

# Webアクセシビリティの基礎と 2010年JIS改定のポイント

株式会社ビジネス・アーキテクツ  
太田 良典

# アクセシビリティの基礎

# アクセシビリティとは？

- アクセシビリティ - Accessibility
  - Accessible = Access + able
  - アクセスできる、アクセスが可能
  - さまざまな利用者がアクセス可能であること
    - 主に障害のある利用者や高齢者
- 類義語
  - ユーザビリティ
  - ユニバーサルデザイン (UD)

# vs. ユーザビリティ

- ユーザビリティ
  - 想定された利用者にとって「使いやすい」
    - 想定の間幅によってはアクセシビリティと重なる
- アクセシビリティ
  - さまざまな利用者が「使える」
    - 使いやすさは必ずしも問われない
  - 複数のアプローチ(代替)を用意することも
    - メインの利用者には使いにくい、使えない手段も
    - 例: 駅の階段についている車椅子用リフト

# vs. ユニバーサルデザイン

- Universal = 普遍的な
  - 特定の人のためのデザインではない、普遍的なデザイン
- ある一つのものが普遍的に使える
  - さまざまな環境の人が同じものを使える
  - 例: 右利きでも左利きでも使えるハサミ
- 「アイデアというのは、複数の問題を一気に解決するものである」(宮本茂)

# 例: 駅のホームへのアクセス

- 階段
  - 足腰弱い人は大変、車椅子では利用できない
- エスカレーター
  - ユーザビリティ向上、しかし車椅子では?
- 階段に車椅子用のリフト
  - アクセシビリティ向上、ユーザビリティは?
- エレベーター
  - 健常者でも車椅子でも使いやすい

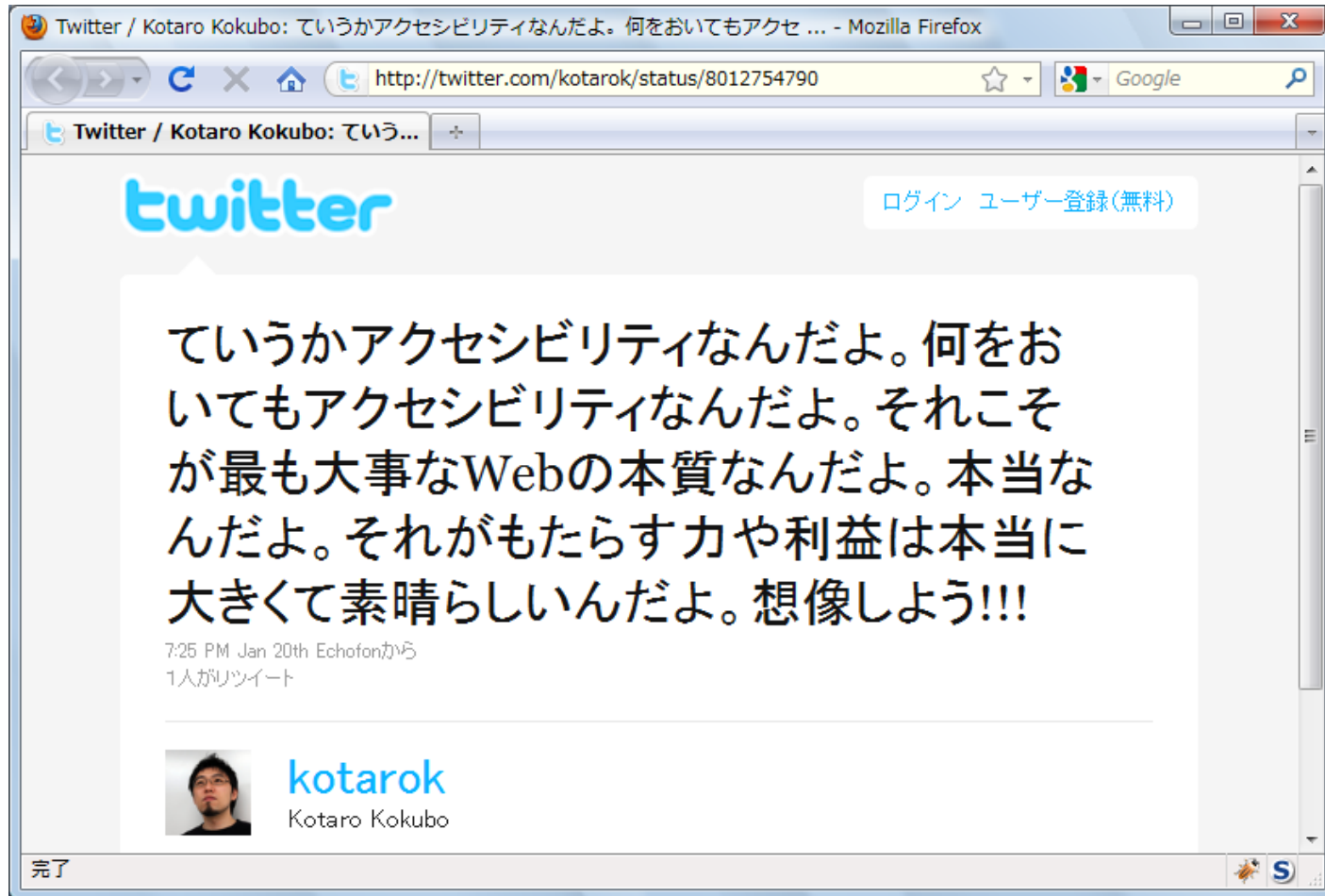
# ウェブサイトで言うと？

- 「目の見えない方用のサイト」は？
  - アクセシビリティ的にはOK
  - ユニバーサルデザインではない
- ウェブでは、ユニバーサルデザインを目指すことが一般的

# ウェブアクセシビリティ入門



# とあるつぶやき



# アクセシビリティなんだよ!

<http://twitter.com/kotarok/status/8012754790>

- ていうかアクセシビリティなんだよ。何をおいてもアクセシビリティなんだよ。
- それこそが最も大事なWebの本質なんだよ。本当なんだよ。
- それがもたらす力や利益は本当に大きくて素晴らしいんだよ。想像しよう!!!

# 言い換えると

- ウェブと他のメディアの決定的な違いはアクセシビリティ
- ウェブは本質的にアクセシブル
  - 圧倒的なアクセシビリティを誇る
  - ユニバーサルデザインも実現しやすい
- ウェブのアクセシビリティは人に、社会に大きな利益をもたらす

# 圧倒的なアクセシビリティ

- ウェブは圧倒的にアクセシブル
  - 車椅子の人が気軽に買い物に行ける？
  - 全盲の人が気軽に新聞を読める？
- ウェブなら、さまざまな特性の人が自由に利用できる
  - 時間と場所の制約もない
  - 誰もが社会に参加できるチャンス

# 利用者のさまざまな特性

# 利用者の特性

- 利用者自身の特性 (いわゆる障害)
  - 視覚障害 (見えない・見えにくい)
  - 聴覚障害 (聞こえない・聞こえにくい)
  - 学習障害、認知障害 (文章を理解しにくい)
  - 運動障害 (手が動かない・動かすにくい)
  - 光過敏 (激しい光の点滅で気分が悪くなる)
- 加齢による能力低下

# 一時的な特性、環境による特性

- 一時的な特性
  - － 目の病気、眼鏡を忘れた
  - － 手に怪我をした
- 環境による特性
  - － モノクロプリンタで印刷した
  - － マウスが壊れた
  - － 食べながら片手で

# 実際にあった例

- モニタを変えた
- なぜか16色表示になってしまった
- ドライバをインストールすれば直る?
- 最新のドライバをダウンロードしたい
- メーカーのサイトにアクセスした
- メーカーのサイトは画像だらけ.....
  - モニタは16色表示の状態
  - 細かい色の違いが分からない



# さまざまな支援技術

# 支援技術とは

- 利用者の特性に合わせて使われる技術
  - 多くはブラウザ+ で構成される
- 例:
  - スクリーンリーダー (読み上げ)
  - 点字ディスプレイ、点字プリンタ
  - 視線入力、専用のポインティングデバイス
  - 専用のキーボード

# 支援技術による自由な出力

- 利用者の好きな形で見られる
  - 好きなブラウザを使って良い
- 出力形式自体もさまざま
  - 画面で見る、印刷して紙で読む
  - 音声で聴く
  - 点字で出力
- マッシュアップによるカスタマイズも

# マッシュアップとは？

- 複数のWebサービスのAPIを組み合わせて別のサービスを作ること
- APIが提供されていない場合は、Webページの内容を読んで解析して利用することもある
  - 「スクレイピング」と呼ばれる

# マッシュアップの例



# カーリル

- <http://calil.jp/>
- 各図書館の蔵書データを読み取り、  
各地図書館の横断検索を可能にしている
- より使いやすいインターフェイスを提供
  - 図書館のサイトはたいてい使いにくい

# なぜマッシュアップが可能か？

- APIが提供されていればベストだが.....
  - 最悪、テキストデータさえあれば良い
  - テキストがあればスクレイピング可能
- 図書館がウェブサイトを持ち、  
蔵書データを公開しているからこそ可能

ウェブサイトが存在するだけでも  
さまざまな可能性を秘めている

# ウェブがアクセシブルな理由

- ウェブのデータの多くはテキスト
- テキストは処理しやすい
  - 利用者の特性に合わせた処理が可能

プログラムで処理できることが重要

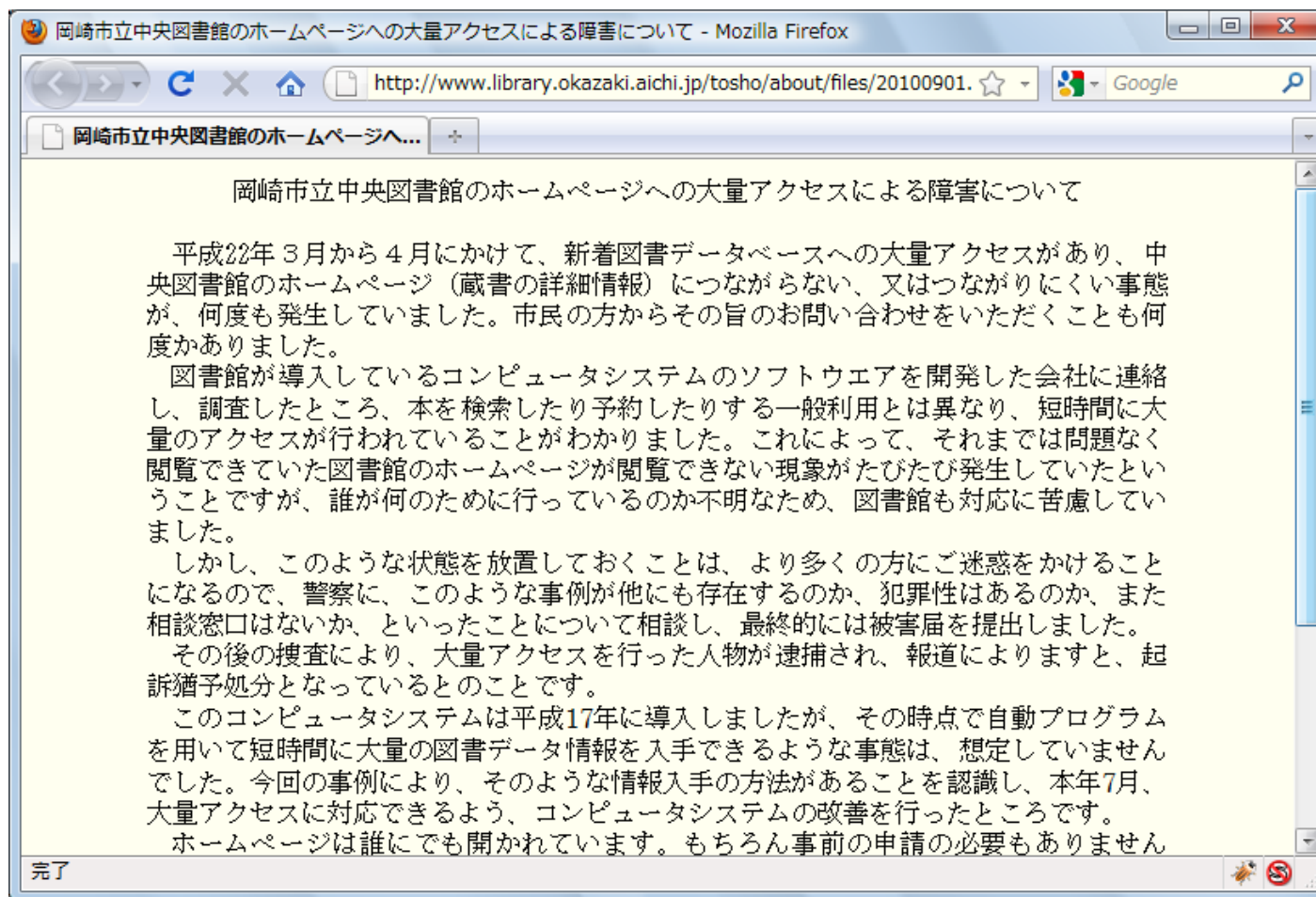
machine readable



なぜウェブは  
アクセシブルでないのか

# ある日の岡崎市立中央図書館

<http://www.library.okazaki.aichi.jp/tosho/about/files/20100901.html>



# 一部抜粋

- 調査したところ、本を検索したり予約したりする一般利用とは異なり、短時間に大量のアクセスが行われていることがわかりました
- このコンピュータシステムは平成17年に導入しましたが、その時点で自動プログラムを用いて短時間に大量の図書データ情報を入手できるような事態は、想定していませんでした

# 想定していない?

- 「一般利用とは異なる」
- 「このような利用法は想定していない」
- 支援技術の利用者にも同じことを言う?
- 支援技術によるアクセスを想定しないと問題が起きる場合がある

# 例: 画像、映像、音声

- 画像や映像、音声を解析して自動で適切な字幕をつける技術はない
  - 今はまだ使い物になるレベルではない
- 利用者側の努力ではアクセスできない
  - コンテンツ側の対応が必要になる
  - コンテンツ側が「想定していない」と、どうしようもない

# コンテンツ側の対応の例

- 画像に代替テキストをつける
- 映像に字幕をつける
- 画像や映像が見えなくても内容を理解できるように作る
  - 特に図やグラフなど

これらを怠ると、  
アクセスできない利用者が出てくる

# 基本はテキスト

- テキストなら支援技術で処理できる
  - ただし、アスキーアートは無理
- テキストでないものには代替を用意
  - 要素をテキストに置き換えたときに意味が通じるようにする
  - 代替テキストの用意の仕方は実装依存
    - HTMLならimg要素のalt属性

# 代替テキストあれこれ

- 画像の説明ではない
- 何が適切かは場合によって異なる
  - 画像化文字の場合
  - 写真の場合
  - 写真 + キャプションの場合
  - 本文で十分な説明がある場合
- 画像をテキストに置き換えて読んだ時、内容が理解でき、冗長でないこと



# HTML4.01の無念

- img要素のlongdesc属性
  - 画像に長い説明をつけることができる
    - 説明ページのURIを記述する
  - が、ほとんどのブラウザがサポートせず
    - アクセシビリティ・サポーテッドでない

# アクセシビリティ・サポーテッド

- アクセシビリティに関する機能が、ブラウザや支援技術によって実際にサポートされていること
- 仕様では定義されていても、実装されていないケースがある
  - img要素のlongdesc属性
  - td要素のaxis属性
- 理論上はうまく行くはずが.....

# 音声用スタイルの罫

```
@media screen{  
  div.for-speech{  
    display: none;  
  }  
}  
  
@media speech{  
  div.for-speech{  
    display: block;  
  }  
}
```

- CSS2 media types
- div.for-speech は読み上げ環境で表示されることを期待
- しかし.....

# スクリーンリーダーの実際

- 多くのスクリーンリーダーは  
自身でHTMLやCSSを解釈しない  
– 普通のブラウザ (IEやFirefoxなど) が  
表示した結果を読み上げる
- IEやFirefoxは@media screenを解釈し、  
@media speech を無視
- .for-speech の内容はスクリーンリーダーでは  
読み上げられない

# コンテンツだけでは成立しない

- コンテンツを読み取って利用者に伝えるユーザエージェント(ブラウザ)があってはじめて成立する
  - 例: コンテンツをキーボードで操作可能に  
そもそもブラウザがキーボードに対応していなければならない
- コンテンツがいくら配慮しても、ブラウザが対応していなければ無意味

制作者が知っておくべきこと

# 制作者が知っておくべきこと

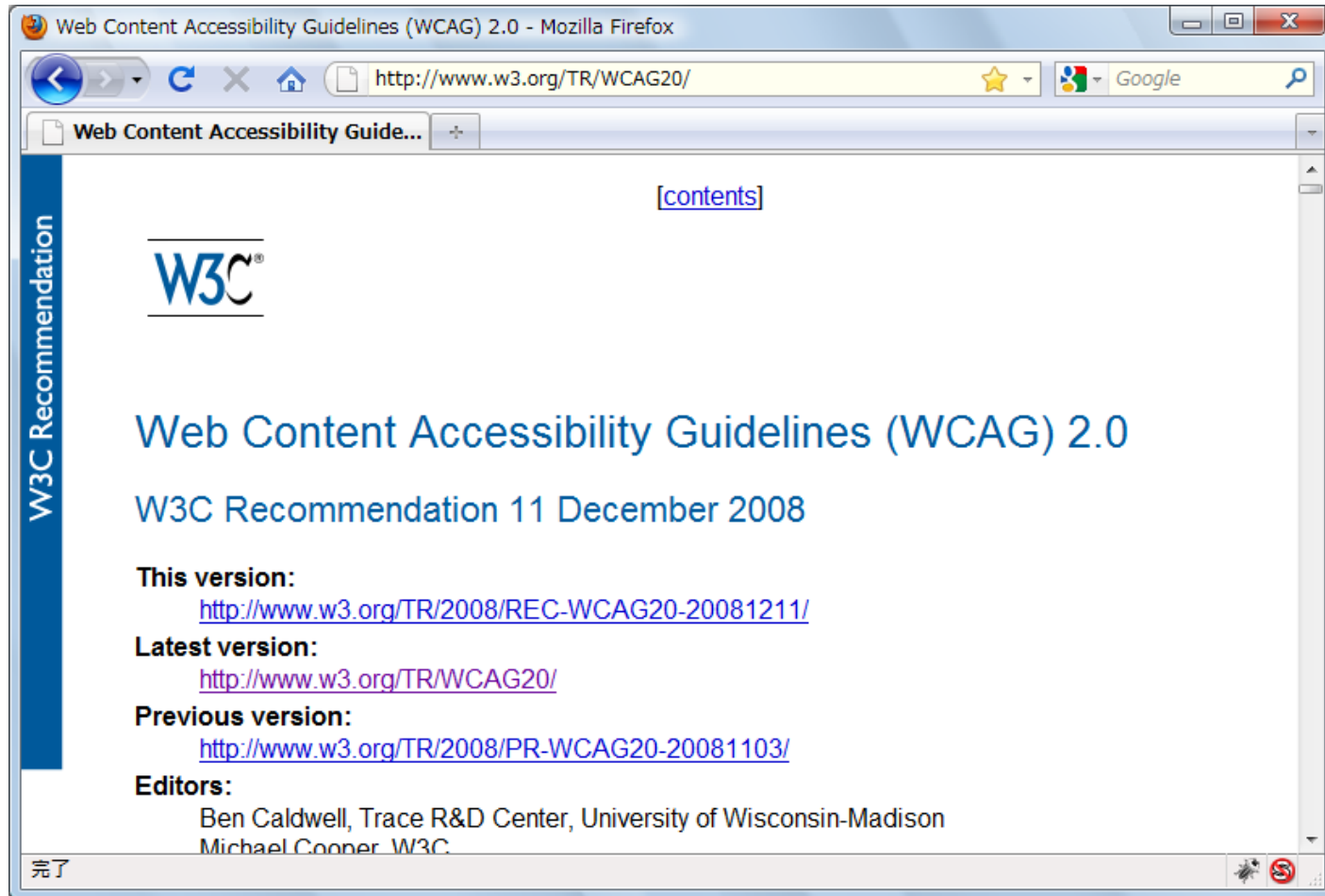
- 利用者のさまざまな特性
- さまざまな支援技術とその挙動
- さまざまな実装方法
  - 適切な実装方法を選択する必要がある
  - 論理的に正しいだけでなく、実際に機能する実装方法を知る必要がある
- 知らなければならないことは膨大

# アクセシビリティガイドライン

- アクセシビリティを確保するために、ウェブのコンテンツ側でやるべきことを定めたもの
- W3C の WCAG
  - Web Content Accessibility Guidelines
- 日本規格協会の JIS X 8341-3
- 各企業が独自に定めるもの
  - ただし意味合いが若干違う



# WCAG2.0



# JIS X 8341-3 と 2010年版改正のポイント

# JIS X 8341-3

- 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
- 8341 = やさしい
- 初版は2004年に制定
  - JIS X 8341-3:2004
- 2010年8月20日に改正
  - JIS X 8341-3:2010

# JIS X 8341-3:2004の特徴

- WCAG1.0をベースに、独自の項目を追加
  - 米国リハビリテーション法508条なども参考
  - 日本独自の内容も追加
- 項目のレベルは2段階
  - 必須(.....ねばならない)
  - 推奨(.....望ましい)
  - WCAG1.0では priority1 ~ 3の3段階だった
- 具体的な実装方法の例も解説

# その後の状況

- 新デバイス、新技術の登場
  - モバイル端末、タッチパネルなども普及
  - HTML / CSS以外の技術も普及
    - Flashやスクリプト等を用いた RIA の増加
      - RIA = Rich Internet Application
    - RSSやJSON等、HTMLでないデータも増加
- WCAG1.0 の問題点も明らかに
  - HTML前提の規定、変化に対応しにくい
  - 不明瞭な基準が多い

# WCAG2.0では?

- HTML / CSSにこだわらない  
汎用的に適用できる基準
  - tableやframeなど名指しの規定が無くなる
- 一部基準が数値化され、判断可能に
  - 十分なコントラスト     4.5:1のコントラスト

# 2010年版はWCAG2.0に対応

- JIS X 8341-3 も WCAG2.0 に対応
  - 達成基準を WCAG2.0 に対応させた
    - 完全な一対一対応、事実上同一の規定
- 国内向けに別の施策を実施する必要がなくなった

# 2010年版改正の概要



# 2010年改正のポイント

- 1. 達成等級の概念が追加
- 2. 達成基準の変更
- 3. 企画・制作プロセスの具体化
- 4. 検証・試験プロセスの具体化

# 1. 達成等級の概念が追加

- 2004年版では、各達成基準が必須、推奨のいずれかとなっていた
- 2010年版では「A」「AA」「AAA」の3段階の「達成等級」の概念が導入された
  - A: 守らないと全くアクセスできない場合がある
  - AA: 守らないとアクセスしにくい場合がある
  - AAA: 守ると幸せになる人がいる

## 2. 達成基準の変更 (その1)

- WCAG2.0 と同等の達成基準を採用
- AやAAの達成基準については、  
2004年版からの大幅な変更はない
  - 2004年版とは分類が大きく異なるが、考え方が大きく変わった点はない
  - 2004年版に対応しているコンテンツは、追加の対応がなくても A や AA の多くを達成可能
- AAAでは厳しいものが追加されている
  - 全てのサイトで達成可能というわけではない

## 2. 達成基準の変更 (その2)

- 基準が明確化(数値化)されたものがある
  - 2004年版の例:
    - 5.6 c) フォントの色には、背景色などを考慮し見やすい色を指定することが望ましい
  - 2010年版では:
    - 7.1.4.3 最低限のコントラストに関する達成基準: 少なくとも4.5:1 のコントラスト比がなければならない
    - 7.1.4.6 より十分なコントラストに関する達成基準: 少なくとも7:1 のコントラスト比がなければならない

## 2. 達成基準の変更 (その3)

- 内容が一般化・抽象化されたものがある
  - 2004年版の例:
    - 5.2 d) 表組みの要素をレイアウトのために使わないことが望ましい
      - 「表組みの要素」はHTMLのtable要素を指しており、HTMLに限定された達成基準
  - 2010年版では:
    - 7.1.3.2 意味のある順序
    - 7.2.4.3 フォーカス順序に関する達成基準
      - HTMLを使用する・しないに限らず、読み上げ順序やフォーカス移動順に問題がなければ良い

### 3. 企画・制作プロセスの具体化

- 必要なプロセスや文書が規定された
  - 2004年版:「配慮する」という曖昧な定義
    - 4.1.a) ウェブコンテンツを企画・制作するときに、可能な限り高齢者・障害者が操作又は利用できるように配慮する
    - 4.1.c) ウェブコンテンツの企画から運用に至るプロセスで情報アクセシビリティを常に確保し、さらに向上するように配慮する
  - 2010年版:
    - 6.1、6.2をまるごとプロセスの説明に充てる

## 4. 検証・試験プロセスの具体化

- 試験のプロセスがかなり明確にされた
  - 2004年版: 「6.3 ウェブコンテンツがこの規格の要件を満たしていることを検証しなければならない」
    - 参考や例が掲げられているが、具体的にこれを実行しなければならないという規定ではない
- 2010年版: 箇条8.2で具体的な手順
- 適合性評価、自己適合宣言にも言及
  - 「適合」などの表示のガイドラインもある

# 2010年版改正の影響



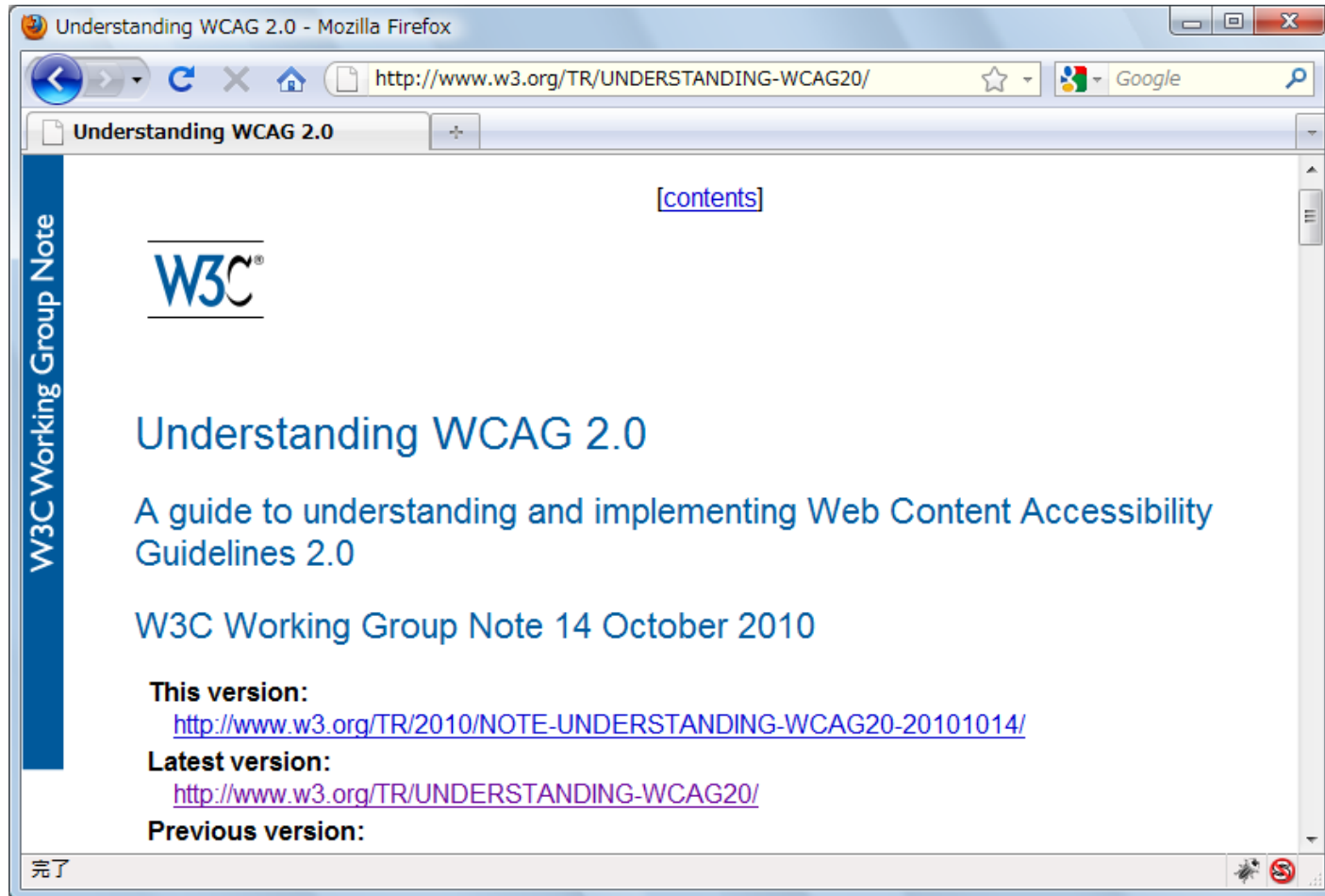
# 一般化の影響 (その1)

- PDFやFlashなどのコンテンツも、JISの達成基準を満たすことが求められる
  - HTMLコンテンツと同じ達成基準を適用することが可能であり、達成が求められる
- PDFやFlashにうまく適用できなかった欠点が改められた、と見るべき

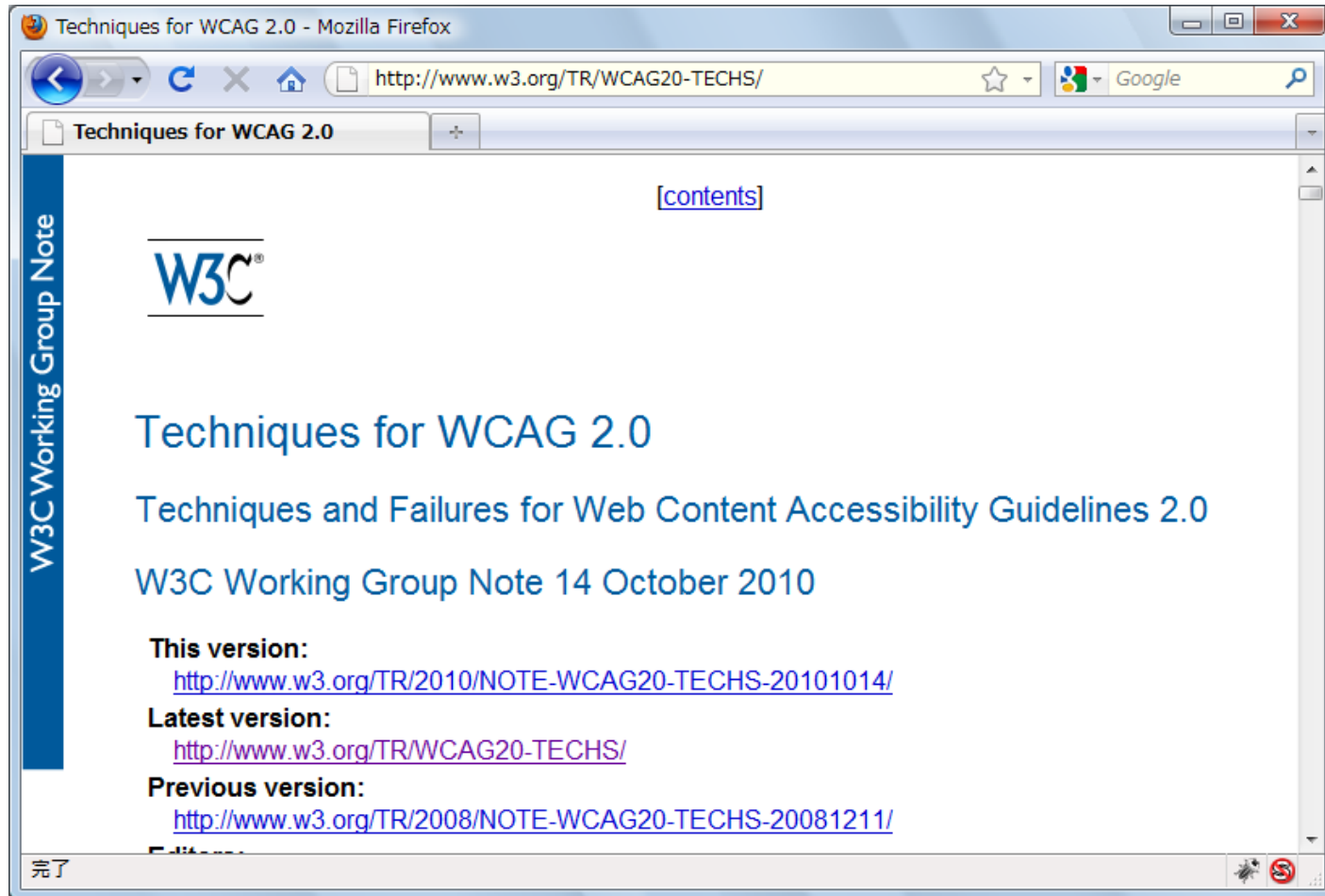
# 一般化の影響 (その2)

- わかりにくくなった
  - 達成基準を読んでも理解できない
    - 一般的に、一般化された内容は理解しにくい
- 規格だけ読んでも分からないため、  
関連する解説文書を読み込む必要がある
  - Understanding WCAG 2.0
    - <http://www.w3.org/TR/UNDERSTANDING-WCAG20/>
  - Techniques for WCAG 2.0
    - <http://www.w3.org/TR/WCAG20-TECHS/>

# Understanding WCAG 2.0



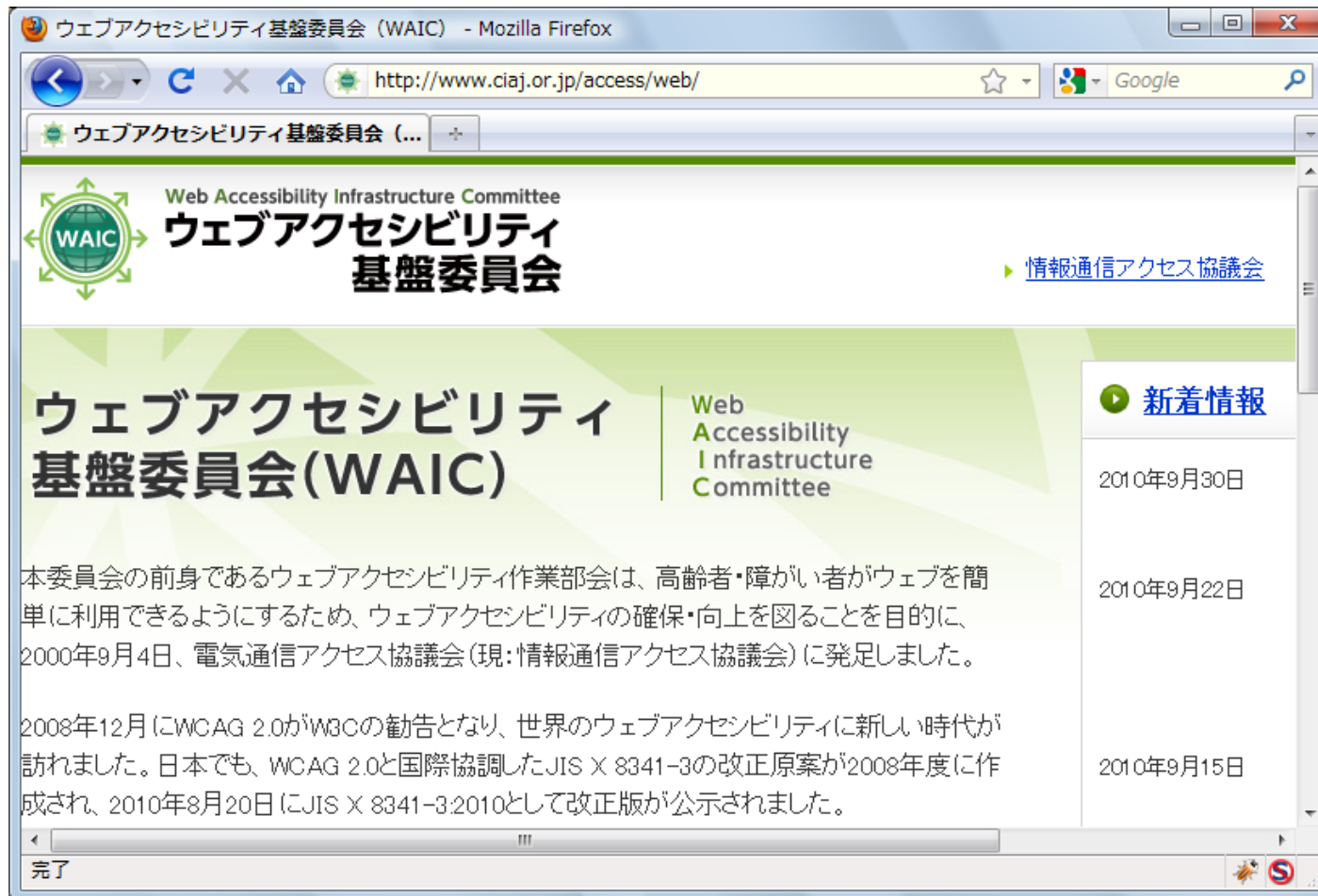
# Techniques for WCAG 2.0



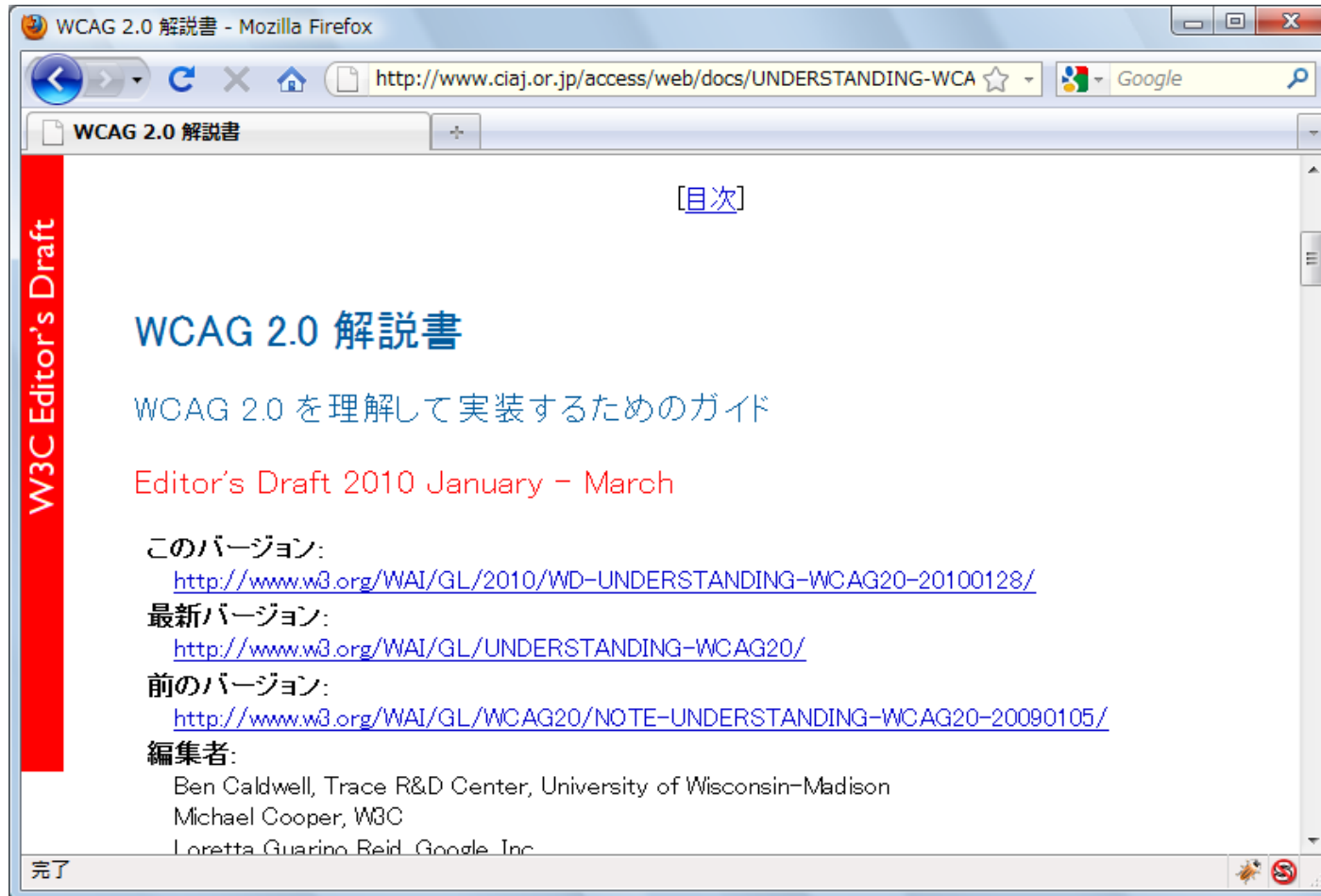
# ウェブアクセシビリティ基盤委員会 (WAIC) による和訳

- WCAG 2.0 解説書
  - <http://www.ciaj.or.jp/access/web/docs/UNDERSTANDING-WCAG20/Overview.html>
- WCAG 2.0 実装方法集
  - <http://www.ciaj.or.jp/access/web/docs/WCAG-TECHS/Overview.html>

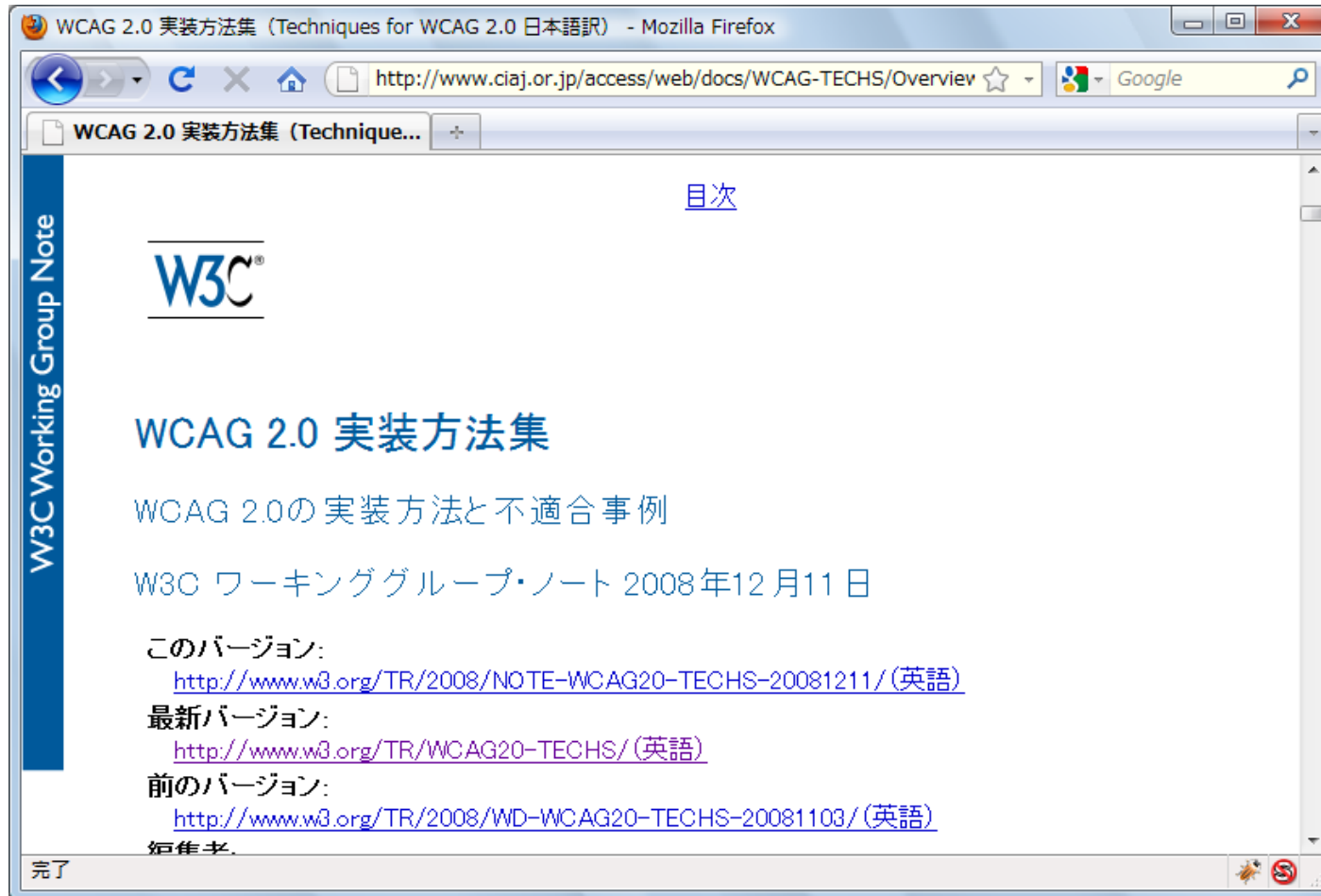
# ウェブアクセシビリティ 基盤委員会(WAIC)



# WCAG2.0解説書



# WCAG 2.0 実装方法集





# 基準は厳しくなった?

- 厳しくなったとも言える
  - HTML以外の技術を活用していると厳しい
  - AAAを前提にすると、非常に厳しい
    - ライブ配信の映像に字幕をつけることは可能?
    - つけないとアクセスできないのは事実なので妥当ではあるのだが、実行するのは大変
- 変わっていないとも言える
  - テキスト中心のサイトならAの達成は容易
  - 昔から取り組んでいれば影響は少ない

# 適合などの表示

- 「適合」などの表示を行うルールが存在
  - ウェブコンテンツの JIS X 8341-3:2010 対応度表記ガイドライン 第1版 - 2010年8月20日
    - <http://www.ciaj.or.jp/access/web/docs/jis2010-compliance-guidelines/>
  - 表示を目指すなら取り組みが必要だが.....

# 試験可能になった?

- すべてを機械的に試験することは困難
  - alt属性がない場合は指摘できる
  - alt属性の値が適切でない場合は指摘できない
- 事実上検証不可能な項目も存在する
  - 7.3.1.5 読解レベルに関する達成基準
- 機械試験の通過を強制すると.....?

# 改正のポイントまとめ

- 達成基準の一般化、汎用化
  - わかりにくい
    - 例もなく、規格の内容が非常に分かりにくい
    - 関連文書を大量に読むべし
- 「準拠」などの表示のルールができた
  - 基本的には試験結果の表示が求められる
- ほかは以前から普通にやっていたこと
  - その「普通」がなかなかできていない場合も

# わかりにくい達成基準の例

# JIS X 8341-3:2010 7.1.4.4

- テキストのサイズ変更に関する達成基準
  - コンテンツ又は機能を損なうことなく、テキストを支援技術なしで200%までサイズ変更できなければならない。ただし、キャプション及び画像化された文字は除く。
    - 注記1 サイズ変更は、ユーザエージェントの初期設定を基準とする。200 %は、幅及び高さを2 倍にすることである。
    - 注記2 この達成基準は、等級AA の達成基準である。

# あるサイト



# 「大きく」とすると.....





# 「重要なお知らせ」に行くと



# お詫びのページ



# 文字サイズ変更ボタンの 問題点

# わかりにくい

- 配置やデザインがまちまち
  - サイトによって右にあったり上にあったり
  - 目立たないことも
    - グラフィックデザインとしての収まりを優先?
- 機能やラベルがサイトによってまちまち
  - トグルの場合と拡大率増減の場合
  - 拡大、縮小、ああ、AA、など
- サイトごとに使い方を学習する必要あり

# 力不足、機能不足

- 画像は拡大されないことが多い
  - 画像メインのページではほぼ無意味
- 拡大率が不足していることが多い
  - 200%の拡大ができるケースは少ない
- 一貫性がないことが多い
  - 詳細ページなどについていないことがある
  - 関連サイトに移動すると  
設定が元に戻ってしまうことがある

# vs. ブラウザのズーム機能

- 最近のブラウザにはズーム機能がある
- ブラウザのズーム機能を使うと.....
  - どのサイトでも共通の操作
  - 画像も拡大できる
  - 十分な拡大率 (IE400%、Chrome300%)
  - どのページに行っても設定が維持される
- ユーザーが機能を知らない可能性はある
  - サイト側でブラウザ機能の説明をすることも

# Understanding WCAG2.0では

Understanding Success Criterion 1.4.4 | Understanding WCAG 2.0 - Mozilla Firefox

http://www.w3.org/WAI/GL/UNDERSTANDING-WCAG20/visual-audio-contrast-scale.html

Understanding Success Criterio...

W3C<sup>®</sup>

Understanding WCAG 2.0

Contents Intro Previous: SC 1.4.3 [Contrast (Minimum)] Next: SC 1.4.5 [Images of Text]

## Resize text

### Understanding SC 1.4.4

**1.4.4 Resize text:** Except for captions and images of text, text can be resized without assistive technology up to 200 percent without loss of content or functionality. (Level AA)

**On this page:**

- Intent
- Examples
- Related Resources
- Techniques & Failures
- Key terms

#### Intent of this Success Criterion

The intent of this Success Criterion is to ensure that visually rendered text, including text-based controls (text characters that have been displayed so that they can be seen [vs. text characters that are still in data form such as ASCII]) can be scaled successfully so that it can be read directly by people with mild visual disabilities, without requiring the use of assistive technology such as a screen magnifier. Users may benefit from scaling all content on the Web page, but text is most critical.

The scaling of content is primarily a user agent responsibility. User agents that satisfy [UAAG 1.0 Checkpoint 4.1](#) allow users to configure text scale. The author's responsibility is to create Web content that does not prevent the user agent from scaling the content effectively. Authors may satisfy this Success Criterion by verifying that content does not interfere with user agent support for resizing text, including text-based controls, or by providing direct support for resizing text or changing the layout. An example of direct support might be via server-side script that can be used to assign different style sheets.

The author cannot rely on the user agent to satisfy this Success Criterion for HTML content if users do not have access to a user agent with zoom support. For example, if they work in an environment that requires them to use IE 6 or Firefox.

If the author is using a technology whose user agents do not provide zoom support, the author is responsible to provide this type of functionality directly or to provide content that works with the type of functionality provided by the user agent. If the user agent doesn't

完了

# Resize text:

## Understanding SC 1.4.4

- The scaling of content is primarily a user agent responsibility.
- コンテンツを拡大縮小することは、  
第一にユーザーエージェントが  
果たすべき役割である。



# Sufficient Techniques

Understanding Success Criterion 1.4.4 | Understanding WCAG 2.0 - Mozilla Firefox

http://www.w3.org/WAI/GL/UNDERSTANDING-WCAG20/visual-audio-contrast-scale.html

Understanding Success Criterio...

W3C Editor's Draft

## Sufficient Techniques

1. [G142: Using a technology that has commonly-available user agents that support zoom](#)
2. Ensuring that text containers resize when the text resizes **AND** using measurements that are relative to other measurements in the content by using one or more of the following techniques:
  - o [C28: Specifying the size of text containers using em units](#) (CSS)
  - o Techniques for relative measurements
    - [C12: Using percent for font sizes](#) (CSS)
    - [C13: Using named font sizes](#) (CSS)
    - [C14: Using em units for font sizes](#) (CSS)
  - o Techniques for text container resizing
    - [SCR34: Calculating size and position in a way that scales with text size](#) (Scripting)
    - [G146: Using liquid layout](#)
3. [G178: Providing controls on the Web page that allow users to incrementally change the size of all text on the page up to 200 percent](#)
4. [G179: Ensuring that there is no loss of content or functionality when the text resizes and text containers do not resize](#)

## Additional Techniques (Advisory) for 1.4.4

Although not required for conformance, the following additional techniques should be considered in order to make content more accessible. Not all techniques can be used or would be effective in all situations.

- Providing large fonts by default (future link)
- Using page-percent for container sizes (future link)
- Avoiding scaling font sizes smaller than the user-agent default (future link)

Note: The author won't actually know the font size, but should avoid percentage scaling that results in less than 100%

- Avoiding justified text (future link)

完了

# 基準を満たす実装方法

- 1. ズーム機能を使える技術を用いる
  - HTMLなら、ズームできるブラウザがある
  - それだけでOK、コンテンツ側の対応不要
- 2. 文字サイズ変更に対応できるように
  - 文字サイズを絶対値で指定しない、  
サイズを変えても崩れないようにする、など
- サイト側の文字サイズ変更も3番目に挙げられているが、優先度は低い

# ポイント

- ブラウザの機能に頼っても良い
  - コンテンツだけでは成立しない
  - ブラウザが実装している機能をコンテンツ側でも実装することは冗長、推奨されない
- Understanding WCAG2.0 は重要
  - 規格票だけ読んでもダメ
  - 関連文書が重要

おわりに

# なぜアクセシビリティが重要？

- ウェブは本来アクセシブルなメディア
- アクセシブルなウェブサイトが存在するだけで価値がある
- 無限の可能性を秘めている

可能性を失ってしまうのは  
もったいない!!

ありがとうございました