WCAG3 Style Guide

Intended for remigration back to the wiki version of the style guide
Merges the wiki style guide from Michael and the Guidelines, Outcome and Method from Rachael with the SIlver Style Guide originally drafted by Angela.

Other style guides that also need to be followed

1. W3C Manual of Style
2. Chicago Manual of Style
3. EO Style Guide

Before you write

- Think of how readers will use the info in WCAG3.
  - How will they react to it?
  - What do they need to understand and use each guideline?
- Gather your ideas and think of the best way to structure them.
- As you write and trim unneeded content, be sure you convey complete thoughts about your topic.

Tips for writing WCAG3's by section

- WCAG3's Guidelines are technology neutral. Assume that what you write will apply to web *and* other formats. Ask yourself, "Does this guidance make sense for an online shopping website, a social media app, a pizza store website, a mobile app, a virtual reality app, a home assistant, and other technologies?"
- WCAG3's Methods are specific to a technology like web HTML, web SVG, or Android apps. You may use terms—without defining them—that are common for a platform.

Guidelines

- Guidelines group related outcomes.
- Guidelines are in plain language.
- Guidelines have many outcomes.
- Guidelines are a sentence that starts with an imperative verb and end with a period. (For example, "Provide text alternative for non-text content."
- Guidelines have a unique short name that uses initial capitalization and no capitalization on subsequent words.
- Write this last

Outcomes

- Outcomes have an “and” relationship within a guideline. All outcomes must be met or not apply to satisfy the guideline.
- Outcomes have a title (result statement) and a description (benefits).
- Outcomes have many methods.
- All methods within an outcome apply to the same functional category or set of functional categories.
    - If a method applies to a different functional category or set of functional categories than other methods in the outcome, a new outcome should be created.
- Outcomes may not always apply. Outcomes need to clearly state when they should be marked N/A and when they should give credit for not doing something.
    - When an outcome doesn’t apply to a technology, it should be marked N/A (example: Customizing captions in HTML).
    - When creating an outcome includes avoiding something, it should be scored as a success (example: Flashing).
- Outcomes have a unique name as a title. The first word is capitalized and the subsequent words are not.
- The description of the outcome is a sentence that ends with a period and that begins with a conditional verb (plain verb ending in "s") such as: “Uses,” “Provides,” “Organizes,” “Translates.”

Critical errors

- Critical errors are problems that are guaranteed to cause a user to fail.
- If a critical error occurs, the outcome will not pass.
- A critical error shouldn’t duplicate the conditions of a 0 rating on the rating scale. They act as an automatic 0.
- Critical errors can apply to a single object or view.
- Critical errors can be cumulative.
- Three types of critical errors:
    1. These may be items that will stop a user from being able to complete a task if it exists anywhere on the page (examples: flashing, keyboard trap, audio with no pause). They most closely resemble Non-Interference from WCAG 2.2. Critical errors from WCAG 2.x (noninterference, type 1 above):
        - 1.4.2 - Audio Control,
        - 2.1.2 - No Keyboard Trap,
        - 2.3.1 - Three Flashes or Below Threshold, and
        - 2.2.2 - Pause, Stop, Hide.
    2. These may be errors that, when located within a process, mean the user can’t complete the process (example: submit button not in tab order).

3. These may be errors that, when aggregated within a view or across a process, cause failure (example: a large amount of confusing, ambiguous language)

Outcome rating

- The outcome ratings should use the scores that come out of the methods.
- 0-4 Scale
- Each outcome has a customized rating scale.

Methods

- Methods have an “or” relationship within an outcome, when dealing with a specific object.
- There are three types of methods
    - All - These methods apply regardless of technology.
    - Technology specific - These methods apply to one of a set list of technologies.
        - HTML, PDF, VR, etc.
    - Fallback - These methods are used for emerging or proprietary technology.
- Outcomes will have either 1) methods for all or 2) technology-specific methods and fallback methods.
    - Many methods may apply with in a view or process
- Methods contain a description, examples, tests, and scoring.
    - Tests and techniques may have an “and” or an “or” relationship within a method.

Method scoring

- There are two general types of scoring methods
    - Binary - Used when objects being tested have clear boundaries and clear pass/fail conditions.
    - Rating Scale - Used when objects do not have clear boundaries and/or have quality judgements and gradations of good.

Write clearly and consistently

- Write directly to the reader (“you”). Explain to readers what they must or should do (“Use a heading for each section.”).
- Use a friendly and conversational tone, but don’t be overly familiar.
- Guide readers through the content.
    - Be sure that each piece of content relates to the content immediately before and after it.
    - Show readers how parts contrast or relate to each other.
    - Use structure, such as steps and outlines, to make content flow smoothly.

- Write in the active voice.
- Use contractions (“can’t,” “won’t,” “it’s”).
- Use simple words with as few syllables as possible.
- View a list of words to avoid.
- If you use technical terms or jargon, explain them to the reader:

  > “That element is deprecated (it will be obsolete soon, and you should avoid using it now).”

- When you introduce an acronym, first use the complete term and add the acronym in parentheses:

  > “The World Wide Web Consortium (W3C) is an international community…
  > Contact W3C for more information.”

- Use commonly-used words that readers use when they use search engines.
- Write short sentences: Most of your sentences should have 10 or fewer words.
- Write short paragraphs: Most of your paragraphs should have no more than four or five sentences per paragraph, when possible.
- Avoid gerunds (“Write for the web” instead of “Writing for the web”).
- Don’t use casual, incomplete sentences (“Have questions?”)—instead, use a noun and verb in each sentence (“Do you have questions?”).

Inclusive language

- Use culturally-appropriate terms that are meaningful and respectful of all cultures.
- Use “people-first” language when writing about people with disabilities.

  > For example, use “people with disabilities” instead of “disabled people.”

- See the Disability Language Style Guide and the Diversity Style Guide for help with inclusive language.

Note: Translators should translate the inclusive language with sensitivity to the language and culture and not feel obligated to do a direct translation from the US-English.

Additional Resources (pasted from Charles Hall)

Guidelines for Writing About People With Disabilities — ADA National Network
Disability Language Style Guide — National Center on Disability and Journalism
Self-Defined — Tatiana Mac
A modern dictionary about us. We define our words, but they don’t define us.
Content Style Guide §Ableist Language — a11yproject
Ableism / Language — Lydia X Z Brown

Bias-free communication — Microsoft
Alex.js — Titus (@wooorm)
Conscious Style Guide — Karen Yin
Disability Language Is A Nuanced Thing — Nicolas Steenhout

Structuring content

- Use the “inverted pyramid” method for writing headings, sentences, paragraphs, and links: Put the most important content first and add supporting details after it.
- Use headings and sub-headings to break up content.
- Use bullets and numbered lists.
- Write about only one main idea or topic in a paragraph.
- Use enough white space to make content readable.

Formatting

- Avoid italics when possible.

Punctuation

- Use punctuation consistently.
- Use a serial comma in a list of three or more items in a sentence (“Writing in plain language helps people with cognitive disabilities, people with low-language proficiency, and people who are non-native language speakers.”).
- Quotation marks
  - Use straight quotation marks when you write code samples (<img src="example.gif" alt="Image description">).
  - Use curly quotation marks in narrative content (We learned about “alt text” for non-text content.).
  - Use closing quotation marks after periods and commas (We use the term “alt attribute.”), and before other punctuation (What is an “alt tag”?).
  - But always place punctuation inside quotation marks when the punctuation is part of a quote (My teacher asked, “What is the best alt text for this graphic?”).

Here’s a good online resource for detailed standard punctuation advice, including examples.

Grammar (needs work)

[To be written]

Here’s a good online resource for detailed standard grammar advice, including examples.

Capitalization

Special terms ("outcomes", "success criteria")

- Lower case when referenced inline
- Sentence case in the glossary

Headers

- Sentence case

Lists in W3C documents

- Lists that continue a sentence: each list item is lowercase, a single clause, terminated by a semi-colon except:
    - second-to-last terminated by semi-colon and "and";
    - the last one that is terminated by a period.
- Other lists: each list item is a single sentence in sentence case, terminated by a period.
- List items don’t include paragraphs or other block-level elements inside.
- List items of more than one sentence generally should become sections instead.
- **To Be Discussed** When to use ordered lists and when to use bulleted lists.

Writing user needs

Process for Writing User Needs:

1. List a bullet point for each user group that is affected by a technology and how that technology can be a barrier (as if you were a self-advocate or other advocate). Write this list of bullet points in the User Needs Extended Description section.
   1b - If you’re creating a new guideline, then the team writing the guideline should/must include people with disabilities...
2. Identify any needs that user groups have in common and summarize these as the first sentence of the User Need Extended Description. Next, identify any additional needs of specific user groups and include those as bullet points below the sentence. Last, identify any conflicts that may exist within and between user group needs, and list these conflicts as additional bullet points.
3. Check if there are terms in the first sentence that need examples or description to make the context clear? This doesn’t replace eliminating words nor is it a glossary. This becomes the second sentence of the User Needs Extended Description.
4. Write the crux of the solution to meet the shared need. This minimal concept is the third sentence of the User Needs Extended Description.

5. Think about the perspective of a beginner user. Re-draft the common needs paragraph so your beginner user will understand how the barrier affects people. Copy the refined paragraph and bullets into Step 4 Get Started under Why?
6. Refine list from Step 1 in plain language. Copy this into the “Who?” in the Step 4: Get Started page.
7. Write a one-sentence summary line of the user needs paragraph. This is the short description of the “identifying user needs Short Description” as well as the first sentence of the “Summary” in the Get Started page.

Writing tests (needs work)

Note: This section needs a lot of work and should wait for the joint meeting with ACT.

1. Use the “How to Meet WCAG” or “Understanding” to identify the list of existing Techniques.
2. Use the list of applicable WCAG 2.1 Techniques (Sufficient, Advisory and Failure Techniques). Link to them rather than trying to modify the Techniques. We’ll focus on writing new tests that don’t already exist.
3. Identify where new tests will be needed.

HTML rules for WCAG3

HTML structure

- Sections
    - Use section elements with headings for every sub-section
    - Do not use sub-sections unless there is more than one
- Inline
    - Use q instead of " for quotes, to allow generation of curly quotes

Images

- Coding
    - Use <figure> with <figcaption> for main images
    - Use <img> with alt for small inline images
    - Do not use size attributes, allow size to come from native size and / or CSS
- Screen shots
    - Size images to display well at a width of 400 px
    - Image should be in 4:3 width to height ratio (800 x 600 recommended)
- Relationship diagrams
    - Use SVG

    - Size containing box to drawing

Informative content

- Written in plain language

Normative content

- Include a plain-language summary (see Plain Language section)
- Guidelines
    - Name
        - short label
        - clause with no sentence punctuation
    - Guideline
        - full sentence
- Outcomes
    - Name
        - short description of result
        - clause with no sentence punctuation
    - Outcome
        - is benefits to the user
        - full sentence

Definitions

- The term in the glossary should be the singular form, even when commonly used in plural.
- First paragraph of the definition should be a concise description suitable for incorporation into a tooltip.
- Definitions should be full sentences or dictionary-like clauses, not “drop-in” substitutions as in WCAG 2.

Term references

- Link the first time in each top-level section
- When terms and sections are the same, link to the term, which links to the section

WCAG plural vs. singular

- The W3C Content Accessibility Guidelines 3.0 are...
- WCAG 3.0 is...