

Week 2 – Web Development

Progressive Enhancement

Progressive enhancement means building a website so it works for everyone, no matter what device they use. We do this by first focusing on content and structure, then adding styling for appearance, and finally adding functionality for interactivity.

Using progressive enhancement, we must design our web site/application in three stages:

Content and Structure: First, decide what information goes on the page and use HTML to organize it. The page should be readable on any device or browser, even with screen readers.

Styling (CSS): Next, use CSS to make the page look nice and organized. Make sure the design still works everywhere, including for users who rely on screen readers.

Functionality (JavaScript): Finally, add JavaScript to make the site interactive and dynamic. It should improve the experience but not be required to use the website fully, so everyone can still access it.

A website should always work for every user and device, no matter what. Styles and JavaScript should make the experience better but never block access for anyone. For example, people using screen readers need the page structure to stay logical, like reading the header first, links in order, and the footer last. If JavaScript features don't work for them, there should be another option so they can still use the site completely.

Advantages of Progressive Enhancements

Developing your site following the PE approach has some advantages:

1. **Accessibility:** Progressive enhancement makes websites accessible to everyone by separating content, design, and functionality into layers. Each layer works on its own, so even if JavaScript isn't supported, users can still use the site.
2. **Portability:** Portability means your site can work well on any device or browser. With progressive enhancement, even users with assistive tools like screen readers can access your site without problems.
3. **Modularity:** Modularity is like building with Lego blocks—each piece works independently but fits with others. Progressive enhancement helps you create reusable parts that work across different projects without changes.
4. **Performance:** Progressive enhancement improves loading speed because browsers can cache style and script files separately. This means when users revisit a page, only the HTML reloads, saving time and resources.

Exercise: Answer these questions down below with the article links:

What is progressive enhancement (PE)?

Building a website that works for everyone on any device by focusing on content and structure, adding style for appearance and then finally, functionality for interactivity.

Why is it important?

It ensures that websites remain accessible, usable, and visually appealing across all browsers, devices, and assistive technologies.

What is graceful degradation, and how is it different from progressive enhancement?

Graceful degradation starts by building for advanced browsers and then trying to make the site still work (though less perfectly) on older ones, while progressive enhancement starts with basic functionality and builds upward for modern capabilities.

What are the four major reasons for developing with progressive enhancement in mind? Make sure you're able to briefly explain each in your own words.

The four reasons for using progressive enhancement are accessibility (everyone can access content), portability (it works across devices and browsers), modularity (each layer can be developed or fixed independently), and performance (pages load faster because browsers cache separate HTML, CSS, and JavaScript files).

Accessible Web Design and Development

Accessible websites and apps are essential because they must work for everyone, including people with disabilities. Many countries legally require all digital tools used for work or education to be accessible. If a site excludes certain users, organizations won't be allowed to use it. Even if it's not required by law, making your project accessible shows inclusivity and respect for all users.

Start with Accessible Design

When building a website or app, you need to understand what barriers can make it hard for some people to use. It's easier and more efficient to include accessibility from the beginning rather than fixing it later.

Accessibility: Always test your website or app for accessibility by trying it with just a keyboard, no sound, or a screen reader. On Windows, you can use the built-in screen reader called Narrator to do this. In its settings, turn on the option that makes the voice emphasize formatted text so you can check how well your site is read aloud.

Narrator

Change the level of detail Narrator provides about text and controls

Press Narrator + V to change how much detail Narrator provides about text and controls.

Change how capitalized text is read

Press Narrator + 4 to change how capitalized text is read.

- Have Narrator's voice emphasize formatted text
- Hear phonetics as you read by character
- Have Narrator use punctuation to determine pauses when reading
- Hear advanced detail, like help text, on buttons and other controls

Ex. Windows Narrator: "Have Narrator's voice emphasize formatted text." is checked.

There are two sets of standards for web accessibility:

- WCAG (Web Content Accessibility Guidelines)
- WAI-ARIA (Web Accessibility Initiative – Accessible Rich Internet Applications)

Both of these standards are maintained by the World Wide Web Consortium (W3C).

WCAG is the international standard that explains how to make websites and apps accessible to everyone. It includes guidelines that set goals for accessibility and success criteria that help you test if those goals are met. WCAG also provides techniques you can follow to reach those accessibility standards in your web projects.

WAI-ARIA helps make web applications with dynamic or interactive content accessible to everyone. Many screen readers can't interact with JavaScript features, which makes it hard for users with disabilities to use such sites. ARIA provides special attributes and techniques that let developers create interactive elements that can still be used without a mouse or visual cues. It mainly supports users with vision or mobility challenges but also improves accessibility for others.

What are the four principles of WCAG?

Perceivable

To perceive means to notice or become aware of something through your senses. For example, you might feel your dog drop slippers at your feet or your cat swat your hand when it wants you to stop petting it.

Perception is all about how you sense and interpret what's happening around you.

- *You smell smoke – you perceive that something is burning.*
- *You hear a crash upstairs and perceive that the kids or pets probably just destroyed something expensive.*
- *You perceive the burner on the stove is on by putting your hand above it and feel the warmth.*
- *You perceive that the cookies are done by their golden brown tops.*
- *You taste the sauce you're making and perceive that it needs more salt.*
- *Some people perceive that the spirit of a passed loved one is with them when you feel a warm calm come over them.*

A blind person can't tell if a form input is wrong if the only clue is a red border they can't see. Someone with memory problems might not realize they've already visited a link if its color doesn't change. A Deaf person can't follow instructions in a video if there are no captions or transcripts provided.

This section of the guidelines explains how to ensure the content on your web site can be perceived by all users using at least one of their senses.

- If your website depends on visuals, you need to make sure blind or low-vision users can still access the same information in another way.
- If your website uses audio, provide text descriptions or captions so Deaf or hard-of-hearing users can understand the content.
- Smell isn't relevant to web design—unless future technology makes "click and sniff" websites possible.

This part of the guidelines explains how to add text alternatives and captions or transcripts for things like images, videos, and audio. It also covers how to design content clearly using proper layout, color, contrast, size, and spacing so everyone can easily understand it.

Operable

Means something that can be used or function properly. For example, a broken computer or an unplugged microwave isn't operable, just like a car isn't operable for someone with low vision unless it drives itself.

This section of the guidelines states that:

- All navigation menus, links, and interface elements on a website or app must be usable by everyone.
- Users should be able to use the site or app entirely with a keyboard if needed.
- Everyone should have enough time to read, listen to, and interact with the content.
- The design should make it easy for users to find what they're looking for and move around.
- Content should avoid flashing or intense animation that could cause seizures or physical discomfort.
- If your site uses gestures like dragging or pinching, you must provide alternative ways for users who can't perform those actions.

Understandable

Making content understandable means everyone should be able to clearly grasp what it says or shows. This includes using clear language, defining abbreviations, and matching the reading level to your audience. It also means keeping navigation consistent, preventing sudden screen changes, and offering helpful feedback or guidance when users make mistakes.

Robust

means strong and reliable—something that keeps working under different conditions. In web design, it means making sure your site or app works with all kinds of browsers, devices, and assistive tools like screen readers. A robust site stays functional even as new technologies or user agents are developed.

Checking for Accessibility

There are two very useful tools you can use to check the accessibility of your web pages:

1. The browser's Accessibility Tree
2. WebAIM's WAVE tool and browser extension

Accessibility Tree

The Accessibility Tree is a structure your browser automatically builds from the webpage's DOM to store information about every element for assistive technologies.

NOTE

The DOM, or Document Object Model, is how a browser represents your HTML as a collection of objects that can be accessed and modified with JavaScript. Each HTML element becomes a DOM object, even if your code isn't perfect—though valid HTML runs faster and more efficiently. You can view and inspect the DOM and element styles by right-clicking on a webpage and choosing "Inspect" in your browser.

The accessibility tree lists all the elements in a webpage and assigns each one specific properties based on its type or the HTML tag used to make it.

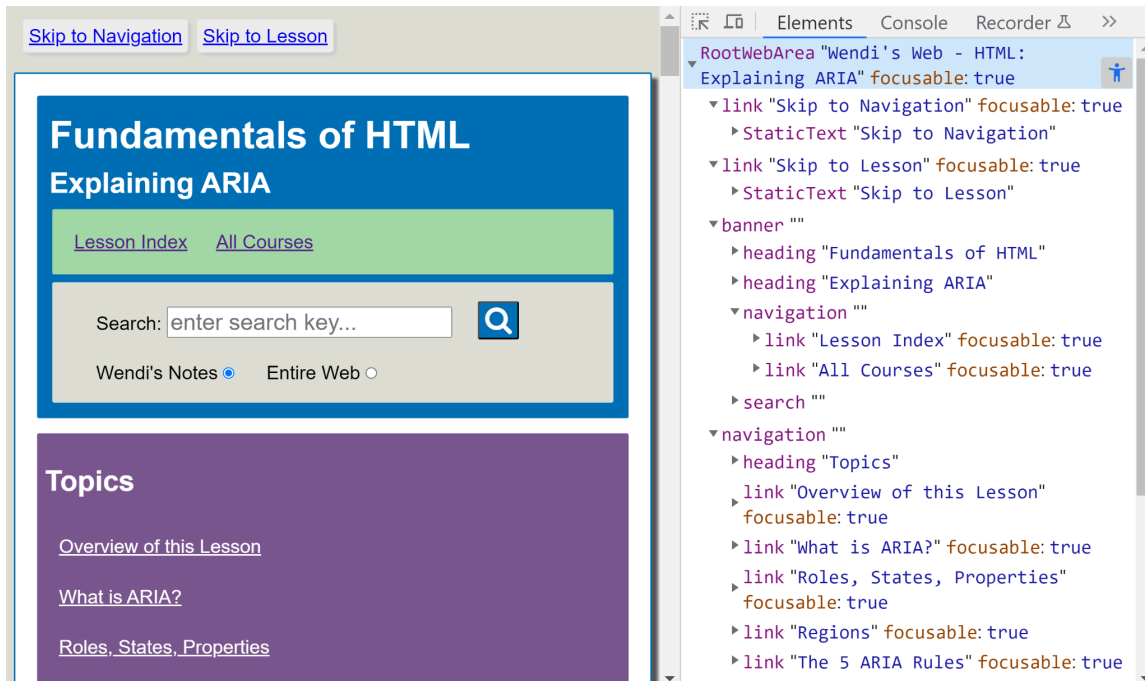
- **Name:** is what identifies the item, usually the visible text for things like buttons or links, or the element's name for structural items.
- **Role:** defines what the element does, such as "banner" for a <header> or "navigation" for a <nav>.
- **Description:** adds extra details that help explain what the element or object is.
- **State(s):** shows the element's current condition, like a checkbox being checked or an input field marked invalid.

The image shows a browser window with a webpage titled "Fundamentals of HTML Explaining ARIA". The page has a blue header with the title, a green navigation bar with "Lesson Index" and "All Courses", a search bar with the text "enter search key...", and a purple "Topics" section. The browser's developer tools are open on the right, showing the "Elements" panel with the Accessibility Tree icon highlighted in a red box. The Accessibility Tree view shows the following structure:

```
<!DOCTYPE html>
<html lang="en">
  <head>
  </head>
  <body >> $0
    <div id="skip">
    </div>
    <div class="container main"> flex
      <header class="container bg-accent1">
        <h1>Fundamentals of HTML</h1>
        <h2>Explaining ARIA</h2>
        <nav class="container">
        </nav>
        <div role="search" class="search-bar container">
        </div>
      </header>
      <nav id="skipnav" class="container page r">
      </nav>
      <main>
      </main>
      <footer class="container">
      </footer>
    </body>
  </html>
```

Ex. To access the Accessibility Tree in Chrome, right-click your page then select Inspect (or equivalent), and click on the Accessibility Tree icon in the top-right.

The Accessibility Tree is a simplified version of the DOM that helps with keyboard navigation and assistive technologies like screen readers. Screen readers don't read the page visually—they move through the Accessibility Tree and announce each object and its properties. For instance, if a user selects an input field with an error, the screen reader will read its name, say it's editable, read the text inside, and then note that it's invalid.



An example of an accessibility tree in Chrome

WebAIM's WAVE

WebAIM offers a tool called WAVE that lets you test how accessible a webpage is by entering its URL on their website. If your page isn't online yet or is password-protected, you can use the WAVE browser extension to check it locally.

To set up the browser extension, install it for your browser, then go into your browser's settings for the extension and allow it to view local files (by default this is disabled).

For example, in Chrome I go into Settings > Extensions, then find the WAVE extension and click "Details". Scroll down to the permissions and settings, and toggle ON "Allow access to file URLs".

To use the extension, load a local .html file into the browser. Then click on the WAVE extension icon in your tool bar.

The WAVE extension will open up a window on the left side of your page. It contains 6 tabs:

The summary section gives an overview of your webpage's accessibility details.

- *Errors show where your code breaks WCAG rules.*
- *Contrast errors highlight text or design elements that don't meet WCAG's minimum color contrast standards.*
- *Alerts flag potential issues like skipped heading levels or missing accessible elements that aren't strict errors.*
- *Features list accessibility tools you've added, such as labels, skip links, or ARIA roles.*
- *Structural elements display the main parts of your page, like headers, footers, tables, and lists.*

- *ARIA attributes show any custom accessibility roles or properties you've applied.*

The screenshot displays the WAVE web accessibility evaluation tool interface. On the left, the 'Summary' tab is active, showing the following statistics:

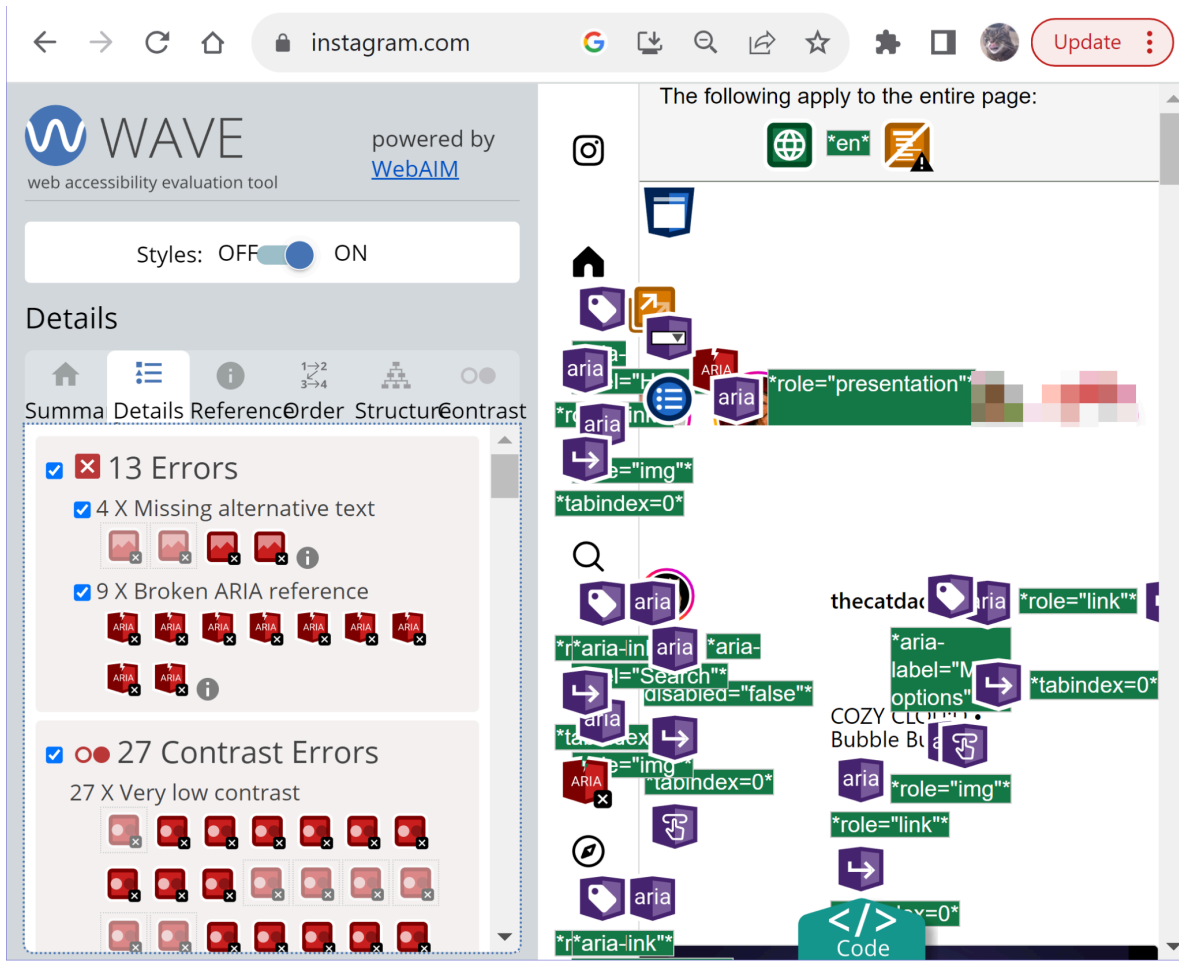
13 Errors	27 Contrast Errors
20 Alerts	27 Features
6 Structural Elements	430 ARIA

The main content area shows a preview of the Instagram homepage with various ARIA attributes overlaid on the elements. The attributes include:

- `*role="presentation"`
- `*aria-label="Search"`
- `*aria-disabled="false"`
- `*tabindex=0`
- `*role="img"`
- `*role="link"`
- `*aria-link`

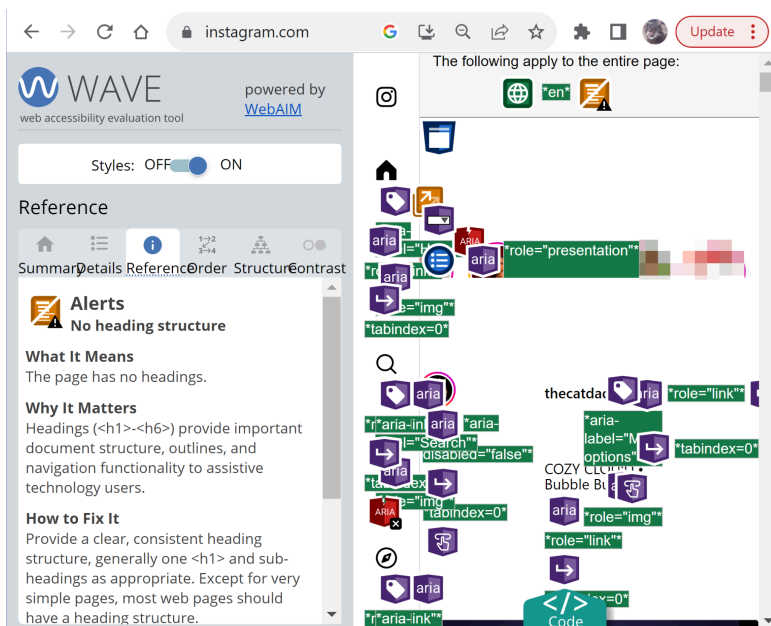
Ex. WAVE Summary tab for Instagram

Details: gives more in-depth information about what's shown in the Summary tab. You can click the info or colored icons to see explanations or locate the specific items on your webpage.



Ex. WAVE Details tab for Instagram

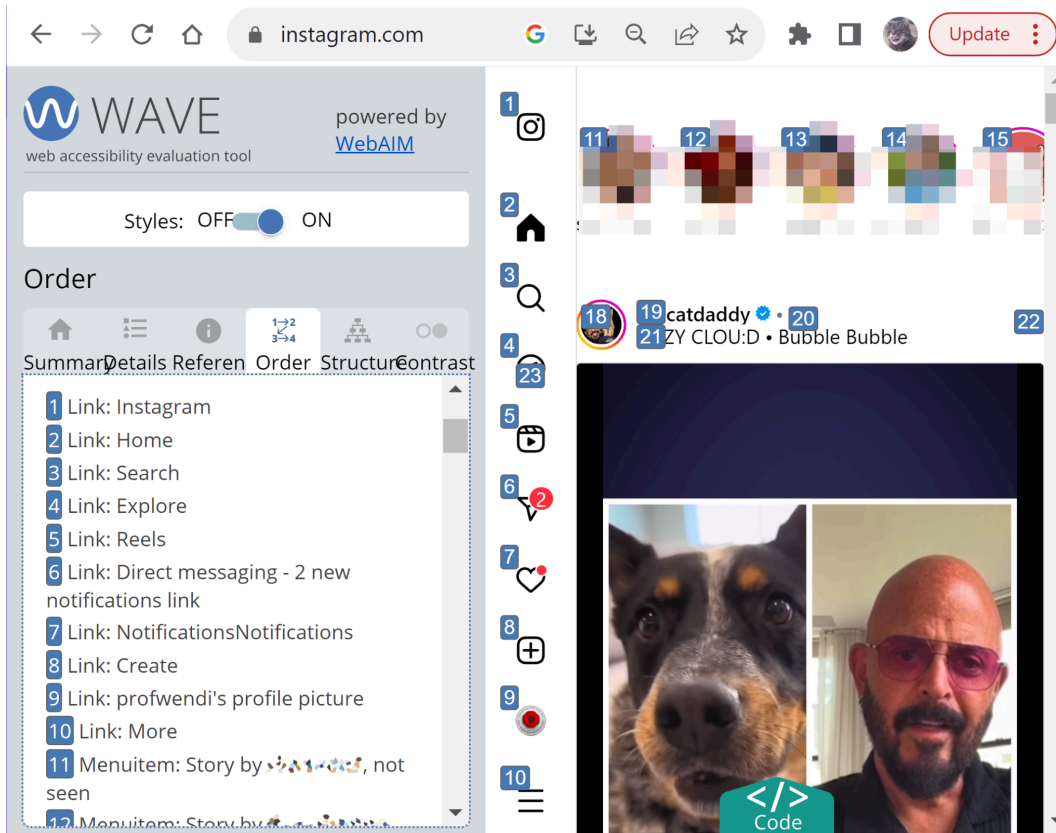
Reference: explains why an item is an error and shows which WCAG guidelines or success criteria relate to it.



Ex. WAVE Reference tab for Instagram

Order: shows the order in which elements on your page can be focused or navigated using the Tab key.

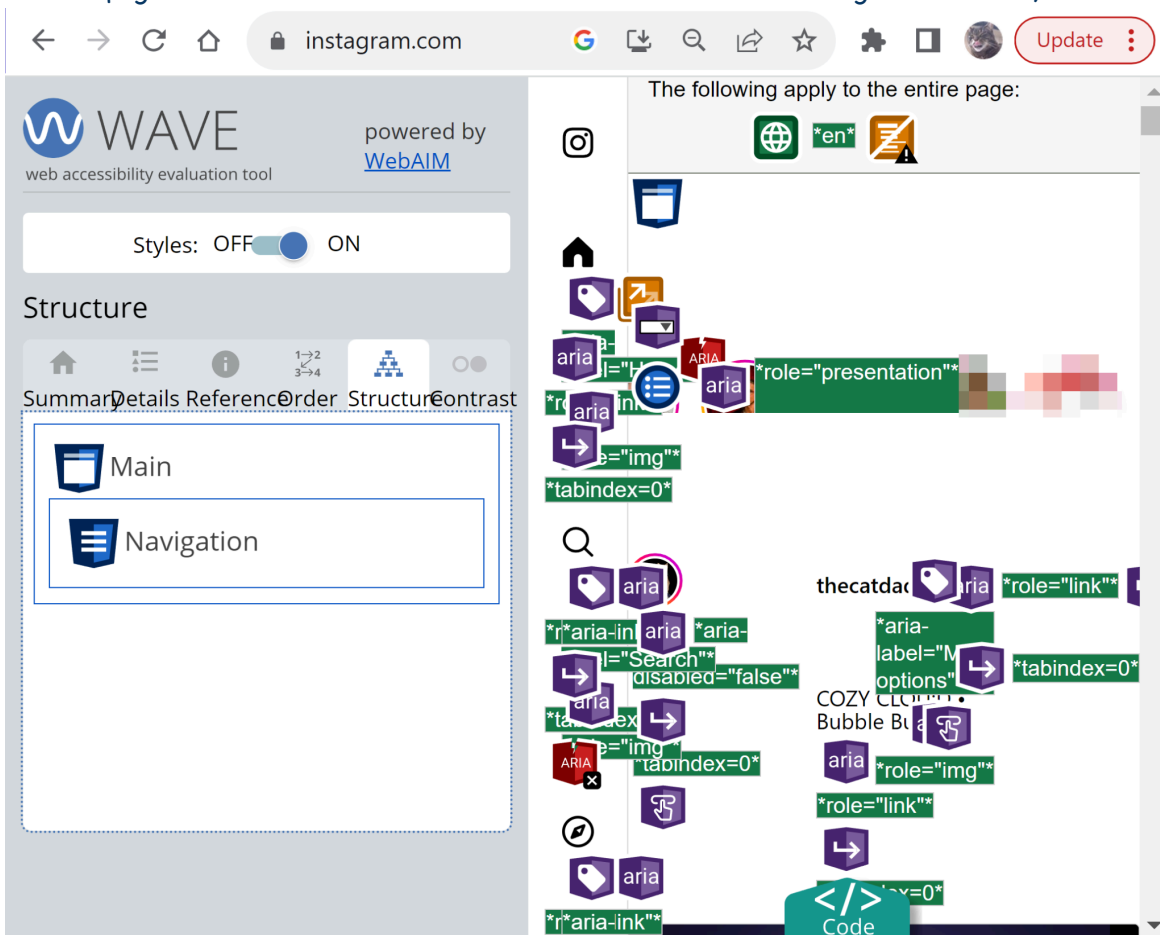
- The numbers show the sequence of focus, with 1 being the first element, then 2, 3, and so on.
- After each number, you'll see the element's role and name, which is what a screen reader announces.
- When you review the Order tab, think about how the page feels and sounds for someone navigating with only the keyboard or a screen reader.



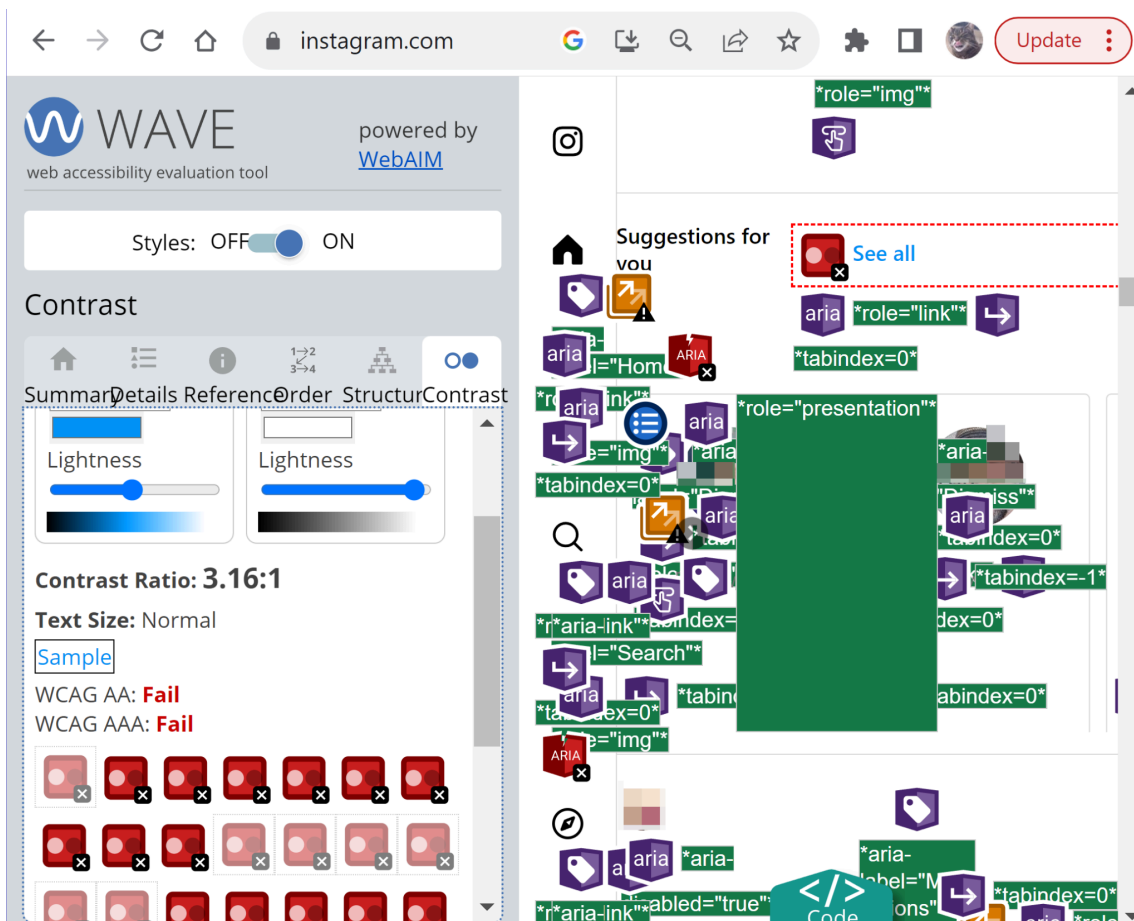
Ex. WAVE Order tab for Instagram

Structure: shows a visual outline of your webpage's structure. Screen readers use these structural "landmarks" to help users move quickly between different parts of the page.

Ex. If a page lacks structure, it becomes harder for users to navigate effectively.



Contrast: shows the color contrast ratio for items on your webpage and lets you test specific color combinations. It also tells you whether your colors meet the minimum accessibility standards for WCAG AA or AAA compliance. To check, click a red icon in the Summary or Details tab, then open the Contrast tab to view the results.



Introduction to Web Accessibility

Introduction

The internet has become such an essential part of life that it's often seen as one of the most revolutionary inventions since the printing press. But its benefits only truly exist for people who can see the screen, hear the audio, and use a mouse.

- **The Web Offers Unprecedented Opportunities**

The internet has been life-changing for people with disabilities. In the past, blind individuals had to rely on Braille books, audio tapes, or others to read printed material aloud. Now, screen reader software can read online content directly to them. This means they can access and explore information independently anytime. People with motor disabilities use assistive tools like mouth wands, special keyboards, or eye-tracking software to navigate websites. Those who are Deaf or hard of hearing can use captions and transcripts to understand multimedia. Even people with cognitive disabilities benefit from the web's organized and adaptable content.

- **Falling Short on the Web's Potential**

The internet has the power to greatly benefit people with disabilities, but much of that potential hasn't been achieved yet. Many websites still depend on visual or audio interaction, making them hard to use for people who can't see, hear, or use a mouse. By recognizing these barriers, we can start removing them to make the web truly accessible for everyone.

People with Disabilities on the Web

One out of every four adults in the U.S. has a disability. Even if not all disabilities affect internet use, excluding any portion of that group limits a business's potential customers. In education and government settings, such exclusion can even be considered illegal discrimination.

Disability Categories

Visual disabilities include blindness, low vision, and color blindness.

Auditory disabilities involve deafness or difficulty hearing.

Motor disabilities affect a person's ability to use a mouse, react quickly, or control fine movements.

Cognitive disabilities can make it hard to learn, focus, or remember information.

Each of these needs thoughtful design strategies that not only help people with disabilities but also make websites easier for everyone to use.

Implementing Web Accessibility

To make a website accessible, you need to understand what accessibility means, commit to it, learn how to apply it, and know the legal requirements.

Commitment and accountability

Awareness begins by recognizing the barriers people with disabilities face online.

Leadership plays a big role—when leaders value accessibility, developers are more likely to build accessible websites.

Policies and procedures turn accessibility into a consistent practice by making it part of everyday workflows.

Training and Technical Support

Many people think accessibility means adding extra features, but accessible websites are often simpler and more efficient. Although it might take more time and money at first, accessibility helps reach a larger audience, making the effort worthwhile. Developers can learn the basics quickly, but mastering the mindset and techniques takes practice. Free WebAIM resources like articles, newsletters, and discussion lists help developers, designers, and administrators learn about accessibility. Organizations can also get expert help through services like training, technical support, and website evaluations. Overall, proper training ensures accessibility becomes a lasting part of web development.

Laws and Standards

In the U.S., web accessibility is covered by laws like the Americans with Disabilities Act (ADA) and the Rehabilitation Act of 1973. Many other countries also have their own accessibility laws. The Web Content Accessibility Guidelines (WCAG), created by the World Wide Web Consortium (W3C), serve as the global standard and are built around four main principles.

Perceivable means users can sense the content—mainly through sight or hearing—using their browser or assistive tools like screen readers.

Operable means users can use all site features with a mouse, keyboard, or other assistive devices.

Understandable means the content is clear and easy to follow without confusion.

Robust means the website works well across different browsers, devices, and assistive technologies, old or new.

Principles of Accessible Design

Provide alternative text: Add descriptive text for images so screen readers can describe them to visually impaired users.

Create logical document structure: Use headings, lists, and layout elements to organize pages and help with keyboard navigation.

Provide headers for data tables: Use proper <th> tags to label rows and columns, making tables easier for screen reader users to understand.

Ensure forms are usable: Label all form fields clearly and allow users to submit forms and fix errors easily.
Write meaningful links: Use descriptive link text that makes sense on its own instead of vague phrases like “click here.”

Add captions or transcripts for media: Provide captions for videos and transcripts for audio so everyone can access the content.

Ensure non-HTML documents are accessible: Make PDFs, Word files, and other documents screen reader-friendly or offer accessible HTML versions.

Allow skipping repetitive content: Include “Skip to Main Content” links so users can bypass repeated navigation.

Don't rely on color alone: Use text or symbols along with color to convey meaning for color-blind users.

Write clearly and use readable design: Keep language simple, fonts clear, and content well-organized for easy reading.

Make JavaScript accessible: Code interactive features so users can use keyboards or assistive devices, not just a mouse.

Design to standards: Use valid HTML and CSS to create flexible, reliable, and accessible website code.

Conclusion

The internet gives people an unmatched level of freedom and independence. When websites aren't accessible, they shut out people who could benefit from that access the most. By learning about accessibility and applying it, you help make the web open to everyone.