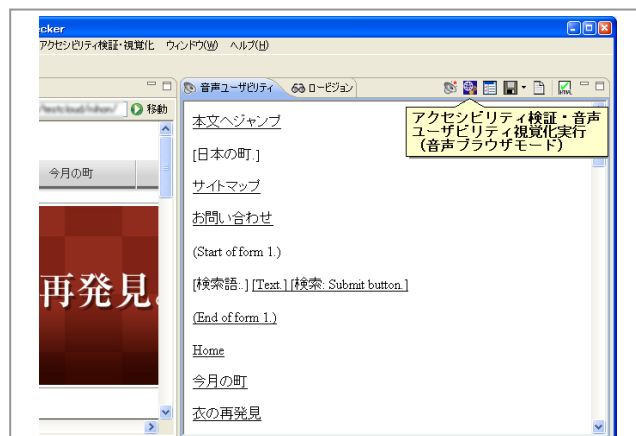


図 29. 音声ブラウザモードによる表示の例

#### 4.5 埋め込みコンテンツを確認する

SNS や動画、地図などの外部コンテンツをウェブページに埋め込んでいる場合、それらの埋め込みコンテンツも含めてアクセシビリティを確保する必要があります(※)。miChecker では、他のウェブページの埋め込みを行う iframe 要素を見つけた際に、スクリーンリーダーやキーボードを用いたアクセスが可能な確認を促すと共に、それぞれの埋め込みコンテンツを新たなタブで開く様に促す機能を有します(図 30)。

(※「みんなの公共サイト運用ガイドライン (2016 年版)」6.5. 外部サービスを活用した情報発信における取組も参照ください)

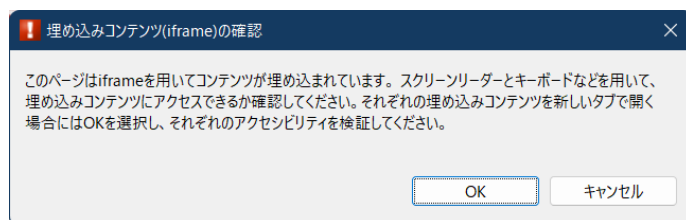


図 30 iframe を用いた埋め込みコンテンツが発見された例

図 30 の確認画面で「OK」を選択すると、画面左上のブラウザ部分に新たなタブが追加されます。現在、アクセシビリティの検証を進めているウェブページ(埋め込みコンテンツを含むページ)の検証が完了した後、新しく開かれたタブについても検証を行ってください。埋め込みコンテンツに問題が発見された場合には、コンテンツの提供元や埋め込みを行った担当者などと連携して問題を修正してください。

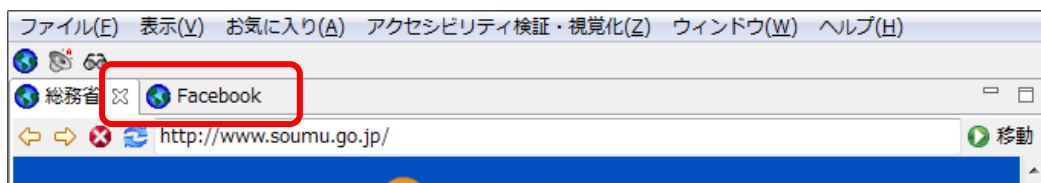


図 31 新しく開かれたタブの例

なお、埋め込みコンテンツを新たなタブで開いた場合、他のページに埋め込まれていた場合と同じコンテンツが得られるとは限らないため、当初の評価対象であったウェブページ(埋め込みコンテンツを含むウェブページ)に対してスクリーンリーダーやキーボードなどを用いて確認を行うことを推奨します。

## 4.6 評価対象を切り替える（HTML ファイルとブラウザ内の DOM）

miChecker ではアクセシビリティを評価する際に、3.2 評価ページの指定において評価対象ページの URL として指定したウェブページの HTML ファイルそのものを評価する方式と、このウェブページをブラウザ内で表示する際に用いられている Document Object Model（DOM）と呼ばれる文書構造を対象に評価する方式の 2 つの評価方法を選ぶことが出来ます。前者の[HTML ファイル]を評価する方式は、発見された問題箇所の行番号なども明らかになるため、HTML ファイルの修正が容易である一方、JavaScript などを用いて動的に変更されるウェブページのアクセシビリティ評価が難しいという問題があります。後者の[ブラウザ内の DOM]を評価する方式は、問題箇所のソースコード内の行番号などは指摘できなくなる一方で、JavaScript などを用いて動的に変更された結果についてもアクセシビリティ評価を行うことが出来るという利点があります。対象となるウェブページの種類や評価の目的に応じて、音声ユーザビリティビューのボタン、もしくは設定の[アクセシビリティ検証]設定パネルの[チェック対象]を切り替えて評価してください（図 32）。

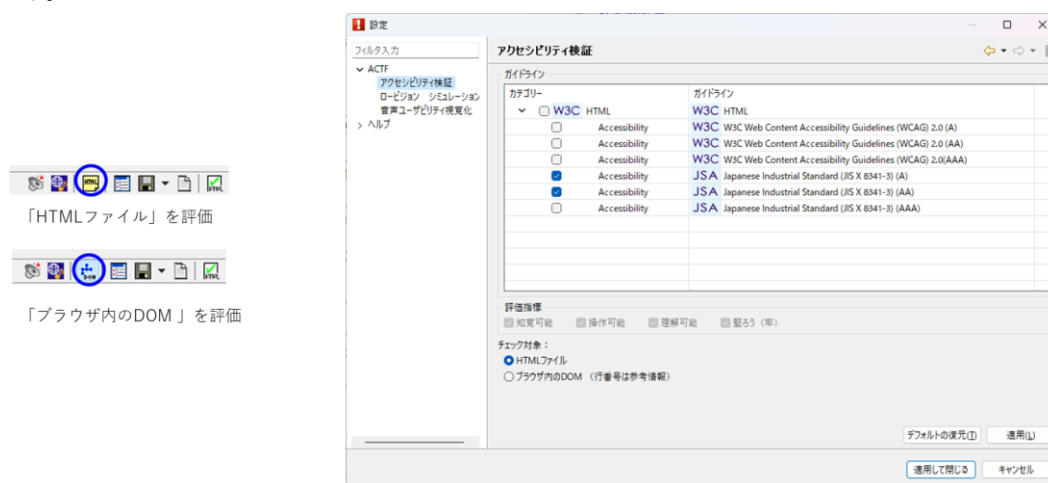


図 32 アクセシビリティ検証設定（音声ユーザビリティビューおよび設定パネル）

なお、miChecker 内で用いられているブラウザの機能上の制限から、評価対象ページの URL として指定したウェブページの HTML ファイルそのものを入手できず、miChecker のブラウザで表示されている内容に相当するアクセシビリティ評価が実施できない場合があります（ログイン管理が行われている場合や、一部の検索、情報入力などのフォーム入力操作を行った結果のページなど）。その場合には、[ブラウザ内の DOM]を評価する方式に切り替えてアクセシビリティ評価を実施してください。

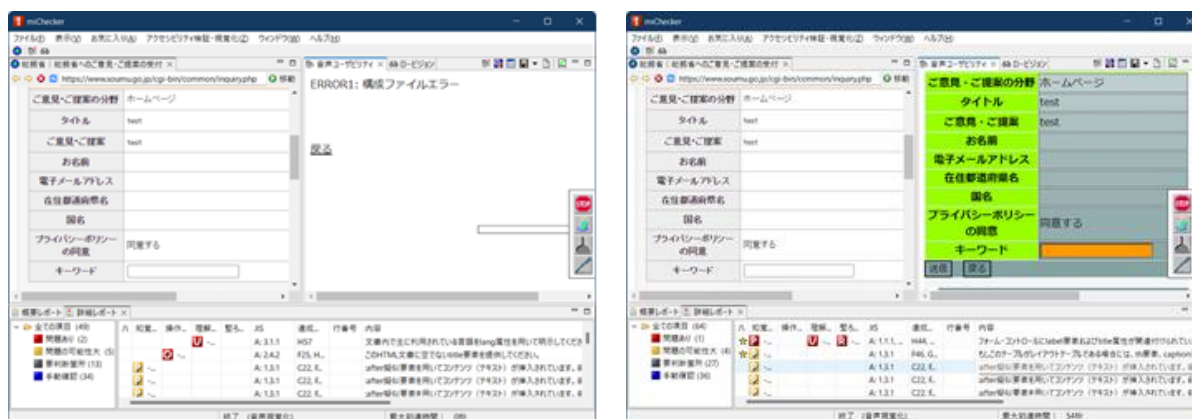


図 33 フォーム入力操作を行った結果のページで[HTML ファイル]を評価するとエラーとなり、[ブラウザ内の DOM]を対象とすると評価できる場合の例



また、これらのウェブページの HTML ファイルそのものを評価したい場合には、以下のような方法を用いることで評価が可能になる場合があります。

- 一般的なブラウザの HTML ファイルを保存する機能（「名前を付けて保存」など）を用いて保存した HTML ファイルを miChecker の「ブラウザ」で開いて評価を行ってください。miChecker の「ブラウザ」においても、コンテキストメニューを表示（右クリックもしくは Shift+F10）することで、表示中のページを「名前を付けて保存」することが可能です。なお、保存の際はファイルの種類として「Web ページ、HTML のみ (\*.html,\*.htm)」を選択するようにして下さい。
- 一般的なブラウザの HTML ファイルのソースを表示する機能（「ページのソースを表示」など）を用いて評価を行いたいウェブページのソースを表示し、表示されたソースをメモ帳などにコピー＆ペーストした上で HTML ファイル(\*.html,\*.htm)として保存します。最後に、保存した HTML ファイルを miChecker の「ブラウザ」で開いて評価を行ってください。

## 4.7 ブラウザのサイズを指定する

スマートフォンやタブレットなどのモバイル端末での閲覧や、レスポンシブデザインを採用した Web ページを対象とした評価を支援するため、代表的な画面サイズ（ブラウザのサイズ）での評価を可能とする機能を追加しました。miChecker の「表示 (V)」から「ブラウザのサイズ (W)」を選択することで利用できます (図 34)。各ブラウザでの開発者モードなどと合わせて活用してください。

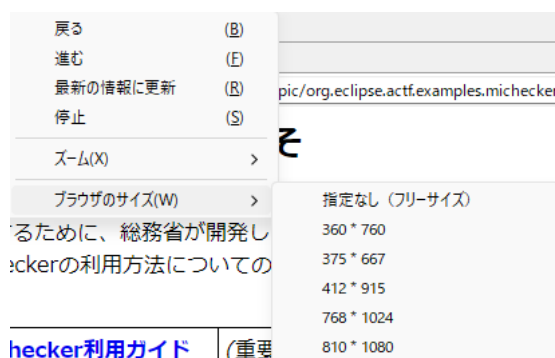


図 34 ブラウザのサイズの指定

## 4.8 オンラインヘルプを活用する

miChecker の詳細な使い方について調べるには、オンラインヘルプを活用してください。オンラインヘルプは、メニューバーにある「ヘルプ(H)」から「ヘルプ目次(H)」を選択することで利用できます (図 35)。オンラインヘルプでは、音声ユーザビリティ視覚化モードとロービジョンモードが持つパラメーターの意味やその設定方法、各モードの使い方などが説明されています。

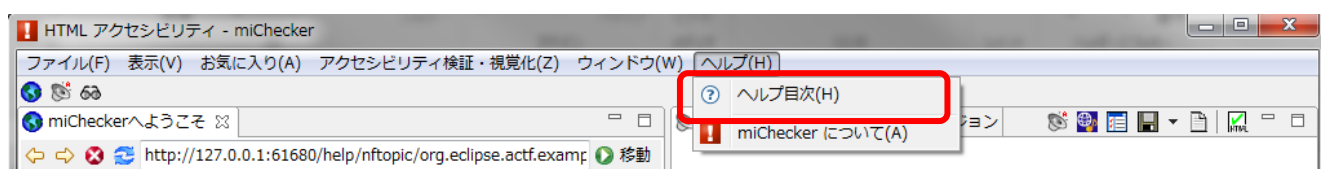


図 35 オンラインヘルプの起動

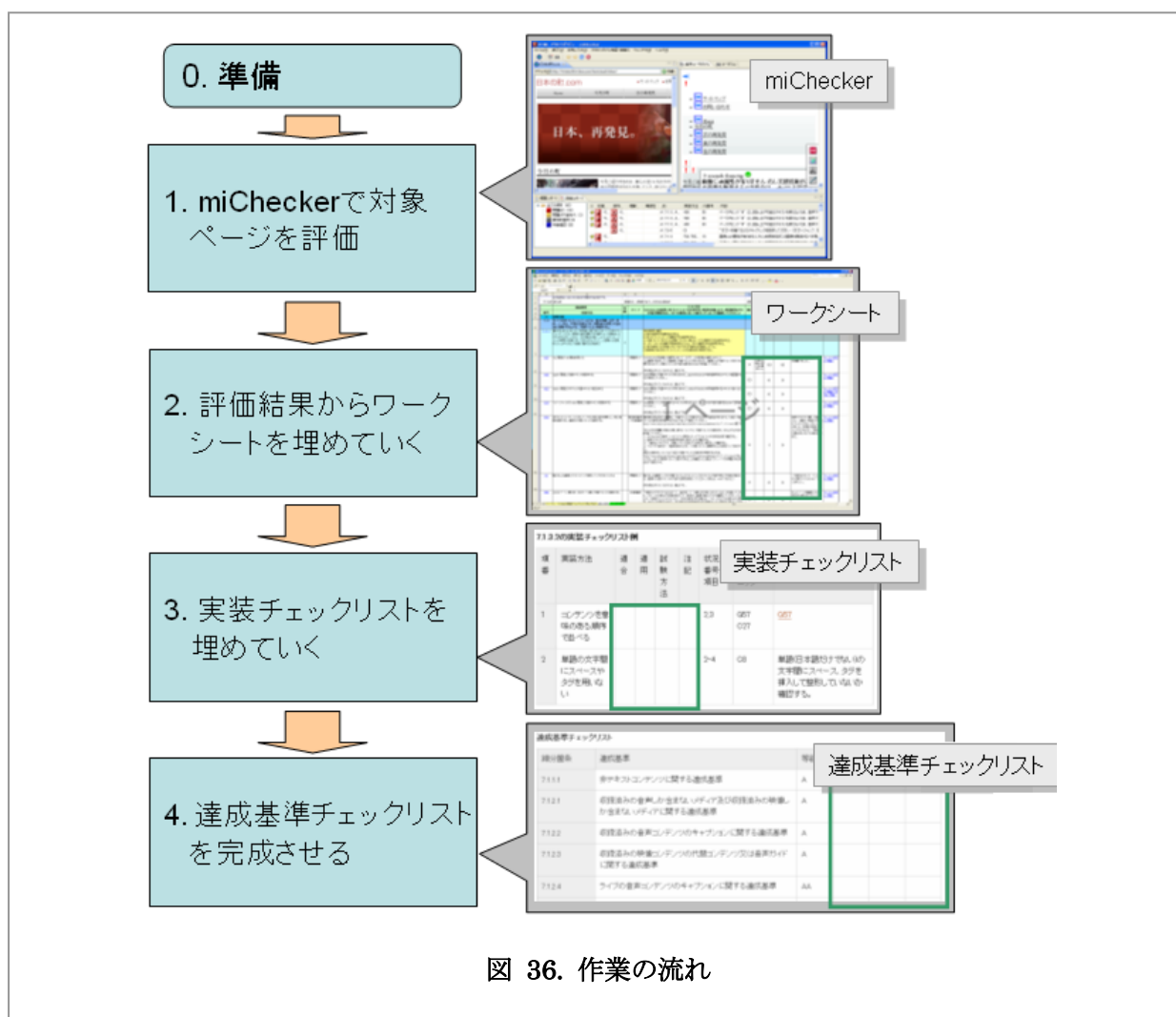


## 5. JIS X 8341-3:2016 における達成基準チェックリストの作り方

この章では、miChecker を活用して JIS X 8341-3:2016 の達成基準チェックリストを完成させる作業の概要を説明します。JIS X 8341-3:2016 における試験手順の詳細については JIS 規格の附属書 JB（参考）「試験方法」や、「みんなの公共サイト運用ガイドライン（2016 年版）」7.2. ウェブアクセシビリティの実現内容の確認（試験）と公開、および miChecker の付属文書「miChecker を用いた試験手順書」を参照してください。また、ウェブアクセシビリティ基盤委員会が提供する「JIS X 8341-3:2016 試験実施ガイドライン」（<https://waic.jp/docs/jis2016/test-guidelines/>）においても詳細な手順や各種チェックリストの例やカスタマイズ方法が紹介されているので、参考にしてください。

### 5.1 作業の流れ

1. miChecker で対象ページを評価します。
2. miChecker が出力した問題一つ一つについて、ワークシートを埋めつつ問題を検証していきます。同じ問題が複数ある場合は、その数も記入しておきます。
3. ワークシートの結果を、「達成方法及びその検証方法を特定できる技術的根拠」としてまとめます。ここでは、ウェブアクセシビリティ基盤委員会（WAIC）が提供している「実装チェックリストの例」に転記する方法を例にとって説明します。実装チェックリストでは、条件の組み合わせなどを考慮して達成基準としての合否を判定します。説明が必要な部分は注記に記入します。
4. 最後に、実装チェックリストの結果に基づいて達成基準チェックリストを完成させます。備考なども記入します。



## 5.2 詳細レポートの確認とワークシートへの記入例

「3.7 検証結果を保存する」に従って結果を保存することで、miChecker の検証結果を「ワークシート」に取り込むことができます。その手順を以下に説明します。

1. miChecker の起動画面からリンクされている「ワークシート」を保存しておきます。（「ワークシート」は miChecker の ZIP ファイル展開後の「2\_手順書」フォルダーにも保存されていますので、こちらからコピーすることも可能です。）

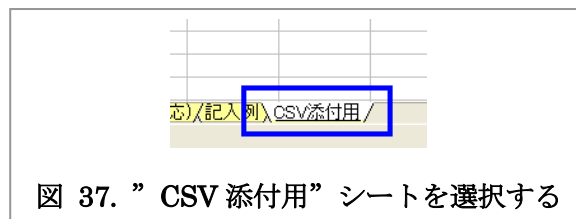


図 37. ” CSV 添付用” シートを選択する

2. miChecker で対象ウェブページの検証を行った後「結果を保存」機能を用いて検証結果を csv ファイルとして保存します。（図 16）
3. 1 で保存した「ワークシート」ファイルを開き、CSV”添付用”シートに切り替えます。（図 37）



図 38. 保存結果を開く

4. 続けて、保存結果の csv ファイルも開き、開いたらシート全体を全選択状態にします。（図 38）全選択にするには、Ctrl+a を押すか、左上のコーナーをクリックします。全選択状態にしたら、Ctrl+c を押してクリップボードにコピーします。
5. 再びワークシートに戻り、” CSV 添付用” シートを同様に全選択状態にして、そこにクリップボードにコピーしたものを貼り付けます。全選択には Ctrl+a、貼り付けには Ctrl+v を押すなどします。
6. 結果を確認します。ワークシートの” ワークシート (WAIC 実装チェックリスト例に対応)” シートを開いてください。miChecker 評価結果欄に結果が反映されています。図 39 では、「img 要素の alt 属性を用いる」という達成方法について、合計 13 個の問題があったことが示されています。

達成基準	適用レベル	タイプ	テスト方法 (miCheckerの結果に基づいてテスト方法を参考 に検証を実施します。検証箇所は多くの 場合複数存在します。)	適合	適用	該当 個数	miChecker 評価結果	特記事項	WAIC実装 チェックリスト 例-該当箇所
1 知覚可能 H37	達成方法								
img 要素の alt 属性を用いる	A	問題あり	img要素で表示している画像に代替テキストがありません。画像には代替テキストを付ける必要があります。代替テキストの内容の適切さはG94で評価してください。				13		1.1.1-状況A- 項番1

図 39. 結果が反映されていることを確認する

6. もし、”適用”のところに”-”あるいは不適用とあれば、その達成方法を評価する必要はありません。図 39 の例の場合は、空欄のままですからこの達成方法を適用するものとして、検証を進めます。
7. 検証する項目の”テスト方法”を読んでください。この例では、画像に対する代替テキストの有無を検証する項目であることが判ります。もし、代替テキストが無いのであれば、不適合と判断し、適合欄に”×”あるいは不適合と記載します。

8. 該当する全ての問題について検証してください。検証の結果、不適合だったものの数を数え” 該当個数” に記入します。

9. 何か特記事項があれば、それを特記事項欄に記入します。(図 40)

達成基準	達成方法	達成レベル	タイプ	テスト方法 (miCheckの結果に基づいてテスト方法を参考 に検証を実施します。検証箇所は多くの 場合複数存在します。)	適合	適用	該当 個数	miCheck 評価結果	特記事項	WAIO実装 チェックリスト 例:該当箇所
1	知覚可能 H37 img 要素の alt 属性を用いる	A	問題あり	img要素で表示している画像に代替テキストが ありません。画像には代替テキストを付ける必要 があります。代替テキストの内容の適切さはG94で 評価してください。	x		13	13	ページ内で用いられている写真に代 替テキストが存在していなかった	1.1-状況A- [番号]

図 40. ワークシートの該当する部分を埋めた例

10. ワークシートの全ての項目が埋まるまで6～9の作業を繰り返します。

## 5.3 実装チェックリストへの記入例

次に、ワークシートの内容を実装チェックリストに転記し、達成基準の可否を実装チェックリストの上で判断します。実装チェックリストの利用方法は、実装チェックリストの作成者や提供元の指示に従ってください。

1.3.2の実装チェックリスト例								
項 番	実装方法	適 合	適 用	試験 方法	注 記	状況- 番号- 項目	関連する 実装テク ニック	検証方法
1	コンテンツを意味 のある順序で並 べる					2,3	G57 C27	<a href="#">G57</a>
2	単語の文字間に スペースやタグを 用いない					2-4	C8	単語(日本語だけでなく)の文字間 にスペース、タグを挿入して整形し ていないか確認する。

図 41. 実装チェックリストの例

### ポイント

- 一つの達成方法に不適合が合ったとしても、条件や達成方法の組み合わせによって達成基準レベルでは適合となる場合があります。
- その達成基準の目的を良く理解することで、優先度の高いものを先に評価する等により、作業を効率的に進めることができます。
- ウェブサイト内で類似する構造のページが多数存在する場合には、それらのページの評価結果を参照することで、作業を効率的に進めることができる場合があります。
- ウェブアクセシビリティ基盤委員会が提供する JIS X 8341-3:2016 試験実施ガイドラインに実装チェックリストの例やカスタマイズ方法が紹介されているので、参考にしてください。

## 5.4 達成基準チェックリストの完成例

実装チェックリストが完成したら、達成基準チェックリストを完成させます。達成基準チェックリストの利用方法は、達成基準チェックリストの作成者や提供元の指示に従ってください。

達成基準チェックリスト					
細分箇条	達成基準	適合レベル	適用	適合	備考
1.1.1	非テキストコンテンツに関する達成基準	A			
1.2.1	収録済みの音声しか含まないメディア及び収録済みの映像しか含まないメディアに関する達成基準	A			
1.2.2	収録済みの音声コンテンツのキャプションに関する達成基準	A			
1.2.3	収録済みの映像コンテンツの代替コンテンツ又は音声ガイドに関する達成基準	A			
1.2.4	ライブの音声コンテンツのキャプションに関する達成基準	AA			

図 42. 達成基準チェックリストの例

## 6. miChecker のアクセシビリティ評価機能と CMS 等との連携

miChecker のアクセシビリティ評価機能は、CMS 等と連携して活用して頂くことも可能です。例えば、CMS を利用して新しいウェブページが作成された際や、ページの内容が変更された際に、miChecker のアクセシビリティ評価機能を呼び出してウェブページのチェックを行い、問題がある箇所や人手による確認が必要な箇所をコンテンツ作成者に伝えることなどにも利用できます。これらの機能を実装するためには、CMS 等の開発を担当されている方の協力が必要になります。miChecker の「2\_手順書」フォルダーに同梱されている「miChecker のアクセシビリティ評価機能と CMS 等との連携手順書」（CMS 等との連携手順書.pdf）を活用して、CMS 等の開発を担当されている方と詳細な仕様や開発方針などを検討するようにしてください。

## (参考) miChecker Ver. 2 から Ver.3 への変更に関する注意点

Internet Explorer 11 のサポート終了に伴い、miChecker 内で利用しているブラウザを Internet Explorer 相当のブラウザから、Microsoft Edge 相当のブラウザに変更しました。基本的な使い方や機能に大きな変更はありませんが、下記の点に注意してください。

チェック対象として、[HTML ファイル]を選択した場合であっても、miChecker のブラウザで表示中のページと同一の HTML ファイルを入手、評価できない場合があります。

- ログイン管理が行われている場合
- 一部の検索、情報入力などのフォーム入力操作を行った結果のページ
- アクセスするタイミングによってコンテンツが変更となるページなど

その場合には、[ブラウザ内の DOM]を評価する方式に切り替えてアクセシビリティ評価を実施してください。

また、これらのウェブページの HTML ファイルそのものを評価したい場合には、以下のような方法を用いることで評価が可能になる場合があります。

- 一般的なブラウザの HTML ファイルを保存する機能（「名前を付けて保存」など）を用いて保存した HTML ファイルを miChecker の「ブラウザ」で開いて評価を行ってください。miChecker の「ブラウザ」においても、コンテキストメニューを表示（右クリックもしくは Shift+F10）することで、表示中のページを「名前を付けて保存」することが可能です。なお、保存の際はファイルの種類として「Web ページ、HTML のみ (\*.html,\*.htm)」を選択するようにして下さい。
- 一般的なブラウザの HTML ファイルのソースを表示する機能（「ページのソースを表示」など）を用いて評価を行いたいウェブページのソースを表示し、表示されたソースをメモ帳などにコピー&ペーストした上で HTML ファイル(\*.html,\*.htm)として保存します。最後に、保存した HTML ファイルを miChecker の「ブラウザ」で開いて評価を行ってください。



## （参考）HTML 文書の堅ろう（牢）（Robust）性の確保

ウェブアクセシビリティを確保するためには、支援技術やブラウザなどが HTML 文書を確実に解釈できるよう、十分に堅ろうである（正しく記載されている）必要があります。miChecker でアクセシビリティの評価・検証を開始する前に、W3C の Markup Validation Service（HTML 文書が正しく記載されているか検証するサービス、<https://validator.w3.org/>）などを用いて、HTML 文書の堅ろう性を確保することが重要です。

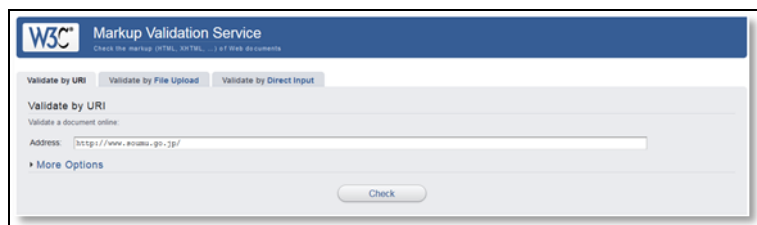


図 43. W3C の Markup Validation Service

HTML 文書の堅ろう性の確保は、本来、アクセシビリティ評価の前段として行っておくことが望ましい項目ですが、miChecker の利用時にあわせて検証を行うことも可能です。ここでは、HTML 文書の堅ろう性について、miChecker で開いているウェブページの URL に基づいて検証を行う場合と、ローカルに保存されたページ (html ファイル) を用いて検証する場合の 2 通りの方法を紹介します。いずれの方法も miChecker の「HTML バリデーション」ボタン（図 44）を利用して、上記の W3C の Markup Validation Service を呼び出して検証を行います。

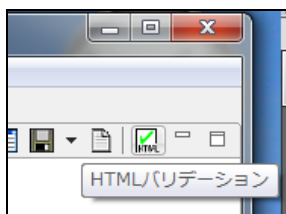


図 44. miChecker の HTML バリデーションボタン

初めて「HTML バリデーション」ボタンを選択した際は、W3C の Markup Validation Service の利用条件などに関する情報がブラウザに表示されると共に、図 45 の様な確認ダイアログが表示されます。利用条件等を確認の上で、「OK」ボタンを選択することで、miChecker から W3C の Markup Validation Service を呼び出すことが可能になります（次回以降は、この確認はスキップされます）。

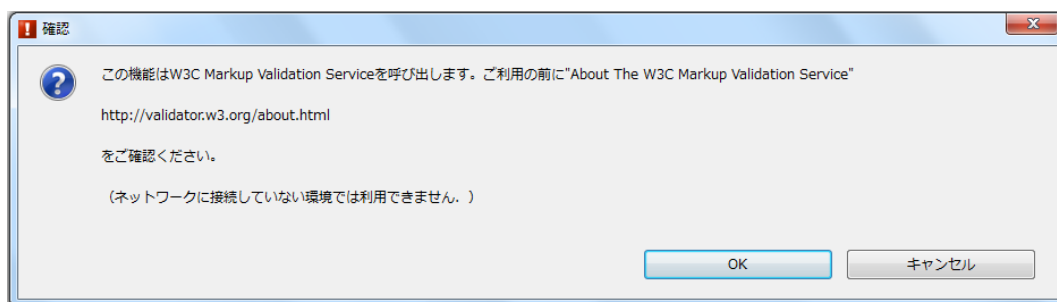


図 45. 確認ダイアログの例

#### A) ウェブページの URL に基づいて堅ろう性を検証する場合

評価対象のウェブページの URL を miChecker のアドレスバーに入力し、「移動」ボタンを選択して該当ページを開いた後、画面右上の「HTML バリデーション」ボタンを選択します（図 44）。

評価対象のウェブページが一般に公開されているウェブページであった場合、システムの既定のブラウザに以下の例の様な検証結果が表示されます。この例では、125 件のエラーと 127 件の警告が検証結果として出力されています（図 46、図 47）。

The screenshot shows the W3C Markup Validation Service interface. At the top, it says "W3C® Markup Validation Service" and "Check the markup (HTML, XHTML, ...) of Web documents". Below this, there's a "Jump To:" section with "Validation Output" selected. The main section is titled "Errors found while checking this document as XHTML 1.0 Transitional!". It shows a "Result:" of "125 Errors, 127 warning(s)". Below this, there's a table with fields: "Address:" (with a redacted URL), "Encoding:" (utf-8), "Doctype:" (XHTML 1.0 Transitional), "Root Element:" (html), and "Root Namespace:" (http://www.w3.org/1999/xhtml).

Errors found while checking this document as XHTML 1.0 Transitional!	
Result:	125 Errors, 127 warning(s)
Address :	[Redacted URL]
Encoding :	utf-8 (detect automatically)
Doctype :	XHTML 1.0 Transitional (detect automatically)
Root Element:	html
Root Namespace:	<a href="http://www.w3.org/1999/xhtml">http://www.w3.org/1999/xhtml</a>

図 46. 検証結果サマリーの例

The screenshot shows the "Validation Output: 125 Errors" section. It displays the first error: "Line 5, Column 54: end tag for 'meta' omitted, but OMITTAG NO was specified". The error message includes the code snippet: `<meta http-equiv="X-UA-Compatible" content="IE=edge" >`. Below the code, it explains: "You may have neglected to close an element, or perhaps you meant to 'self-close' an element, that is, ending it with '>' instead of '>'." The second error is: "Line 16, Column 16: there is no attribute 'property'". The error message includes the code snippet: `<meta property="fb:app_id" content="137119773101625" />`. Below the code, it explains: "You have used the attribute named above in your document, but the document type you are using does not support that attribute for this element. This error is often caused by incorrect use of the 'Strict' document type with a document that uses frames (e.g. you must use the 'Transitional' document type to get the 'target' attribute), or by using vendor proprietary extensions such as 'marginheight' (this is usually fixed by using CSS to achieve the desired effect instead). This error may also result if the element itself is not supported in the document type you are using, as an undefined element will have no supported attributes; in this case, see the element-undefined error message for further information."

**Validation Output: 125 Errors**

✖ Line 5, Column 54: end tag for "meta" omitted, but OMITTAG NO was specified

`<meta http-equiv="X-UA-Compatible" content="IE=edge" >`

You may have neglected to close an element, or perhaps you meant to "self-close" an element, that is, ending it with ">" instead of ">".

⬆ Line 5, Column 1: start tag was here

`< meta http-equiv="X-UA-Compatible" content="IE=edge">`

✖ Line 16, Column 16: there is no attribute "property"

`<meta property="fb:app_id" content="137119773101625" />`

You have used the attribute named above in your document, but the document type you are using does not support that attribute for this element. This error is often caused by incorrect use of the "Strict" document type with a document that uses frames (e.g. you must use the "Transitional" document type to get the "target" attribute), or by using vendor proprietary extensions such as "marginheight" (this is usually fixed by using CSS to achieve the desired effect instead).

This error may also result if the element itself is not supported in the document type you are using, as an undefined element will have no supported attributes; in this case, see the element-undefined error message for further information.

図 47. サマリー下部に表示されるエラーの詳細の例

## B) HTML ファイルを指定して堅ろう性を検証する場合

一般に公開されていないウェブサイトが存在するウェブページや、PC に保存された HTML ファイルなどを対象に miChecker の「HTML バリデーション」(図 40) を実行すると、「W3C Markup Validation Service のページからファイルをサブミットして下さい。」というダイアログが表示され、W3C の「Validate by File Upload」のページへ移行します。

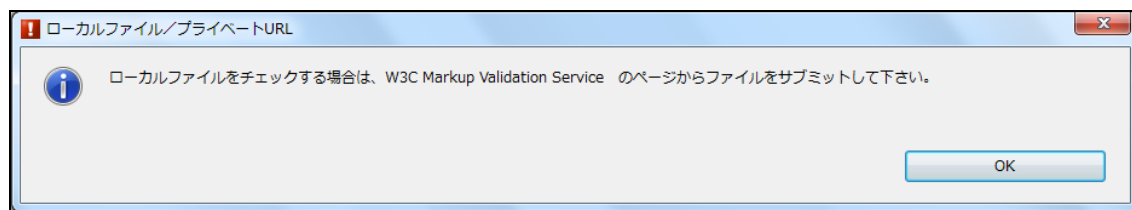


図 48. ローカルファイル/プライベート URL を指定した場合に表示されるメッセージ

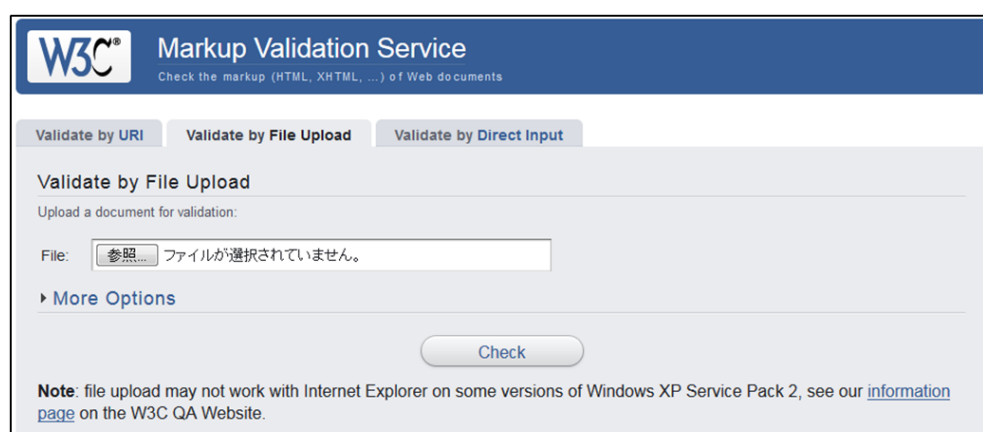


図 49. W3C の Validate by File Upload より検証対象のファイルを選択

ここで「参照」ボタンを選択して検証対象となる HTML ファイルを指定した後、「Check」ボタンを選択すると、A) の手順の場合と同様に検証結果サマリーとエラー詳細が表示されます。

## C) 検証結果の確認と修正

W3C の Markup Validation Service を用いた検証でエラーが発見された場合は、HTML ファイルの修正を行います。表示されたエラーおよび説明文 (英語) を確認しながらウェブページの修正を行い、再度検証という手順を繰り返すことで、エラーを減らしていきます。最終的に、エラーがなくなれば作業は完了です。

**「みんなのアクセシビリティ評価ツール miChecker」について**

「みんなのアクセシビリティ評価ツール miChecker」（以下、miChecker）は、JIS X 8341-3:2016（高齢者・障害者等配慮設計指針—情報通信における機器，ソフトウェア及びサービス—第3部：ウェブコンテンツ）に基づくウェブアクセシビリティ対応の取組を支援するために、総務省が開発し、提供するアクセシビリティ評価ツールです。

**【総務省担当部署】**

情報流通行政局 情報流通振興課 情報活用支援室

電話： 03-5253-5685

FAX： 03-5253-5752