Alliance for Qualification

Release Version 1.002 EN

June 2025

A4Q Syllabus

Certified Professional

Digital Accessibility

Copyright Notice and Disclaimer

Versions

Version	Datum	Remark
1.0 EN	30.06.2025	Internally reviewed document

Disclaimer

This document is intended solely for general informational purposes regarding the evaluation of digital accessibility, based on the legal requirements, standards, and technical specifications available at the time of publication (e.g., WCAG, EN 301 549, EAA, ADA, Section 508).

The authors make no guarantees as to the completeness, accuracy, or timeliness of the information provided. This document does not constitute legal advice or a formal accessibility audit in a legal or regulatory sense.

For the legally compliant implementation of accessibility requirements, it is strongly recommended to consult the most current legislation and official technical standards, and to seek expert advice where necessary.

Note on the Use of AI

This syllabus was developed in part with the assistance of AI-based tools and has been editorially reviewed for accuracy and consistency.

Table of Contents

Release Version 1.00

This document was officially published by A4Q GmbH on June 30, 2025.

0 Preamble

0.1 Purpose of the Syllabus

This syllabus defines the intended business outcomes, learning objectives, and core concepts of the Certified Professional Digital Accessibility program. It serves as the foundational framework for developing, delivering, and evaluating training aligned with internationally recognized accessibility standards. The syllabus is designed to guide learners toward a shared understanding of digital accessibility as a strategic, technical, and content-driven discipline that supports inclusive digital environments.

0.2 Target Audience

This curriculum is designed for a diverse group of professionals and decision-makers involved in the planning, creation, development, and strategic management of digital products and services. It addresses both technical and non-technical roles, with the goal of embedding accessibility into all phases of digital project lifecycles. The program supports the implementation of the Web Content Accessibility Guidelines (WCAG 2) and compliance with internationally recognized accessibility standards and legal frameworks, such as the Americans with Disabilities Act (ADA), Section 508 of the Rehabilitation Act (U.S.), and the European Accessibility Act (EAA).

The curriculum is specifically tailored for the following target groups:

Web Developers (Frontend and Backend)
Responsible for implementing accessibility features such as semantic HTML, ARIA roles, and keyboard operability. They ensure that accessible components and functionalities are built into websites and web applications.

UX and UI Designers
Incorporate accessibility principles into the design of user interfaces, considering elements such as sufficient color contrast, focus management, and accessible interaction patterns to support inclusive user experiences.

Content Managers and Editors
Create and maintain accessible digital content, including text, images with descriptive alt text, multimedia with captions and transcripts, and accessible documents. They apply plain language, clear structure, and consistent formatting practices.

Testers and Quality Assurance (QA) Professionals
Conduct manual and automated accessibility testing, identify barriers, and document issues in accordance with WCAG 2 success criteria.

Project Managers and Product Owners
Plan and allocate resources for accessibility implementation, coordinate efforts across design, development, content, and testing teams, and ensure that deliverables meet quality and compliance standards.

Executives and Decision-Makers (e.g., Department Heads, C-Level Leaders)
Provide strategic leadership for digital accessibility initiatives, ensure compliance with legal obligations (e.g., ADA, EAA, Section 508), and foster a culture of inclusion through policy development, investment, and organization-wide training.

IT Strategy, Compliance, and Data Privacy Officers
Align accessibility with enterprise governance, compliance, and risk management frameworks. They support documentation, monitoring, and oversight of accessible digital solutions.

External Service Providers, Agencies, and Freelancers
Are contracted to deliver accessible digital products and services and must comply with the accessibility requirements and quality standards defined by their clients or applicable regulations.

By addressing the specific responsibilities of each role, this curriculum fosters a unified, organization-wide approach to digital accessibility.

0.3 Exam-Relevant Learning Objectives and Cognitive Knowledge Levels

The learning objectives support the defined business outcomes and serve as the foundation for the development of the Certified Professional Digital Accessibility certification exams. Candidates are expected to demonstrate the ability to recall key content from each chapter of the syllabus, show comprehension of core concepts, apply knowledge in practical contexts, and analyze scenarios related to digital accessibility.

These objectives are structured to align with progressive cognitive levels—ranging from remembering and understanding to applying and analyzing—and form the basis for both instructional design and assessment criteria.

The knowledge levels assigned to each learning objective are indicated at the beginning of each chapter and are classified as follows:

- K1 – Remembering:
  The ability to recall and reproduce facts, terms, and specific information.
- K2 – Understanding:
  The ability to grasp, interpret, and explain meanings and concepts.
- K3 – Applying:
  The ability to apply knowledge in concrete situations or to solve specific tasks.

0.4 The “Certified Professional Digital Accessibility” Examination

The Certified Professional Digital Accessibility examination is based on the content of this syllabus. Answers to exam questions may require the integration of knowledge from multiple sections of the syllabus. All sections are relevant for the exam, apart from the introduction and the appendices.

Standards, books, and legal regulations may be referenced for context; however, only the content explicitly summarized within this syllabus is considered exam relevant.

The exam consists of 40 multiple-choice questions. Each correct answer is awarded one point. A minimum score of 65%, that is, at least 26 correct answers, is required to pass the exam. The allotted time for completing the exam is 60 minutes. 25% extra time is available where English is not the candidate’s native language.

If the candidate’s native language is different from the language of the exam, an additional 25% of exam time (15 minutes) may be granted.

0.5 Accreditation of Training Providers

Training providers that wish to offer courses for the *Certified Professional Digital Accessibility* certification should be accredited by A4Q and are required to use the official training materials for *Certified Professional Digital Accessibility*. This ensures consistent quality, alignment with the syllabus, and proper preparation of candidates for the certification exam.

0.6 Level of Detail

Structure and Scope of the Syllabus

This syllabus consists of the following elements:

- **Business Outcomes** that define what participants will be able to achieve after completing the training in order to effectively address business-related challenges.
- **General learning objectives** that describe the overall purpose of the certification.
- **Glossaries or lists of key terms** that participants are expected to be able to explain.
- **Detailed learning objectives for each knowledge area**, specifying the required level of cognitive learning to be attained (aligned with knowledge levels K1–K3).
- **Descriptions of core concepts**, including references to recognized standards, literature, or authoritative sources.

This syllabus is not intended to exhaustively cover the entire field of accessibility and related standards. Instead, it reflects the level of detail appropriate for training and certification at the Foundation Level.

This syllabus does not include specific learning objectives related to particular process models or development approaches in software engineering

0.7 Structure of This Syllabus

This syllabus consists of six chapters containing exam-relevant content. Each chapter title indicates the recommended duration; no additional time allocations are provided at the subchapter level. For training purposes, the syllabus suggests approximately 840 minutes (14 hours) of instruction, distributed across the chapters and practical exercises as outlined.

The actual time required may vary depending on factors such as participants' prior knowledge, specific areas of interest, the depth and number of practice examples, and other instructional considerations.

Course Structure and Suggested Time Allocation

Chapter	Title	Duration (approx. minutes)	Duration (approx. hours)
1	Introduction to Digital Accessibility	30	0.5
2	User groups with accessibility needs	90	1.5
3	Legal Requirements, Guidelines, Standards, and Regulations (Overview)	90	1.5
4	Conducting an Accessibility Evaluation	480	8
5	Strategies for Successful Implementation of	120	2

Chapter	Title	Duration (approx. minutes)	Duration (approx. hours)
	Accessibility		
6	Automated Testing Tools in Accessibility Evaluation	30	0.5
	Total	**840**	**14**

0.8 Business Outcomes (BOs)

Chapter 1: Introduction to Digital Accessibility

BO 1.1 Establish a solid and shared understanding of digital accessibility and its four core principles—Perceivable, Operable, Understandable, and Robust—as a foundation for digital projects.
BO 1.2 Make informed decisions regarding the appropriate WCAG version and conformance level based on project requirements and legal or organizational objectives.

Chapter 2: User Groups with Accessibility Needs

BO 2.1 Design digital products that can be perceived and operated by the widest possible range of users—regardless of whether their limitations are temporary, permanent, or situational—by applying core accessibility principles.
BO 2.2 Develop digital solutions tailored to the specific needs and requirements of diverse user groups with disabilities, including visual, auditory, motor, and cognitive impairments.
BO 2.3 Ensure compatibility and effective use of digital offerings with assistive technologies to enable equal access and participation.
BO 2.4 Validate the usability of digital products through the inclusion of users with various disabilities in testing processes.

Chapter 3: Legal Requirements, Guidelines, Standards, and Regulations

BO 3.1 Ensure compliance with binding legal requirements and relevant standards to achieve legal certainty and avoid penalties.
BO 3.2 Use standardized evaluation frameworks to systematically assess, document, and demonstrate conformance with legal accessibility requirements.
BO 3.3 Adapt digital offerings reliably in response to evolving legislation to ensure long-term legal compliance.

Chapter 4: Conducting an Accessibility Evaluation / WCAG Success Criteria

BO 4.1 Reliably implement key WCAG Level AA success criteria (e.g., text alternatives, contrast ratios, keyboard operability, semantic markup, form labels, status messages) to ensure digital accessibility.
BO 4.2 Carry out structured evaluation procedures—both manual and tool-supported—to assess conformance with WCAG success criteria, identify barriers, and derive actionable recommendations.
BO 4.3 Effectively prioritize and remediate identified barriers based on their severity and impact on user groups, in order to optimize resource allocation.

Chapter 5: Strategies for Successful Implementation of Accessibility

BO 5.1 Embed accessibility early and continuously throughout the entire lifecycle of digital products—from concept to maintenance—to achieve efficient, high-quality, and sustainable accessible solutions.
BO 5.2 Plan and organize accessibility testing systematically (defining objectives, scope, test cases; selecting appropriate methods and tools) to obtain reliable results.
BO 5.3 Build and establish internal expertise and clearly defined responsibilities for digital accessibility across all relevant teams, including design, development, and content.

Chapter 6: Automated Testing Tools in Accessibility Evaluation

BO 6.1 Increase the efficiency and consistency of accessibility evaluations through the purposeful use of automated testing tools.
BO 6.2 Ensure comprehensive accessibility by understanding the limitations of automated tools and supplementing them with manual testing methods.
BO 6.3 Optimize development and quality assurance processes by integrating automated accessibility testing to enable early detection of issues.

0.9 Abbreviations

Abbreviation	Meaning / Description
A4Q	Qualification organization responsible for publishing the Certified Professional Digital Accessibility syllabus.
ACA	Accessible Canada Act – A Canadian federal law requiring digital content to conform to WCAG 2.0 or 2.1 Level AA.
ACAA	Air Carrier Access Act – U.S. law mandating accessible digital services within the airline industry.

ADA	Americans with Disabilities Act – U.S. civil rights law prohibiting disability-based discrimination, including digital services.
ADHD	Attention Deficit Hyperactivity Disorder – A neurodevelopmental disorder affecting attention and concentration.
AI	Artificial Intelligence – Machines performing tasks that typically require human intelligence.
ANDI	Accessible Name & Description Inspector – Browser-based tool by SSA for testing web accessibility, used in Section 508 workflows.
AODA	Accessibility for Ontarians with Disabilities Act – Canadian law requiring WCAG 2.0 Level AA compliance in Ontario.
APD	Auditory Processing Disorder – Difficulty processing auditory information despite normal hearing.
AR/VR	Augmented/Virtual Reality – Technologies addressed in WCAG 3.0.
ARIA	Accessible Rich Internet Applications – Enhances HTML to improve accessibility of dynamic content.
BFSG	Barrierefreiheitsstärkungsgesetz – German national accessibility strengthening law.
BITV	Barrierefreie Informationstechnik-Verordnung – German regulation for accessible information technology.
BOs	Business Outcomes – Skills demonstrated after CPDA training.
CART	Communication Access Realtime Translation – Converts speech to text in real time.
CI/CD	Continuous Integration / Continuous Delivery – Development practice integrating accessibility tests.
CMS	Content Management System – Platform where accessibility tools monitor entire websites.
CONADIS	National Council for Disability Inclusion – Mexican authority aligning with WCAG-based guidelines.
CRPD	Convention on the Rights of Persons with Disabilities – UN treaty for digital inclusion standards.
CSS	Cascading Style Sheets – Used to maintain accessibility in visual presentation.
CSV	Comma-Separated Values – Format for automated accessibility test results.
DDA	Disability Discrimination Act 1992 – Australian law requiring equal digital access; WCAG is the benchmark.

DHS	Department of Homeland Security – Maintains the Trusted Tester accessibility evaluation program.
EHRC	Equality and Human Rights Commission – UK authority for the Equality Act 2010.
EIT	Electronic and Information Technology – Must be accessible under Section 508 (U.S.).
EN 301 549	EU harmonized accessibility standard for ICT, referencing WCAG; used in public procurement.
EU	European Union – Member states require WCAG compliance by law.
GDS	Government Digital Service – UK agency managing public sector accessibility compliance.
GIGW	Guidelines for Indian Government Websites – WCAG-based national standard.
HTML	HyperText Markup Language – Semantic use improves accessibility in web development.
ICT	Information and Communication Technology – Covers all digital services in accessibility law.
ISO/IEC	International standardization bodies – Published WCAG 2.0 as ISO standard.
JIS X 8341-3	Japanese Industrial Standard – Aligns with WCAG; governs public services in Japan.
JSON	JavaScript Object Notation – Format for accessibility test results.
K1	Knowledge Level 1 – Recall and reproduce facts and terms.
K2	Knowledge Level 2 – Interpret and explain meanings and concepts.
K3	Knowledge Level 3 – Apply knowledge in concrete situations.
LO	Learning Outcome – Skills or knowledge demonstrated in exams.
MIIT	Ministry of Industry and Information Technology – China's authority for voluntary accessibility standards.
NDIS	National Disability Insurance Scheme – Australian support for people with disabilities.
NHS	National Health Service – UK provider offering services to the legally blind.
NVDA	NonVisual Desktop Access – Screen reader for speech/Braille output of screen content.
PDF	Portable Document Format – Must be tested or offered in accessible alternatives.

PEAT	Photosensitive Epilepsy Analysis Tool – Checks flash frequency to meet WCAG.
POUR	Perceivable, Operable, Understandable, Robust – Four WCAG accessibility principles.
Pa11y	Automated accessibility test tool – Checks alt text, contrast, ARIA, and more.
QA	Quality Assurance – Professionals testing accessibility manually or using tools.
SA Human Rights Commission	South African rights body for accessibility law enforcement.
SC	Success Criteria – Individual WCAG requirements for digital accessibility.
SEO	Search Engine Optimization – Affected by accessibility-related metadata.
SSA	Social Security Administration – Developed the ANDI accessibility tool.
TTS	Text-to-Speech – Converts text into speech for users.
U.S. / USA	United States – ADA and Section 508 define federal accessibility rules.
UI	User Interface – Must support contrast, labeling, and keyboard focus.
UK	United Kingdom – National accessibility laws govern digital access.
UX	User Experience – Accessibility enhances usability for all users.
W3C	World Wide Web Consortium – Maintains the WCAG standard.
WAD	Web Accessibility Directive – EU law requiring WCAG 2.1 AA for public sector sites.
WAI	Web Accessibility Initiative – W3C project developing WCAG.
WAI-ARIA	Web Accessibility Initiative – ARIA – Enhances web application accessibility.
WAVE	Web Accessibility Evaluation Tool – Visual accessibility checking tool.
WCAG	Web Content Accessibility Guidelines – International digital accessibility standard.
WCAG-EM	Website Accessibility Conformance Evaluation Methodology – W3C process for WCAG 2.1 evaluation.

1 Introduction to Digital Accessibility

⏱ Lesson duration: approximately 30 minutes

🎯 Learning objectives of the chapter

LO 1.1.1 – Recall the definition of digital accessibility. – K1
LO 1.1.2 – Classify different types of impairments (permanent, temporary, situational). – K2
LO 1.1.3 – Explain the WCAG levels of conformance (A, AA, AAA). – K2
LO 1.2.1 – Explain the WCAG principle of Perceivability. – K2
LO 1.2.2 – Explain the WCAG principle of Operability. – K2
LO 1.2.3 – Explain the WCAG principle of Understandability. – K2
LO 1.2.4 – Explain the WCAG principle of Robustness. – K2

Digital Accessibility: An Introduction

Digital accessibility refers to the design of digital content, services, and applications, such as websites, mobile apps, or documents, in a way that ensures they can be used effectively by all people, regardless of physical, sensory, cognitive, or technological limitations. The primary goal is to enable equal participation in the digital world and to eliminate discrimination caused by inaccessible digital environments.

Implementing digital accessibility offers a wide range of benefits. It not only grants access to people with disabilities but also improves the overall usability of digital services. Older adults, individuals with temporary impairments (e.g., due to illness or injury), users with limited digital literacy, and those using mobile devices in various environments also benefit from accessible design. Accessibility contributes to increased reach, user satisfaction, and customer loyalty across digital platforms.

There is also a close connection between accessibility and usability. Both concepts aim to make digital content intuitive, understandable, and efficient to use. Measures that enhance accessibility—such as clear content structure, sufficient contrast, alternative text for images, and simple navigation—also improve general user experience and functionality for everyone.

On a regulatory level, digital accessibility is anchored in international and national legislation. The foundational technical standard is the **Web Content Accessibility Guidelines (WCAG)**, developed by the **World Wide Web Consortium (W3C)**. The most current version (WCAG 2.2) outlines the criteria digital content must meet to be considered accessible. The WCAG are based on four guiding principles: content must be **perceivable, operable, understandable, and robust**.

In the United States, digital accessibility is mandated by laws such as **Section 508 of the Rehabilitation Act** and the **Americans with Disabilities Act (ADA)**. Section 508 requires federal agencies to make their electronic and information technology accessible to people with disabilities. The ADA has been interpreted to apply to digital services, including commercial websites and mobile applications. These laws are supported by technical standards aligned with the WCAG.

In the United Kingdom, accessibility requirements are defined by the **Equality Act 2010**, which prohibits discrimination against people with disabilities in accessing goods and services—including digital services. Public sector organizations are additionally bound by the **Public Sector Bodies (Websites and Mobile Applications) Accessibility Regulations 2018**, which require websites and mobile apps of public bodies to meet the WCAG 2.1 AA standard. These regulations apply to government departments, local authorities, universities, and other public institutions. Private sector organizations are expected to follow the Equality Act, which has been interpreted by courts to include websites and digital content as part of the "services" offered to the public.

Globally, countries increasingly reference the WCAG in their own accessibility regulations, including standards like **EN 301 549** in the European Union, which guides public procurement of accessible ICT products and services.

The growing legal obligations and social importance of digital accessibility highlight its role as a core element of inclusive, user-focused digital communication and service design. Accessibility is not only a legal requirement, but also an opportunity to make digital products and services more usable, inclusive, and future-ready for the broadest possible audience.

1.1 What Is Digital Accessibility?

Terms: Digital Accessibility, Temporary Impairment, Permanent Impairment, Situational Impairment, WCAG 2.0, WCAG 2.1, WCAG 2.2, Levels of Conformance A, AA, AAA

This section covers the fundamentals of digital accessibility.

1.1.1 WCAG 2 – Overview

The Web Content Accessibility Guidelines (WCAG) are an international standard for ensuring the accessibility of web content.

WCAG 2.0 (2008)

- Introduced the four principles: perceivable, operable, understandable, robust

- Contains 12 guidelines and 61 success criteria across three levels (A, AA, AAA)
- Focus: fundamental accessibility, e.g., text alternatives (1.1.1), keyboard navigation (2.1.1), minimum contrast (1.4.3)
- Technology-neutral, legal standard in many countries

WCAG 2.1 (2018)

- Focus: mobile use, cognitive disabilities, visual impairments

WCAG 2.2 (2023)

- Focus: improved usability, simplified authentication
- New criteria: visible focus (2.4.11), accessible login without CAPTCHAs (2.2.6), larger target areas (2.5.8)

All versions define three levels of conformance:
A = minimum requirements
AA = commonly required by law
AAA = highest level of accessibility

1.1.2 Digital Accessibility

Digital accessibility refers to the design, development, and delivery of digital content, applications, and services in a way that ensures equal access and usability for all individuals — regardless of physical, sensory, cognitive, or situational limitations.

The **Web Content Accessibility Guidelines (WCAG)** define four fundamental principles that digital content must follow to be considered accessible:

1. **Perceivable**
   Content must be presented in ways that all users can perceive. This includes text alternatives for images, sufficient color contrast, resizable text, and accessible audiovisual content.
2. **Operable**
   All interactive elements and navigation structures must be fully operable — whether by keyboard, mouse, or other input methods. Examples include large clickable areas, avoiding time constraints, and clear focus indicators.
3. **Understandable**
   Information and interface behavior must be easy to understand and predictable. Key elements include consistent layout, plain language, intuitive navigation, and helpful error messages.

4. **Robust**
   Content must be developed to work reliably with a wide range of browsers, devices, and assistive technologies — and remain functional as technology evolves.

WCAG 2.2 expands upon earlier versions and places special emphasis on **operability for users with motor and cognitive impairments**. New success criteria include improved accessibility of interactive elements and simplified user interactions.

Importantly, digital accessibility is not limited to people with permanent disabilities. It also supports individuals with **temporary impairments** (e.g., an injury) and **situational limitations** (e.g., using a device in bright sunlight or in a noisy environment). In this sense, accessibility is about enabling full participation in the digital world for **everyone**.

1.1.3 Categories of Impairments

The four principles of digital accessibility — perceivable, operable, understandable, and robust — address various types of impairments that may affect the use of digital content. These impairments can be grouped into three categories:

- **Permanent impairments** are long-term physical, sensory, or cognitive limitations. Examples include blindness, deafness, limited mobility, or learning disabilities. Digital content must be designed so that it can be fully accessed and used by people with these conditions.
- **Temporary impairments** last for a limited time. These may include an injury, illness affecting motor skills or perception, or fatigue that reduces focus. Accessibility standards must account for these users as well.
- **Situational impairments** arise from external circumstances. They affect users regardless of their physical condition — for example, when using a phone with one hand, in poor lighting, or in a noisy environment.

According to the WCAG, considering all three types of impairments within the four accessibility principles ensures **universal access** to digital content. As such, digital accessibility becomes a foundational component of an inclusive and user-centered digital environment.

1.1.4 WCAG Levels of Conformance

The Web Content Accessibility Guidelines (WCAG) define three levels of conformance that indicate the degree to which digital content is accessible to users with disabilities:

Level A (Minimum Requirements)

- **Definition:** The most basic accessibility requirements are met. Without meeting these criteria, a website may be unusable for many users.
- **Examples:**
  - Alternative text is provided for images.
  - Interactive elements are accessible via keyboard.
- **Application:**
  - Represents the foundational level of accessibility.
  - Often **insufficient** to meet legal standards on its own.

Level AA (Standard Requirements)

- **Definition:** All Level A criteria are met, along with additional requirements that significantly improve accessibility for a wider range of users.
- **Examples:**
  - Sufficient contrast between text and background (at least 4.5:1).
  - Websites function in both portrait and landscape orientations.
- **Application:**
  - **United States:** Section 508 of the Rehabilitation Act requires federal websites to meet **WCAG 2.0 Level AA** standards.
  - **United Kingdom:** Public sector websites and mobile apps must conform to **WCAG 2.1 Level AA**, as required by the *Public Sector Bodies (Websites and Mobile Applications) Accessibility Regulations 2018*.
  - **Recommended standard** for private organizations, especially in sectors like e-commerce, education, banking, and healthcare.

Level AAA (Highest Requirements)

- **Definition:** All criteria from Levels A and AA are met, along with more advanced accessibility measures.
- **Examples:**
  - Enhanced contrast ratios (e.g., text and image text must have a contrast ratio of at least 7:1).
  - Content is provided in plain language or simplified versions.
- **Application:**
  - **Not legally required** under U.S. or U.K. law.
  - May be adopted voluntarily as a **quality benchmark**, particularly for content targeting users with cognitive disabilities or in high-profile, accessibility-focused projects.

Summary:

- Legal requirements typically mandate compliance with **WCAG 2.0 or 2.1 Level AA**.
- **Level AAA** is optional and represents a commitment to the highest level of accessibility.
- **Compliance monitoring** is increasingly enforced, especially in the public sector.

1.2 Principles of Accessibility

Terms: Perceivable, Operable, Understandable, Robust

This section introduces the **four principles of accessibility** as defined in WCAG 2.2: *Perceivable*, *Operable*, *Understandable*, and *Robust*. These principles ensure that digital content is accessible to all users, regardless of their abilities or the technologies they use. Each principle is defined, linked to specific guidelines, and explained through practical examples.

1.2.1 Principle: Perceivable

This subsection introduces learners to the concept of *perceivability*. Content must be presented in ways that can be perceived by all human senses. This means information must not depend on a single sensory channel such as sight or sound alone.

Examples include:

- **Text alternatives for images** (for blind users),
- **Captions for videos** (for deaf users),
- **Sufficient color contrast** (for users with low vision).

The principle addresses temporary impairments (e.g., eye infection), permanent impairments (e.g., blindness), and situational limitations (e.g., glare from sunlight). The aim is to ensure that content remains accessible under all conditions.

Definition: Information and user interfaces must be presented in ways that users can perceive. Content should not rely solely on color, sound, or other single-sense presentations.

Guidelines: Text Alternatives, Time-Based Media, Adaptable, Distinguishable

Examples and Explanations:

Example: A product image on an e-commerce site includes the alt text “Red cotton t-shirt with round neck.”
Explanation: All informative images must have descriptive alt text so users who rely on screen readers can understand the visual content.

Example: A website uses dark blue text on a white background to ensure readability.
Explanation: To assist users with visual impairments, the contrast ratio between text and background must be at least 4.5:1 for normal text and 3:1 for large text.

Example: A blog post uses <h1> for the title, <h2> for main sections, and <h3> for sub-sections.
Explanation: A logical heading structure helps screen reader users understand and navigate the page efficiently.

Example: An educational video includes captions that display spoken content and relevant sounds, such as “(applause)” or “(door slams)”.
Explanation: Captions must be accurate and synchronized so users who are deaf or hard of hearing can fully understand the video content.

Example: A web form marks required fields with both red text and an asterisk symbol (*) next to the label.
Explanation: Information conveyed by color must also be provided in another form so that users with color blindness can perceive it.

1.2.2 Principle: Operable

This subsection introduces learners to *operability*. All users must be able to interact with and navigate the website using various input methods.

This includes:

- **Keyboard accessibility** for users who cannot use a mouse,
- **Sufficient time** to complete tasks for users with slower response times,
- **Visible focus indicators** for orientation.

This principle supports users with motor impairments (e.g., paralysis), temporary limitations (e.g., broken arm), and situational restrictions (e.g., one-handed use while holding a child).

Definition: Interface elements and content must be operable by all users, regardless of input method. Interfaces should accommodate diverse physical and cognitive abilities, including alternative input devices and assistive technologies.

Guidelines: Keyboard Accessible, Enough Time, Seizures and Physical Reactions, Navigable, Input Modalities

Examples and Explanations:

Example: A website allows users to open menus, fill out forms, and activate buttons entirely using the Tab, Enter, and Arrow keys.
Explanation: All interactive components must be fully operable via keyboard so that users who cannot use a mouse can still navigate and interact with the site.

Example: An online quiz provides an option to extend the session time before it expires.
Explanation: Users must be given enough time to read, understand, and interact with content, especially those with cognitive or motor impairments.

Example: A landing page avoids using flashing banner ads or animations that blink more than three times per second.
Explanation: Content that flashes rapidly can trigger seizures in users with photosensitive epilepsy and must therefore be avoided or strictly limited.

Example: A website features a consistent navigation menu at the top of each page and uses clear section headings throughout.
Explanation: Clear and logical navigation structures help users, especially those using screen readers or with cognitive disabilities and find content more easily.

1.2.3 Principle: Understandable

This subsection focuses on *understandability*. Both the content and the interface must be easy to understand and predictable.

This includes:

- Readable, logically structured content,
- Simple and clear instructions,
- Helpful error messages for input fields.

This supports users with cognitive impairments (e.g., learning disabilities), as well as those experiencing temporary stress or navigating in a foreign language.

Definition: Users must be able to understand the information presented and how to interact with the interface. The complexity must not exceed the users' abilities.

Guidelines: Readable, Predictable, Input Assistance

Examples and Explanations:

Example: A government service website uses short sentences and avoids technical jargon when explaining how to apply for a permit.
Explanation: Using plain and simple language improves comprehension for users with cognitive disabilities, low literacy, or for whom the language is not native.

Example: An online shop places the shopping cart icon in the same top-right position on every page and keeps the checkout process consistent.
Explanation: Consistent and predictable interfaces help users understand how to navigate digital environments and reduce confusion or errors.

Example: A login form shows placeholder text like “Enter your email address” and provides a clear message if the input is incorrect: “Please enter a valid email in the format name@example.com.”
Explanation: Clear input guidance and meaningful error messages help users complete tasks successfully and recover from mistakes without frustration.

1.2.4 Principle: Robust

This subsection introduces the concept of *robustness*. Robust content is built using clean, standard-compliant code that functions across different browsers, platforms, and assistive technologies (e.g., screen readers, Braille displays).
This principle supports long-term accessibility by ensuring compatibility across older and newer technologies.

Definition: Content must be robust enough to be reliably interpreted by a wide variety of user agents, including assistive technologies. This ensures ongoing accessibility as technology evolves.

Guideline: Compatible

Examples and Explanations:

Example: A university website is tested and displays correctly in Chrome, Firefox, Safari, and Edge without layout or functionality issues.
Explanation: Compatibility with multiple browsers is essential to ensure that all users—regardless of their preferred browser—can access and interact with content reliably.

Example: A development team uses browser developer tools and tests their app on both a smartphone and a tablet to check usability at 320×768 and other screen sizes.
Explanation: Mobile devices must be included in accessibility testing to ensure that responsive designs work well across various screen sizes and input methods, supporting all users on all platforms.

2. User Groups

⏱ Estimated class time: approx. 90 minutes

🎯 **Learning Objectives of the Chapter**

LO 2.1.1 – Recall different types of blindness. – K1

LO 2.1.2 – Describe software requirements and user needs of blind individuals. – K2

LO 2.1.3 – Identify assistive technologies and their functions for blind users. – K1

LO 2.2.1 – Recall different types of visual impairments. – K1

LO 2.2.2 – Describe software requirements and user needs of individuals with visual impairments. – K2

LO 2.2.3 – Identify assistive technologies and their functions for users with visual impairments. – K1

LO 2.3.1 – Recall different types of hearing impairments. – K1

LO 2.3.2 – Describe software requirements and user needs of individuals with hearing impairments. – K2

LO 2.3.3 – Identify assistive technologies and their functions for users with hearing impairments. – K1

LO 2.4.1 – Recall different types of motor impairments. – K1

LO 2.4.2 – Describe software requirements and user needs of individuals with motor impairments. – K2

LO 2.4.3 – Identify assistive technologies and their functions for users with motor impairments. – K1

LO 2.5.1 – Recall different types of cognitive impairments. – K1

LO 2.5.2 – Describe software requirements and user needs of individuals with cognitive impairments. – K2

LO 2.5.3 – Identify assistive technologies and their functions for users with cognitive impairments. – K1

This chapter introduces different user groups and their specific requirements for digital accessibility. It is important to note that digital barriers can affect everyone. The categorization is based on types of impairments. Each section focuses on the specific needs and assistive technologies used to improve software accessibility.

Terms

Blind users, visually impaired users, users with hearing impairments, users with motor impairments, users with cognitive impairments, assistive technologies

2.1 Blind Users

Terms: Blind users, total blindness, functional blindness, legal blindness, residual vision, screen reader, braille display, speech recognition systems, audio description

Definition:
Blind users, in the context of the Web Content Accessibility Guidelines (WCAG), are individuals who cannot perceive website content visually and therefore rely on alternative sensory channels (most often hearing or touch) to access information.

Important:
The term “blind” does not exclusively refer to individuals with complete vision loss.
It also includes users who have minimal or non-functional residual vision and who are unable to use websites visually in a practical sense.

2.1.1 Different Types of Blindness

Blindness varies in form and degree. In medical and legal contexts, as well as in the field of accessibility (e.g., WCAG), different types of blindness are recognized. Within the WCAG framework, the term “blind” is understood functionally: the key factor is not the medical diagnosis, but whether visual access to a website is practically possible.

1. Total Blindness (Amaurosis)

- **Definition:** No visual perception; individuals cannot detect light, shapes, or motion.
- **Consequence:** Complete reliance on other senses such as touch, hearing, or smell.
- **Typical use of technology:** Screen readers, braille displays, voice assistants.

2. Functional Blindness (Practical Blindness)

- **Definition:** Minimal residual visual functions (e.g., light perception or vague outlines), but not sufficient to read or recognize content.
- **Consequence:** Users typically rely on assistive tools like screen readers due to lack of visual usability.
- **Typical example:** Only able to perceive light and dark contrasts, without detail.

3. Legal Blindness – Definitions in English-Speaking Countries

Please Note:
Although terminology and exact measurements may vary, all definitions focus on severely limited central vision and/or peripheral vision. This functional understanding aligns with the WCAG approach, which emphasizes whether visual access to content is practically possible.

United States

- **Term:** *Legally blind*
- **Definition:** Visual acuity of 20/200 or worse in the better eye with best correction, or a visual field of 20 degrees or less.
- **Purpose:** Determines eligibility for disability benefits and accommodations (e.g., Social Security).

United Kingdom

- **Term:** *Severely sight impaired* (equivalent to legal blindness)
- **Definition:** Visual acuity of less than 3/60 with full visual field, or 3/60 to 6/60 with a very restricted field (e.g., tunnel vision).
- **Purpose:** Used for official registration to access support services (e.g., NHS, local authorities).

Canada

- **Term:** *Legally blind*
- **Definition:** Similar to the U.S.: visual acuity of 20/200 or worse in the better eye with best correction, or a visual field of 20 degrees or less.
- **Purpose:** Used for eligibility for federal and provincial disability programs and tax benefits.

Australia

- **Term:** *Legally blind*
- **Definition:** Visual acuity of 6/60 or worse in the better eye with best correction, or a visual field of less than 20 degrees. *(Note: 6/60 is the metric equivalent of 20/200.)*
- **Purpose:** Eligibility for services and benefits under the National Disability Insurance Scheme (NDIS) and other programs.

New Zealand

- **Term:** *Legally blind* (used informally), or *severely vision impaired*
- **Definition:** Visual acuity of 6/60 or worse, or severely limited visual field (20 degrees or less).
- **Purpose:** Used for accessing vision services and disability support through organizations like Blind Low Vision NZ.

Consequence: Entitlement to certain social benefits or assistive equipment.
Note: Legal blindness allows for the presence of some residual vision.

4. Residual Vision (Severe Visual Impairment)

- **Definition:** Individuals retain some functional vision, such as the ability to recognize large objects or navigate using high-contrast environments.
- **Distinction:** These users are typically not considered “blind,” but still face major challenges using websites visually without support.
- **Common technical support:** Screen magnification software, high-contrast displays, personalized settings.

Summary:

- **Totally blind users:** No vision.
- **Functionally blind users:** Minimal residual functions that make visual use impossible.
- **Legally blind users:** Defined visual thresholds indicating severe vision loss.
- **Users with residual vision:** Partial vision is usable in some contexts, but with significant limitations.

2.1.2 Requirements and Needs for Software Used by Blind Users

Blind users have specific software requirements due to their lack of visual perception. Software must be designed so that all content and functions are fully accessible and operable without the need to see.

Key Requirements:

1. **Screen Reader Compatibility**
   - All content must be correctly recognized and read aloud by screen readers.
   - Proper use of semantic markup for headings, lists, tables, and forms is essential.
2. **Keyboard Operability**
   - All functions must be fully and efficiently usable via keyboard only.
   - A logical and predictable tab order is crucial.
3. **Meaningful Alternative Texts**
   - Visual elements (e.g., images, icons) must include descriptive alt text.
   - Decorative images should be marked accordingly to avoid unnecessary interruptions.
4. **Clear and Logical Structure**
   - Content must follow a clear hierarchy.
   - Navigation should be consistent, logical, and easy to learn.
5. **No Visual Dependency**
   - Information must not rely on color, shape, or screen position alone.
   - Additional text or audio cues are necessary.

6. **Error Prevention and Detection**
    - Forms should provide accessible error messages that screen readers can detect and announce.
7. **No Time-Limited Actions Without Alternatives**
    - Time limits should be avoidable or extendable.
    - Time-sensitive content must be compatible with assistive technologies.

Core Needs of Blind Users:

- Independent use without external assistance
- Fast orientation and efficient access to functions
- Clear and reliable feedback from the system
- Accessible documentation and help content compatible with screen readers

2.1.3 Assistive Technologies and Their Purpose for Blind Users

Definition:
Assistive technologies are specialized hardware and software solutions that help people with disabilities access digital content and use systems. For blind users, they replace or support visual perception through alternative sensory input.

Key Assistive Technologies for Blind Users:

1. **Screen Readers**
    - **Function:** Convert screen content (text, structure, controls) into speech or Braille output
    - **Examples:** JAWS, NVDA, VoiceOver (Apple), TalkBack (Android)
    - **Purpose:** Enable blind users to operate websites, apps, documents, and operating systems independently
2. **Braille Displays**
    - **Function:** Display text content as tactile Braille output
    - **Purpose:** Allow precise and discreet reading via touch
3. **Speech Recognition Systems**
    - **Function:** Enable device control and text input through spoken commands
    - **Examples:** Dragon NaturallySpeaking, built-in systems in Windows and macOS
    - **Purpose:** Support productivity and hands-free interaction
4. **Screen Magnifiers** *(for users with limited residual vision)*
    - **Function:** Enlarge and enhance screen content for easier visibility
    - **Examples:** ZoomText, MAGic

  - **Purpose:** Assist users with some vision, including those with severe visual impairments
5. **Audio Descriptions and Sound Output**
  - **Function:** Provide spoken descriptions of visual content (e.g., videos, interface elements)
  - **Purpose:** Make complex visual information accessible

Purpose of Assistive Technologies for Blind Users:

- Independent use of digital media, software, and devices
- Equal participation in communication, education, work, and leisure
- Full access to information and interaction
- Overcoming visual barriers through hearing, touch, or speech input

Summary:
Assistive technologies are essential tools that empower blind users to access and operate digital systems effectively, efficiently, and independently.
Software developers must ensure full compatibility with these technologies.

2.2 Users with Visual Impairments

Key Terms: Users with visual impairments, color vision deficiencies, reduced contrast sensitivity, photophobia, visual field loss, screen magnification, color adjustment, text highlighting

According to the Web Content Accessibility Guidelines (WCAG), users with visual impairments are individuals who have limited vision but are not completely blind. They are able to perceive content visually but require specific adjustments or assistive tools.

2.2.1 Types of Visual Impairments

Common types of visual impairments include:

- **Low Vision:**
  Severely reduced visual acuity even with corrective lenses. Details are difficult to perceive.
- **Color Vision Deficiencies:**
  Difficulty distinguishing certain colors or color contrasts (e.g., red-green color blindness, blue blindness).

- **Reduced Contrast Sensitivity:**
  Trouble perceiving distinctions between foreground and background colors, especially with low contrast.
- **Photophobia (Light Sensitivity):**
  Difficulty seeing in bright or glaring conditions; light causes discomfort or impairs vision.
- **Visual Field Loss:**
  Limited field of view, such as tunnel vision or central blind spots (scotomas).
- **Age-Related Vision Changes:**
  Decreased vision due to aging conditions such as macular degeneration or cataracts.

2.2.2 Requirements and Needs for Software for Users with Visual Impairments

Users with visual impairments can perceive digital content but require tailored design to access and use it effectively. Needs vary depending on the type and severity of the impairment but aim to ensure usability and comfort.

1. **Scalable Content**
   - Text, images, and UI elements must be resizable without quality loss (up to 200% or more).
   - Layouts should remain readable and not overlap when zoomed.
   - Font sizes should be adjustable without forcing horizontal scrolling.
2. **High Contrast and Color Control**
   - Content should have a minimum contrast ratio of 4.5:1 for standard text (per WCAG 2.1 AA).
   - Users should be able to switch to high contrast or dark mode as needed.
   - Color differences should be clearly distinguishable.
3. **No Color-Only Information**
   - Visual cues should not rely on color alone (e.g., red for error).
   - Supplement color with icons, patterns, or text descriptions.
   - Example: Error fields should have both a red border and a text error message.
4. **Clear and Simple Design**
   - Layouts should be clean, with well-structured sections and clearly labeled controls.
   - A consistent hierarchy of headings and navigation aids orientation.
   - Avoid unnecessary animation or visual clutter.
5. **Flexible Customization**
   - Users must be able to adjust font sizes, contrast, and color schemes.

- Software should respect system-wide settings like high contrast or large text.

6. **Avoid Visual Overload**
   - Content should not appear, disappear, or move unexpectedly.
   - Moving elements (e.g., sliders) must be pauseable or turned off, per WCAG 2.2.2 “Pause, Stop, Hide”.

Summary:
Users with visual impairments need scalable, high-contrast, clearly structured, and customizable content. WCAG-compliant design ensures readability, flexibility, and access through multiple visual and non-visual channels.

2.2.3 Assistive Technologies and Their Purpose for Users with Visual Impairments

Assistive technologies for users with visual impairments are hardware and software tools that support or replace visual perception, enabling better recognition, interaction, and access to digital content.

Key Assistive Technologies:

1. **Screen Magnifiers**
   - **Function:** Magnify portions of the screen for detailed viewing
   - **Examples:** ZoomText, Windows Magnifier, macOS Zoom
   - **Purpose:** Aid reading and navigation with limited visual acuity
2. **Built-In OS Accessibility Features**
   - **Function:** Operating systems offer settings for high contrast, larger text, and custom display options
   - **Examples:**
     - *Windows:* Display settings (text scaling, color filters)
     - *macOS:* Accessibility features (increase contrast, invert colors)
     - *Android/iOS:* Magnification gestures, color inversion, adjustable font sizes
   - **Purpose:** Allow personalized device use
3. **Text Highlighting and Reading Tools**
   - **Function:** Visually emphasize text or read it aloud slowly
   - **Examples:** ClaroRead, Kurzweil 3000
   - **Purpose:** Help users follow and comprehend lengthy or complex text
4. **Color Adjustment Tools**
   - **Function:** Adjust color schemes to improve contrast or visibility for color-blind users

- **Examples:**
  - System-based tools like grayscale, inversion, and color filters (Windows, macOS, Android, iOS)
  - Browser plugins for high contrast modes
- **Purpose:** Improve visual perception in users with color sensitivity

5. **On-Demand Speech Output (Screen Reader Functions)**
   - **Function:** Read out content for users with both visual and reading challenges
   - **Examples:** VoiceOver (Apple), Narrator (Windows)
   - **Purpose:** Combine visual and auditory access to information

Purpose of Assistive Technologies for Visually Impaired Users:

- Make content visible and readable that would otherwise be difficult to see
- Improve readability through contrast, color, and text scaling
- Enable personal adaptation to individual vision needs
- Reduce visual strain through clear highlights and alternative outputs
- Promote independent access to education, work, and everyday communication

Summary:
Assistive technologies are vital for users with visual impairments to effectively access and use digital content. Accessible software must support and be fully compatible with these tools.

2.3 Users with Auditory Impairments

Key Terms: Auditory impaired users, deafness, hearing loss, APD (Auditory Processing Disorder), tinnitus, captions, transcripts, sign language, visual alerts

Auditory impaired users are individuals who have reduced hearing or cannot hear at all. According to the Web Content Accessibility Guidelines (WCAG), this group includes people with total hearing loss, partial hearing loss, and auditory processing disorders.

WCAG Objective:
WCAG ensures that content relying on sound or spoken language is also perceivable visually or through alternative channels. All essential information must be available independently of the auditory sense.

2.3.1 Types of Auditory Impairments

Auditory impairments vary widely in type and severity and directly affect access to speech, sound signals, and music.

Types of Auditory Impairment:

1. **Total Deafness**
   - No or minimal perception of sound.
   - Speech, music, and environmental sounds are not accessible or only partially perceived.
2. **Partial Hearing Loss**
   - Reduced hearing ability at specific volumes or frequencies.
   - Sounds may be distorted or perceived incompletely.
3. **Age-Related Hearing Loss (Presbycusis)**
   - Gradual loss of high-frequency hearing due to aging.
   - Difficulty understanding high-pitched or fast speech.
4. **Auditory Processing Disorder (APD)**
   - Normal hearing ability, but impaired processing of auditory information in the brain.
   - Speech comprehension is difficult, especially in noisy environments.
5. **Single-Sided Deafness**
   - Hearing loss in one ear, normal hearing in the other.
   - Impairs spatial sound localization and understanding speech in noise.
6. **Tinnitus**
   - Perception of sounds like ringing or buzzing without an external source.
   - Can interfere with concentration and hearing real audio content.
   - May occur alone or alongside other hearing loss.

Note:
While tinnitus is not technically hearing loss, it affects auditory perception and may increase the need for accessible formats such as clear audio and visual alternatives.

2.3.2 Requirements and Needs for Software for Users with Auditory Impairments

Deaf and hard-of-hearing users, or those with auditory processing challenges, require software that provides accessible alternatives to sound-based content.

Key Requirements and Needs:

- **Captions and Transcripts:**
  Spoken audio (e.g., videos, podcasts, webinars) must include accurate captions or full transcripts, including relevant sound cues.
- **Sign Language Support:**
  Sign language interpretation or videos should be provided, especially for live formats and public services.
- **Visual Alternatives to Sound:**
  Notifications, alerts, or sound cues must also be shown visually (e.g., text, symbols, or color changes).
- **Text-Based Communication:**
  Support for text chat, written support channels, and real-time captions in meetings is essential.
- **Adjustable Audio Playback:**
  Users should be able to adjust volume, separate audio tracks (e.g., voice and music), and reduce background noise.
- **Clear Speech Output:**
  Spoken content must be slow, clear, and easy to understand; synthesized voices should be natural and intelligible.
- **Non-Audio Operation:**
  Navigation and feedback must be fully usable without sound, relying on clear visual indicators.
- **Avoidance of Auditory Overload:**
  Complex or overlapping audio should be avoidable or adjustable (e.g., playback speed control).
- **Full Information Without Hearing:**
  No information should be conveyed through audio only; all content must be accessible in written or visual form.

Summary:
Auditory impaired users need text-based, visual, or sign language alternatives to audio content. Accessibility depends on well-planned, multi-channel communication that ensures independence and equal participation.

2.3.3 Assistive Technologies and Their Purpose for Auditory Impaired Users

Assistive technologies for auditory impaired users convert sound-based information into visual or tactile formats. They support individuals who rely on alternative channels due to hearing loss or auditory processing difficulties.

Key Assistive Technologies and Their Purpose:

1. **Captioning Software**
   - **Function:** Provides synchronized text for spoken content in videos or live sessions.
   - **Examples:** YouTube auto-captions, Microsoft Teams captions, live event captioning
   - **Purpose:** Makes spoken content fully accessible without sound
2. **Transcription Services**
   - **Function:** Convert audio files or recordings into written text
   - **Examples:** Otter.ai, Google Recorder, manual transcription for formal materials
   - **Purpose:** Offers full access to spoken information
3. **Sign Language Avatars and Interpreters**
   - **Function:** Translate spoken or written content into sign language
   - **Examples:** On-screen sign language avatars, live interpretation in video calls or events
   - **Purpose:** Supports users who primarily use sign language
4. **Visual Alert Systems**
   - **Function:** Replace audio alerts (e.g., ringtones, alarms) with visual or tactile signals
   - **Examples:** Flashing screen icons, pop-ups, LED lights, phone vibrations
   - **Purpose:** Ensures important notifications are not missed
5. **Speech-to-Text Systems and CART Services**
   - **Function:** Convert speech to text in real time, often used in lectures or meetings
   - **Examples:** Live captioners for educational settings, real-time meeting captions (e.g., Zoom)
   - **Purpose:** Enables participation without needing to hear
6. **Hearing Loops and Audio Transmission Systems**
   - **Function:** Wirelessly transmit audio to compatible hearing aids
   - **Examples:** Induction loops in public venues like theaters or banks
   - **Purpose:** Enhances speech clarity and reduces background noise
7. **Accessibility and Communication Apps**
   - **Function:** Support communication using text or visual display of speech
   - **Examples:** Ava (real-time conversation captions), RogerVoice (captioned phone calls)
   - **Purpose:** Improve daily communication and promote independence

Summary:
Assistive technologies for auditory impaired users ensure equal access to communication and information by converting sound into visual or tactile forms. They play a vital role in digital inclusion, autonomy, and participation in all areas of life.

2.4 Users with Motor Impairments

Key Terms: Users with motor impairments, fine motor skills, muscle tone, tremors, spasticity, paralysis, speech control, eye tracking, switch access, joystick

Users with motor impairments are individuals who have difficulty operating digital devices or software due to physical limitations. These impairments may affect mobility, strength, coordination, or fine motor control and can be either congenital or acquired.

In the context of the Web Content Accessibility Guidelines (WCAG), motor impairments are considered relevant factors that impact a person's ability to effectively use input devices such as keyboards, mice, or touchscreens. Accessible design must ensure that digital content can be operated flexibly, with tolerance for input errors, and without requiring complex motor actions.

2.4.1 Types of Motor Impairments

The following types of motor impairments should be considered in accessible digital design:

1. **Fine Motor Limitations**
    - Difficulty with precise movement of small muscle groups (e.g., hands, fingers)
    - Challenges in using mouse pointers, small buttons, or touchscreens
2. **Reduced Strength or Muscle Tone**
    - Inability to perform physical inputs with enough force (e.g., holding keys, pressing buttons)
    - Common in neuromuscular conditions, muscular dystrophy, or age-related decline
3. **Tremors**
    - Involuntary shaking of hands or arms makes precise input difficult
    - Can lead to accidental clicks, keypresses, or swipes
4. **Spasticity and Uncontrolled Movements**
    - Sudden, involuntary muscle contractions hinder controlled movements
    - Input devices may be difficult to use steadily or without errors
5. **Paralysis and Partial Paralysis (e.g., hemiparesis, paraplegia)**
    - Loss of movement control in specific body parts
    - Input may be limited to one hand or require assistive tools

6. **Amputations or Congenital Limb Differences**
   - Missing or non-functional limbs require use of alternative input methods (e.g., speech, head tracking)
7. **Fatigue-Related Motor Impairments**
   - Conditions like multiple sclerosis or rheumatoid arthritis may cause rapid fatigue during use
   - Interfaces should minimize steps and effort
8. **Coordination Disorders (e.g., ataxia)**
   - Difficulty with precise and coordinated movement
   - Makes tasks like selecting small buttons or scrolling accurately more difficult

Note:
Motor impairments can be permanent, temporary (e.g., after injury), or situational (e.g., operating a device in challenging environments). Accessible design must therefore support flexible input methods, minimize physical effort, and prevent errors.

2.4.2 Requirements and Needs for Software for Users with Motor Impairments

Users with motor impairments have varied needs depending on the nature and severity of their condition. The goal of accessible design is to enable precise control, low physical effort, error tolerance, and alternative input options.

1. **Support Alternative Input Methods**
   - Full keyboard accessibility without requiring a mouse
   - Compatibility with voice control, eye tracking, switch access, head control, or joystick
   - Support for on-screen keyboards and adaptive hardware
   - Consideration for users operating with only one button or simple switch systems
2. **Large and Clearly Labeled Controls**
   - Large, easy-to-reach buttons that work even with imprecise movements
   - Adequate spacing between elements to prevent accidental input
   - Visual focus indicators to support orientation
3. **Minimize Required Input Actions**
   - Reduce the number of clicks or steps needed to complete tasks
   - Support auto-fill, saved preferences, and pre-selected options
   - Provide shortcuts and fast navigation paths
4. **Error-Tolerant Design**

   - Easy correction of input errors (e.g., “undo” functionality, no immediate submission)
   - Require confirmation for critical actions (e.g., deleting or purchasing)
   - Avoid multi-step forms without autosave
5. **Customizable Interfaces**
   - Adjustable element sizes and text scaling
   - Simplified menus, reduced content, and alternate layouts
   - Configurable UI elements (e.g., reorderable menus, quick access features)
6. **Flexible Time Controls**
   - Extendable or removable time limits for forms, logins, and tasks
   - Allow users to acknowledge alerts or pop-ups at their own pace
7. **Reduce Complex Gestures**
   - Avoid mandatory multi-touch or drag-and-drop gestures
   - Provide simple alternatives (e.g., single click or keyboard commands)
8. **Low Physical Effort**
   - No need for prolonged keypresses, double clicks, or high pressure input
   - Allow interaction through tapping or single activation
9. **Visual Feedback and Clear Status Indicators**
   - All actions should trigger clear visual feedback
   - Confirmations should be prominent and understandable
10. **Pause and Resume Capabilities**
    - Long tasks (e.g., forms) should support saving progress
    - Users must be able to pause and resume later

Summary:
Users with motor impairments need digital systems that support flexible, error-tolerant, low-effort, and alternative input methods. Accessible interfaces must accommodate individual input capabilities, reduce physical demands, and avoid complex interaction patterns.

2.4.3 Assistive Technologies and Their Purpose for Users with Motor Impairments

Assistive technologies for motor-impaired users include devices and software that help enable or simplify interaction with digital systems despite reduced mobility or coordination. The aim is to ensure barrier-free, independent, and efficient operation.

Key Assistive Technologies and Their Purpose:

1. **Alternative Keyboards**

- **Function:** Replace standard keyboards with customized versions (e.g., large keypads, one-handed keyboards, on-screen keyboards)
- **Examples:**
  - Large-key keyboards for users with tremors
  - One-handed keyboards for users with hemiparesis or amputations
- **Purpose:** Enable accurate input with less physical strain

2. **Mouse Alternatives**
   - **Function:** Replace traditional mice with specialized devices
   - **Examples:**
     - Joysticks for pointer control
     - Trackballs with large, easy-to-move balls
     - Head-controlled mouse movement
   - **Purpose:** Improve navigation and clicking for limited hand function
3. **Switch Access**
   - **Function:** Operate devices through simple switch signals
   - **Examples:**
     - Single or multiple switches activated by touch, tongue, or breath
   - **Purpose:** Enable input through minimal motion, helpful in cases of severe paralysis or neuromuscular conditions
4. **Speech Recognition Systems**
   - **Function:** Allow users to control devices and enter text using voice commands
   - **Examples:** Siri, Google Assistant, Dragon NaturallySpeaking
   - **Purpose:** Full interaction without physical input; used for navigation and text entry
5. **Eye Tracking Systems**
   - **Function:** Use eye movement to control the pointer or select on-screen elements
   - **Examples:** Tobii Eye Tracker
   - **Purpose:** Full device control via eye movement, especially useful for users with complete paralysis (e.g., ALS)
6. **On-Screen Keyboards**
   - **Function:** Display a virtual keyboard that can be operated using alternative inputs or eye tracking
   - **Examples:** Windows On-Screen Keyboard, gaze-interaction tools
   - **Purpose:** Enter text without a physical keyboard
7. **Auto-Correction and Input Assistance**
   - **Function:** Help users avoid and fix typing errors
   - **Examples:** Auto-complete, predictive text
   - **Purpose:** Reduce input effort and compensate for inaccuracy
8. **Customizable Interfaces and Assistive Software**

- **Function:** Enable personalization, such as larger controls, simplified navigation, or remapped commands
- **Examples:** Accessibility settings in Windows, macOS, and mobile systems
- **Purpose:** Increase accessibility through reduced complexity and user-specific customization

9. **Macros and Automation Tools**
    - **Function:** Combine common actions into a single command
    - **Examples:** AutoHotKey, Shortcuts apps
    - **Purpose:** Reduce required input and physical movements

Summary:
Assistive technologies for users with motor impairments enable flexible control, easy text input, and full navigation of digital systems with minimal physical effort. These tools are essential for independent, efficient, and accurate access to digital content despite physical limitations.

2.5 Users with Cognitive Impairments

Key Terms: Cognitive impairments, learning disabilities, executive functions, plain language, text-to-speech software, symbol systems, focus aids, step-by-step assistants

In the context of the Web Content Accessibility Guidelines (WCAG), the term *cognitive impairment* refers to a wide range of limitations that affect how individuals receive, process, and apply information. These may involve attention, memory, language processing, problem-solving, numerical understanding, or social interaction.

Cognitive impairments can be permanent (e.g., intellectual disabilities), temporary (e.g., due to stress or fatigue), or situational (e.g., under time pressure or distraction). The WCAG addresses this variety by requiring content to be clear, structured, and easy to understand.

To support users with cognitive impairments, WCAG includes specific provisions such as clear content structure, understandable language, predictable user processes, input assistance, and the avoidance of unnecessary complexity. The goal is to reduce cognitive load and enable independent use of digital content.

2.5.1 Types of Cognitive Impairments

Cognitive impairments affect the ability to receive, process, store, and apply information.

In WCAG, cognitive impairments include all conditions that make it harder to understand, learn, remember, solve problems, focus attention, or communicate. The goal of WCAG is to ensure accessible information and digital interaction for all users, regardless of cognitive ability.

Typical types of cognitive impairments include:

1. **Learning Disabilities**
   - Difficulties with reading, writing, or math (e.g., dyslexia, dyscalculia)
   - Trouble understanding complex texts or interpreting symbols
2. **Intellectual Disabilities**
   - Reduced cognitive ability and slower learning (e.g., due to genetic conditions or early brain injuries)
   - Increased need for simplified language and structured content
3. **Memory Impairments**
   - Difficulty storing and recalling information (e.g., due to dementia or stroke)
   - Require repetition, orientation aids, and easy-to-retrieve content
4. **Attention and Concentration Disorders**
   - Limited ability to focus for extended periods (e.g., ADHD)
   - Low tolerance for distraction or overstimulation
5. **Executive Function Disorders**
   - Difficulty with planning, organizing, and adapting to changes
   - Complex navigation and unpredictable workflows are problematic
6. **Autism Spectrum Disorders**
   - Challenges with social interaction and sensory processing
   - Clear structure, predictability, and low ambiguity are essential
7. **Speech and Language Disorders**
   - Trouble understanding or producing spoken language (e.g., aphasia)
   - Need for plain language and visual support
8. **Mental Health Conditions (e.g., anxiety, depression)**
   - May impact cognitive performance, especially focus, decision-making, and stress tolerance

Note:
Cognitive impairments are diverse and often invisible. Accessible design must focus on ease of use, plain language, reduced complexity, and predictable processes to be inclusive.

Summary:
Cognitive impairments under WCAG include challenges in understanding, remembering, learning, communicating, and solving problems. Accessibility requires simple language, clear structure, low cognitive load, and supportive features to ensure digital content is usable by all.

2.5.2 Requirements and Needs for Software for Users with Cognitive Impairments

Users with cognitive impairments need digital environments that are clear, understandable, easy to navigate, and forgiving of mistakes.

The main goals are to reduce cognitive load, increase comprehension, and provide flexible support.

Key Requirements and Needs:

1. **Simple and Clear Language**
   - Use short, active sentences and plain words
   - Avoid jargon, abbreviations, and complex phrasing
   - Provide explanations for unfamiliar terms
2. **Offer Plain Language Versions**
   - Provide essential content in simplified language using defined rules
   - Support with images, symbols, or simple graphics
   - Clearly labeled links to plain language content
3. **Clear and Logical Structure**
   - Use headings, lists, and paragraphs to organize content
   - Break information into small, understandable chunks
   - Maintain consistent layout for navigation and controls
4. **Predictable and Consistent Navigation**
   - Keep navigation in the same place across pages
   - Avoid sudden layout changes or unexpected behaviors
   - Use clear aids like “Back” buttons and breadcrumb trails
5. **Error-Tolerant Design**
   - Make input errors easy to identify and fix
   - Provide friendly, solution-oriented error messages
   - Require confirmation for important actions
6. **Reduce Cognitive Load**
   - Avoid information overload
   - Focus on one task per screen
   - No distracting animations or auto-playing media

7. **Visual Content Support**
    - Use supporting graphics, symbols, or videos
    - Offer alternative ways to present key information (e.g., visual summaries)
8. **Customizable Interface**
    - Allow text resizing, contrast adjustment, and content simplification
    - Provide simplified layouts or “easy mode” options
9. **Task Support Features**
    - Use step-by-step guides, progress indicators, and next-step hints
    - Include help tools that can be revealed when needed
10. **Low Time Pressure**

- Avoid session timeouts or time limits without the option to extend
- Allow users to work at their own pace

11. **Emotional Safety**

- Use positive feedback for successful actions
- Avoid alarming error messages or confusing instructions
- Use friendly and encouraging wording

12. **Offer Multiple Content Formats**

- Provide text, audio, video, and image-based formats
- Allow tasks to be completed in different ways (e.g., dropdowns instead of text input)

Summary:
Software for users with cognitive impairments must be simple, clear, forgiving, adaptable, and supportive. Plain language is especially important to ensure that all users can understand and interact with content independently.

2.5.3 Assistive Technologies and Their Purpose for Users with Cognitive Impairments

Assistive technologies for users with cognitive impairments are tools that support information access, comprehension, concentration, task completion, and communication.

They help people with thinking, learning, memory, and problem-solving challenges to understand and use digital content more easily.

Key Assistive Technologies and Their Purpose:

1. **Text-to-Speech (TTS) Software**
    - **Function:** Converts written content into spoken language
    - **Examples:** Natural Reader, VoiceOver (Apple), ReadSpeaker
    - **Purpose:** Aids understanding, especially for users with reading difficulties or attention issues
2. **Simplified Interfaces and Modes**
    - **Function:** Minimize visual complexity and distractions
    - **Examples:** Easy modes in operating systems, simplified web versions
    - **Purpose:** Prevent overwhelm and help maintain focus
3. **Plain Language Tools**
    - **Function:** Provide content in plain or easy-to-read versions
    - **Examples:** Language simplification tools, dedicated plain-language pages
    - **Purpose:** Improve comprehension of complex information
4. **Picture-Based Communication Tools (Symbol Systems)**
    - **Function:** Use images or symbols to replace or support text and speech
    - **Examples:** Metacom symbols, Boardmaker
    - **Purpose:** Help users with limited language skills understand and communicate
5. **Focus and Concentration Aids**
    - **Function:** Reduce distractions or structure time usage
    - **Examples:** Forest, Pomodoro timers
    - **Purpose:** Assist users in staying on task and managing attention
6. **Adjustable Display Options**
    - **Function:** Let users change font size, spacing, colors, and contrast
    - **Examples:** OpenDyslexic extensions, reader modes in browsers
    - **Purpose:** Improve readability for users with dyslexia or visual processing issues
7. **Interactive Step-by-Step Assistants**
    - **Function:** Guide users through tasks in clearly defined steps
    - **Examples:** Wizard forms, interactive tutorials
    - **Purpose:** Minimize errors and support task completion
8. **Visual Structuring Tools**
    - **Function:** Visualize ideas, processes, or tasks using diagrams or lists
    - **Examples:** MindMeister, digital checklists
    - **Purpose:** Enhance understanding and maintain clarity

Summary:

Assistive technologies for users with cognitive impairments support understanding, memory, task management, communication, and focus. They empower users to work and communicate independently, confidently, and without unnecessary stress.

3. Legal Requirements, Guidelines, Standards and Regulations for Digital Accessibility – International Overview

⏱️ Estimated Class Time: approx. 90 minutes

🎯 Learning Objectives of the Chapter

LO 3.1.1 – Recall key accessibility laws and technical standards relevant in the EU and in English-speaking countries. – K1

LO 3.1.2 – Classify the 13 WCAG guidelines under the four principles of accessibility: Perceivable, Operable, Understandable, and Robust. – K2

LO 3.2.1 – Recall the UK Public Sector Bodies (Websites and Mobile Applications) Accessibility Regulations 2018 and the Equality Act 2010. – K1

LO 3.2.2 – Recall Section 508 of the Rehabilitation Act and the Americans with Disabilities Act (ADA) as key U.S. accessibility laws. – K1

LO 3.2.3 – Recall the Accessible Canada Act (ACA) and relevant provincial laws such as the Accessibility for Ontarians with Disabilities Act (AODA). – K1

LO 3.2.4 – Recall the purpose of the Trusted Tester Program and how it is used to evaluate WCAG conformance in U.S. federal digital services. – K1

LO 3.2.5 – Explain the purpose of the WCAG-EM (Web Content Accessibility Guidelines – Evaluation Methodology) – K2

LO 3.2.6 – Recall the commonly applied WCAG versions and conformance levels (e.g., WCAG 2.1 Level AA) in the EU and in English-speaking countries. – K1

Introduction

This chapter provides a structured overview of the most relevant laws, directives, and technical standards related to digital accessibility at both European and international levels. It begins with the legal framework of the European Union, including the Web Accessibility Directive and the European Accessibility Act, and explores how individual member states have implemented these requirements in national law. Special attention is given to countries both inside and outside the DACH region (Germany, Austria, Switzerland), highlighting differences in enforcement, scope, and national standards.

In the second part, the chapter turns to key non-EU countries across North America, Asia, the Commonwealth, and other global regions. These countries have developed their own approaches based on human rights law, others on sectoral regulation, but almost all are guided by the internationally recognized Web Content Accessibility Guidelines (WCAG).

The goal of this chapter is to offer a comparative and comprehensive understanding of how digital accessibility is regulated worldwide, providing a foundation for further exploration of implementation practices, compliance challenges, and inclusive design.

As shown below. **WCAG 2.1 Level AA** is widely recognized as the most practical and consistent foundation for digital accessibility across the globe. Its clear, technology-neutral criteria have been adopted or referenced in legal frameworks throughout the European Union, North America, Asia, and beyond. Because of its broad international support and technical maturity, WCAG 2.1 AA provides reliable guidance for lawmakers, developers, and organizations alike. Aligning with WCAG 2.1 AA not only helps ensure legal compliance but also promotes truly inclusive and user-friendly digital experiences.

3.1.1 Digital Accessibility in the EU – Key Regulations, Standards, and National Laws

1. EU-Wide Legal Framework

Web Accessibility Directive (WAD) – Directive (EU) 2016/2102

- Requires **public sector websites and mobile applications** to be accessible.
- Enforced in all EU member states.
- Requires publication of:
  - An **accessibility statement**
  - A **feedback mechanism**
  - A **monitoring and enforcement process**
- Technical basis: **EN 301 549**, built on **WCAG 2.1 Level AA**

European Accessibility Act (EAA) – Directive (EU) 2019/882

- Applies to the **private sector**, targeting products and services with broad consumer use (e.g., e-commerce, smartphones, ATMs, banking).
- Must be implemented in national law by **28 June 2025**.
- Technical reference: **EN 301 549** and WCAG 2.1 AA

EN 301 549 – European Standard

- Defines accessibility requirements for **ICT products and services**, including:
  - Software
  - Hardware

  - Websites
  - Electronic documents
- Aligned with **WCAG 2.1 AA**, extended with requirements for non-web content and platforms.

2. National Implementation and Oversight

Country	**Main Law / Regulation**	**Applies To**	**Oversight Body**	***Standard***
Germany	Barrierefreiheitsstärkungsgesetz (BFSG), BGG, BITV 2.0	Public sector + private from 2025	BFIT-Bund (formerly BIK), BMAS	*WCAG 2.1 AA, EN 301 549*
Austria	Bundes-Behindertengleichstellungsgesetz (BGStG), WZG	Public sector	Bundeskanzleramt	*WCAG 2.1 AA*
Switzerland	Behindertengleichstellungsgesetz (BehiG), Web Accessibility Standard	Public sector (federal level)	E-Accessibility/Access for All	*WCAG 2.1 AA*
France	RGAA (Référentiel général d'amélioration de l'accessibilité)	Public sector + some private	DINUM	*WCAG 2.1 AA + national rules*
Spain	Royal Decree 1112/2018	Public sector + education	Observatorio de Accesibilidad Web	*WCAG 2.1 AA*
Italy	Stanca Act (Law 4/2004), AgID Guidelines	Public sector	AgID	*EN 301 549 / WCAG 2.1 AA*
Netherlands	Temporary Decree on Digital Accessibility	Public sector	Ministry of the Interior	*WCAG 2.1 AA*
Sweden	Discrimination Act + WAD implementation	Public sector	DIGG (Digital Government Agency)	*WCAG 2.1 AA*
Finland	Act on the Provision of Digital Services	Public + selected private	Regional State Administrative Agency	*WCAG 2.1 AA*

Country	Main Law / Regulation	Applies To	Oversight Body	*Standard*
Poland	Digital Accessibility Act (2019)	Public sector	Ministry of Digital Affairs	*WCAG 2.1 AA*
Belgium	Law of 19 July 2018	Public sector	FPS BOSA	*WCAG 2.1 AA*
Portugal	Decree-Law No. 83/2018	Public sector	AMA (Modernization Agency)	*WCAG 2.1 AA*
Czech Republic	Act No. 99/2019 Coll.	Public sector	Ministry of the Interior	*WCAG 2.1 AA*
Hungary	Disability Rights Act + WAD transposition	Public sector	Ministry of Technology and Industry	*WCAG 2.1 AA*
Romania	Government Decision No. 585/2019	Public sector	Authority for the Digitalization of Romania	*WCAG 2.1 AA*
Estonia	Public Information Act (Accessibility Chapter)	Public sector	Ministry of Economic Affairs and Communications	*WCAG 2.1 AA*
Lithuania	Law on Accessibility of Websites and Apps	Public sector	Ministry of Social Security and Labour	*WCAG 2.1 AA*
Latvia	Cabinet Regulation No. 445 (2019)	Public sector	Ministry of Environmental Protection and Development	*WCAG 2.1 AA*
Slovakia	Act No. 95/2019	Public sector	National Agency for Electronic Services	*WCAG 2.1 AA*
Slovenia	Accessibility of Websites and Apps Act	Public sector	Ministry of Public Administration	*WCAG 2.1 AA*
Croatia	Web Accessibility Act	Public sector	Digital Society Office	*WCAG 2.1 AA*
Greece	Law 4727/2020 implementing the WAD	Public sector	Ministry of Digital Governance	*WCAG 2.1 AA*
Ireland	EU (Accessibility of Websites and Mobile Apps) Regulations	Public sector	National Disability Authority	*WCAG 2.1 AA*

Country	**Main Law / Regulation**	**Applies To**	**Oversight Body**	***Standard***
Bulgaria	Electronic Governance Act + WAD regulations	Public sector	State e-Government Agency	*WCAG 2.1 AA*

3. Summary Table – Comparison of EU Digital Accessibility Regulations

Scope	**Directive / Law**	***Applies To***	***Technical Basis***
EU-wide (public)	Web Accessibility Directive (2016/2102)	*Public sector websites & apps*	*EN 301 549 / WCAG 2.1 AA*
EU-wide (private)	European Accessibility Act (2019/882)	*Selected private sectors*	*EN 301 549 / WCAG 2.1 AA*
EU-wide (technical std)	EN 301 549	*ICT (web, software, hardware)*	*WCAG 2.1 AA + additions*
National laws	Vary by country	*Public sector (some private)*	*Usually WCAG 2.1 AA*

4. Conclusion

Across the EU, digital accessibility is governed by a **harmonized legal framework**, with country-specific implementations. Almost all countries use **WCAG 2.1 Level AA** as the technical standard, often embedded via **EN 301 549**.

- **Public sector obligations** are already in effect through the **Web Accessibility Directive**.
- **Private sector obligations** will increase significantly with the **European Accessibility Act**, fully enforceable by **28 June 2025**.
- **DACH countries** (Germany, Austria, Switzerland) are aligned in principle, with Germany actively preparing for EAA compliance and Switzerland following its own federal law (not part of the EU).

3.1.2 Digital Accessibility Outside the EU – Key Laws, Standards, and Oversight Structures

1. International Frameworks (Global Scope)

UN Convention on the Rights of Persons with Disabilities (CRPD)

- Ratified by **180+ countries** worldwide.

- **Article 9**: Mandates accessibility to ICT, including websites and digital services.
- Acts as a foundational influence for national legislation globally.

WCAG – Web Content Accessibility Guidelines

- Developed by the W3C's Web Accessibility Initiative (WAI).
- **WCAG 2.1 Level AA** is the most widely adopted benchmark standard for digital accessibility worldwide.
- Basis for legal compliance in the EU, US, UK, Canada, Australia, and others.

2. Overview of Key National Frameworks (Non-EU Countries)

Country	**Main Law / Regulation**	***Applies To***	**Oversight Body**	***Standard Used***
United States	Americans with Disabilities Act (ADA), Section 508 (Rehabilitation Act)	*Federal/public + litigation for private*	DOJ, U.S. Access Board	*WCAG 2.0/2.1 AA (Section 508 updated 2017)*
Canada	Accessible Canada Act (ACA), AODA (Ontario)	*Federal and provincial governments*	Accessibility Commissioner, provincial agencies	*WCAG 2.0 / WCAG 2.1 AA*
United Kingdom	Equality Act 2010, Public Sector Bodies Accessibility Regulations 2018	*Public sector + anti-discrimination for private*	Government Digital Service (GDS)	*WCAG 2.1 AA*
Australia	Disability Discrimination Act 1992 + DDA Advisory Notes	*Public + private (enforceable via court)*	Australian Human Rights Commission	*WCAG 2.0 AA*
New Zealand	Human Rights Act + Web Accessibility Standards (2020 update)	*Public sector*	Department of Internal Affairs	*WCAG 2.1 AA*
Norway	Equality and Anti-Discrimination Act	*Public + large private digital services*	Digitalisation Agency (Digdir)	*WCAG 2.1 AA (mandatory)*

Country	**Main Law / Regulation**	***Applies To***	**Oversight Body**	***Standard Used***
Iceland	Accessibility laws aligned with EEA/WAD	*Public sector*	Ministry of Social Affairs and Labour	*WCAG 2.1 AA*
Switzerland	Behindertengleichstellungsgesetz (BehiG)	*Federal administration (not EU member)*	Access for All Foundation	*WCAG 2.1 AA*
Japan	Act on the Elimination of Discrimination Against Persons with Disabilities	*Public sector (voluntary for private)*	Cabinet Office and ministries	*Japanese JIS X 8341-3:2016 (based on WCAG 2.0)*
South Korea	Act on the Prohibition of Discrimination Against Disabled Persons	*Public + private*	National Information Society Agency	*WCAG 2.1 AA (from 2023)*
Israel	Equal Rights for People with Disabilities Law (2013 update)	*Public + private*	Ministry of Justice	*WCAG 2.0 AA*
Brazil	Brazilian Inclusion Law (2015) + E-MAG standard	*Public sector*	Ministry of Human Rights	*WCAG 2.0 AA (E-MAG v3.0)*
India	Rights of Persons with Disabilities Act (2016)	*Public + large private digital platforms*	Ministry of Social Justice and Empowerment	*WCAG 2.0 AA*
China	Voluntary Web Accessibility Guidelines (national standard GB/T 37668-2019)	*Public + some private*	MIIT (Ministry of Industry and IT)	*WCAG-based, partial*
Russia	Law on Social Protection of Persons with Disabilities	*Public sector*	Ministry of Labor and Social Protection	*WCAG-based national standard*
Turkey	Law No. 5378 + E-Government Accessibility Standards	*Public sector*	Ministry of Family and Social Services	*WCAG 2.0 AA (goal: 2.1)*

Country	**Main Law / Regulation**	***Applies To***	**Oversight Body**	***Standard Used***
South Africa	Promotion of Equality and Prevention of Unfair Discrimination Act	*Public + private*	SA Human Rights Commission	*WCAG 2.0 AA (advisory)*
Mexico	General Law for Inclusion of People with Disabilities	*Public sector*	CONADIS (National Disability Council)	*WCAG-based local guidelines*

3. Summary Comparison – Non-EU Accessibility Frameworks

Region	**Legal Approach**	***Typical Technical Basis***	***Scope***
North America	Statutory + regulatory enforcement (ADA, ACA)	*WCAG 2.0 / 2.1 AA*	*Public and private sectors*
UK & Commonwealth	Human rights + digital laws	*WCAG 2.1 AA*	*Public + anti-discrimination*
Nordic Countries	Full alignment with EU/WAD	*WCAG 2.1 AA*	*Public + some private*
Asia-Pacific	National disability laws + mixed compliance	*WCAG 2.0 / JIS / GB/T*	*Mostly public sector*
Latin America	Constitutional + inclusion frameworks	*WCAG 2.0 AA / regional models*	*Mainly public sector*
Middle East	Gradual enforcement via inclusion laws	*WCAG 2.0 AA*	*Public and some private*
Africa	Emerging frameworks, mostly advisory	*WCAG 2.0 AA*	*Public sector (early stage)*

4. Conclusion

Globally, **WCAG 2.0 or 2.1 Level AA** serves as the de facto standard for digital accessibility. Countries differ significantly in:

- **Legal force** (e.g., advisory vs. enforceable through litigation)
- **Sector scope** (public only vs. including private entities)
- **Implementation maturity**

Many non-EU countries have **sector-specific or provincial laws**, and some (like the US, Canada, and UK) actively enforce compliance through audits or lawsuits. Other countries, particularly in Asia and Latin America, are rapidly developing frameworks, often inspired by the **UN CRPD** and **EU models**.

3.1.3 Introduction to the Web Content Accessibility Guidelines (WCAG)

Background and Purpose

The Web Content Accessibility Guidelines (WCAG) were developed by the **World Wide Web Consortium (W3C)** through its **Web Accessibility Initiative (WAI)**. Their purpose is to ensure that digital content is accessible to people with disabilities, including those with visual, auditory, physical, speech, cognitive, language, learning, and neurological disabilities.

WCAG is a **technical standard**, not a law—but it serves as the foundation for many national and international accessibility regulations (such as Section 508 in the U.S., the Equality Act in the UK, or the EN 301 549 standard in the EU). It is internationally recognized as the core benchmark for web and digital accessibility.

Historical Development of WCAG

- **1999 – WCAG 1.0**
  WCAG 1.0 was the first official version, focusing mainly on HTML-based content. It introduced 14 guidelines with checkpoints at three priority levels. Although groundbreaking, it reflected the web technologies of its time and became outdated with the rise of dynamic and interactive content.
- **2008 – WCAG 2.0**
  WCAG 2.0 marked a major shift to a more **technology-neutral** and **testable** framework. It introduced four core **principles of accessibility**:
  - **Perceivable**
  - **Operable**
  - **Understandable**
  - **Robust**

  These principles are known as the POUR model, and they continue to structure the WCAG framework. WCAG 2.0 contains 12 guidelines and 61 success criteria, organized by three conformance levels: A, AA, and AAA.

- **2018 – WCAG 2.1**
  This version extended WCAG 2.0 by adding 17 new success criteria, especially for:

  - **Mobile accessibility**
  - **Low vision**
  - **Cognitive and learning disabilities**

  WCAG 2.1 became the legal standard referenced in many regulations, such as the **EU Web Accessibility Directive** and the **UK's Accessibility Regulations**.

- **2023 – WCAG 2.2**
  WCAG 2.2 builds upon 2.1 and introduces 9 new success criteria, focusing on:
  - Better support for **users with cognitive disabilities**
  - Improved **keyboard navigation**
  - Simplification of **authentication tasks**

  WCAG 2.2 is **backward compatible**, meaning that content conforming to WCAG 2.2 also meets the requirements of WCAG 2.1 and 2.0.

Ongoing Development: WCAG 3.0

WCAG 3.0 (currently a working draft) is a complete rethinking of how accessibility is measured. It aims to:

- Address **new technologies**, including voice assistants, AR/VR, and AI interfaces.
- Be **more inclusive** of different user needs.
- Provide a **more flexible scoring model**, instead of strict pass/fail criteria.

WCAG 3.0 will eventually **replace** the current versions, but this transition will take several years. Until then, WCAG 2.x remains the active standard.

Summary

WCAG has evolved to keep pace with technological advances and a deeper understanding of users' diverse accessibility needs. Its development reflects a global effort to make the web and digital systems inclusive for everyone. Understanding the history and structure of WCAG is essential for anyone working in digital design, development, content, or compliance.

WCAG Conformance Levels and Legal Relevance (Overview)

WCAG Conformance Levels

The Web Content Accessibility Guidelines (WCAG) define **three levels of conformance** that reflect the degree to which digital content meets accessibility requirements:

- **Level A (Minimum Accessibility)**
  This is the **most basic level** of accessibility. Content that conforms to Level A removes some of the most significant barriers for users with disabilities. However, it does **not ensure broad accessibility** and is **not sufficient** to meet most legal or organizational requirements.
- **Level AA (Recommended Standard)**
  Level AA addresses **the most common and significant barriers** to accessibility for a wide range of users.
  It is the **recommended level** for public-facing websites and applications and is the **benchmark for legal compliance** in most English-speaking countries.
  Examples of Level AA requirements include:
    - Minimum contrast ratios for text
    - Meaningful use of headings and labels
    - Keyboard navigability
    - Error identification in forms
- **Level AAA (Maximum Accessibility)**
  Level AAA includes additional success criteria that offer the **highest standard of accessibility**, such as:
    - Sign language interpretation for prerecorded video
    - Extended time limits
    - Simplified language

      However, **Level AAA is not required for legal compliance** and is often considered aspirational. It is difficult to meet across an entire website, especially for complex applications.

Legal Relevance of WCAG in the English-Speaking World

United States

- **Section 508 of the Rehabilitation Act**: Requires federal agencies to conform to **WCAG 2.0 Level AA**.
- **Americans with Disabilities Act (ADA)**: While WCAG is not explicitly written into the ADA, **courts and the Department of Justice reference WCAG 2.1 Level AA** as the standard for ensuring non-discriminatory digital access.
- **Recommendation**: Conform to **WCAG 2.1 Level AA** for public and private sector compliance.

United Kingdom

- **Public Sector Bodies Accessibility Regulations 2018**: Legally requires public sector websites and mobile apps to meet **WCAG 2.1 Level AA**.
- **Equality Act 2010**: Applies broadly to both public and private sectors and is interpreted to include **WCAG-based standards** for digital accessibility.

Canada

- **Accessible Canada Act (ACA)** and **provincial laws** (e.g., AODA in Ontario): Require digital content to comply with **WCAG 2.0 or 2.1 Level AA** depending on the jurisdiction.

Australia

- **Disability Discrimination Act (DDA) 1992**: Does not mandate WCAG by name, but **WCAG 2.1 Level AA is the accepted standard**, supported by government and court decisions.

New Zealand

- The **New Zealand Government Web Accessibility Standard** aligns with **WCAG 2.1 Level AA** and applies to public sector digital services.

Summary of Legal Standards:

Country	Legal Reference	Required WCAG Level
United States	Section 508, ADA (interpreted)	WCAG 2.0 or 2.1 Level AA
United Kingdom	Accessibility Regulations, Equality Act	WCAG 2.1 Level AA
Canada	ACA, AODA	WCAG 2.0 or 2.1 Level AA
Australia	DDA (case law + guidance)	WCAG 2.1 Level AA (de facto)
New Zealand	Government Digital Service Standards	WCAG 2.1 Level AA

Conclusion

For most public and private sector organizations in the English-speaking world, **WCAG 2.1 Level AA** is the practical **minimum conformance level** to aim for. While Level AAA offers enhanced accessibility, it is not legally required. Meeting Level AA ensures that the majority of users with disabilities can access digital content effectively and helps organizations comply with current legal standards and best practices.

3.1.4 The 13 WCAG 2.1 / 2.2 Guidelines and Their Principles

The **Web Content Accessibility Guidelines (WCAG)** are built around **four foundational principles** that define accessibility.

Content must be:

1. Perceivable – Information must be available to the senses

Guideline	Description
1.1 Text Alternatives	Provide text alternatives for non-text content (e.g., images, icons).
1.2 Time-based Media	Provide alternatives for audio and video (e.g., captions, transcripts).
1.3 Adaptable	Ensure content is adaptable and not dependent on a specific presentation.
1.4 Distinguishable	Make it easier for users to see and hear content (e.g., color contrast, text spacing).

Examples:

1. An image has descriptive alt text so screen reader users understand its purpose.
2. A video includes synchronized captions for users who are deaf or hard of hearing.
3. Text has sufficient color contrast from the background (e.g., dark text on a light background).
4. Headings are marked up semantically using <h1>, <h2>, etc., allowing screen readers to navigate.
5. Charts and graphs are accompanied by data tables or text summaries.

2. Operable – User interface components must be usable via multiple input methods

Guideline	Description
2.1 Keyboard Accessible	All functionality must be operable via keyboard.
2.2 Enough Time	Users must have enough time to read and interact with content.
2.3 Seizures and Physical Reactions	Avoid content that could cause seizures or physical discomfort (e.g., flashing).

Guideline	Description
2.4 Navigable	Help users navigate, find content, and know where they are (e.g., headings, focus indicators).
2.5 Input Modalities *(added in WCAG 2.1)*	Make it easier to operate functionality using various inputs (e.g., touch, speech, sensors).

Examples:

1. All website functionality is available using only the keyboard (no mouse required).
2. A visible focus indicator shows which element is currently selected when tabbing.
3. A form timeout provides a warning and allows users to extend the time limit.
4. There are no flashing elements that could trigger seizures (nothing flashes more than 3 times per second).
5. A mobile app has large, well-spaced buttons that are easy to tap.

3. Understandable – Content and interface must be clear and predictable

Guideline	Description
3.1 Readable	Make text content readable and understandable.
3.2 Predictable	Make web pages appear and behave in predictable ways.
3.3 Input Assistance	Help users avoid and correct mistakes (e.g., error messages, labels).

Examples:

1. A form includes helpful instructions and clearly labeled input fields.
2. Error messages explain what went wrong and how to fix it (e.g., "Password must include a number").
3. The site uses consistent navigation on all pages, menus appear in the same place.
4. The language of each page is correctly declared in the HTML (e.g., <html lang="en">).
5. Technical terms are explained or avoided where possible for easier comprehension.

4. Robust – Content must be reliably interpreted by assistive technologies

Guideline	Description
4.1 Compatible	Maximize compatibility with current and future user agents, including assistive tech.

Notes on WCAG 2.2

WCAG 2.2 (published 2023) **adds 9 new success criteria** but **does not add new guidelines**—the structure of **13 guidelines under 4 principles remains unchanged**.

Examples:

1. All interactive controls use standard HTML elements (e.g., <button>, <input>), which are recognized by screen readers.
2. ARIA (Accessible Rich Internet Applications) roles and properties are used appropriately (e.g., aria-label).
3. The website is tested with multiple screen readers and browsers to ensure consistent accessibility.
4. Form fields include correct attributes like label, id, and for so assistive tech can interpret them.
5. The site avoids using deprecated HTML or JavaScript-only interaction without fallback methods.

3.2 Selected Individual Laws and Regulations

Terms: Section 508, Americans with Disabilities Act (ADA), Trusted Tester Program, Public Sector Accessibility Regulations (UK), Equality Act 2010 (UK), ACT Rules Format

This section provides an overview of the legal frameworks and key accessibility standards related to the rights and inclusion of people with disabilities in the English-speaking world.

3.2.1 United States – Section 508 & Americans with Disabilities Act (ADA)

Section 508 of the Rehabilitation Act (Revised 2017)
Section 508 is a federal U.S. law that requires all electronic and information technology (EIT) developed, procured, maintained, or used by federal agencies to be accessible to people with disabilities.
Key aspects:

- Applies to federal agencies and contractors.
- Aligned with WCAG 2.0 Level AA as the technical standard.
- Covers websites, documents, software, and hardware.

- Evaluated using the **Trusted Tester Program**, which ensures a consistent, human-centered testing approach.

Americans with Disabilities Act (ADA), Titles II and III

The ADA is a comprehensive civil rights law prohibiting discrimination against individuals with disabilities in all areas of public life.

Key aspects:

- **Title II** applies to state and local governments.
- **Title III** applies to private businesses and public accommodations.
- Although the ADA does not specify WCAG explicitly, **courts and the U.S. Department of Justice interpret it to include digital accessibility**.
- Increasing litigation has established **WCAG 2.1 Level AA** as the expected compliance benchmark for websites and mobile apps.

Relevance for digital accessibility education:

Understanding Section 508 and ADA is essential for professionals working with or for U.S. institutions. These laws form the legal basis for accessible digital design and development, and set expectations for both manual and automated accessibility evaluation.

3.2.2 United Kingdom – Public Sector Accessibility Regulations & Equality Act

Public Sector Bodies (Websites and Mobile Applications) Accessibility Regulations 2018

This regulation implements the EU Web Accessibility Directive into UK law and remains in effect post-Brexit. It mandates that public sector websites and mobile apps must be accessible to all users, including people with disabilities.

Key points:

- Requires conformance with **WCAG 2.1 Level AA**.
- Applies to UK public sector bodies, including central and local government, schools, and hospitals.
- Organizations must:
  – Publish an **accessibility statement**.
  – Regularly **test their digital services for accessibility**.
  – Fix identified accessibility issues within a reasonable timeframe.
- Oversight is provided by the **UK Government Digital Service (GDS)** and the **Equality and Human Rights Commission (EHRC)**.

Equality Act 2010

The Equality Act is the UK's overarching anti-discrimination law, which protects individuals from unfair treatment in various areas of life, including access to goods and services.

Key points:

- Applies to both **public and private sectors**.
- Requires **"reasonable adjustments"** to avoid disadvantaging people with disabilities.
- Increasingly interpreted to include **websites, mobile apps, and digital content**.
- Non-compliance can lead to **legal challenges and enforcement by the EHRC**.

Relevance for digital accessibility:

Understanding these laws equips learners to design and manage accessible digital content within the UK legal framework. These regulations set a clear standard—**WCAG 2.1 AA**—and create enforceable obligations that affect both public institutions and private providers. Familiarity with UK accessibility requirements is especially important for working with public sector contracts or ensuring compliance across digital services.

3.3 Web Content Accessibility Guidelines – Evaluation Methodology WCAG-EM (EU) and Trusted Tester Program (USA)

WCAG-EM (Web Content Accessibility Guidelines – Evaluation Methodology) is a framework developed by the W3C, widely used in the European Union to guide the structured evaluation of websites for WCAG 2.1 compliance—particularly in the context of public sector monitoring under the EU Web Accessibility Directive. It focuses on selecting representative pages, applying consistent testing methods, and documenting results transparently.

The Trusted Tester Program, developed by the U.S. Department of Homeland Security, is a formalized manual testing process tailored for U.S. federal agencies. It includes a detailed step-by-step methodology and offers certification for testers, ensuring consistent, code-based evaluations aligned with WCAG and Section 508 requirements.

3.3.1 Trusted Tester Program (USA)

Provider: U.S. Department of Homeland Security (DHS)

Type: Standardized manual accessibility testing method

Purpose:

To ensure consistent, accurate, and repeatable WCAG-based accessibility testing—especially for U.S. federal government websites and applications.

Key Features:

- Based on **WCAG 2.1 Level AA**
- Used to assess conformance of web content, documents, and software
- Offers a defined, step-by-step **manual testing protocol**
- Includes use of tools like **ANDI (Accessible Name & Description Inspector)**
- Ensures human-centered quality control beyond what automated tests can provide

Training and Certification:

- Free online course provided by DHS
- Certification awarded upon passing final exam
- Widely used across U.S. federal agencies for Section 508 compliance

3.3.2 WCAG-EM (Web Content Accessibility Guidelines – Evaluation Methodology)

WCAG-EM is a formal evaluation methodology developed by the **World Wide Web Consortium (W3C)** to support the **structured and consistent assessment of websites** for conformance with the **Web Content Accessibility Guidelines (WCAG) 2.1 or 2.0**. Rather than simply listing test results, WCAG-EM defines a clear **five-step process** for how to plan, conduct, and report an accessibility evaluation—especially when testing entire websites or representative samples.

Structure and Process

WCAG-EM is organized into five core steps:

1. **Define the evaluation scope** – Identify the target website, its purpose, target audiences, technologies used, and applicable WCAG version and level (e.g., 2.1 AA).
2. **Explore the website** – Understand its structure, content types, navigation mechanisms, and templates.
3. **Select a representative sample** – Choose web pages that reflect the variety of content, functionality, and design patterns across the site.
4. **Audit the selected sample** – Evaluate each selected page against WCAG success criteria using a combination of manual and (optionally) automated techniques.
5. **Report the results** – Summarize findings, list successes and issues per criterion, and provide an overall conformance statement.

Official Documentation

WCAG-EM is published by the W3C and freely available online at:

https://www.w3.org/WAI/test-evaluate/conformance/wcag-em/

It includes:

- A detailed methodology guide
- A separate reporting template (WCAG-EM Report Tool)
- Examples of how to apply the methodology in practice

Use in Practice

WCAG-EM is particularly relevant for:

- **Public sector bodies in the European Union**, as part of accessibility monitoring under **Directive (EU) 2016/2102**
- **Accessibility consultants**, conducting full-site audits or client reports
- **Development teams** performing internal accessibility self-assessments
- **Research and benchmarking**, where transparency and repeatability are essential

Although WCAG-EM is **technology-neutral** and does not prescribe specific tools, it aligns closely with the **WCAG success criteria** and supports integration with manual, semi-automated, or hybrid evaluation processes.

4 Conducting an Accessibility Test

⏱ Estimated Instruction Time: Approximately 480 Minutes

🎯 Learning Objectives of the Chapter

LO 4.1.1 – Explain the WCAG success criterion for alternative text and describe how it can be fulfilled. – K2

LO 4.1.2 – Identify appropriate text alternatives for controls and informative images. – K3

LO 4.1.3 – Recall the requirements for text alternatives in CAPTCHAs. – K1

LO 4.2.1 – Recall the benefits of captions for pre-recorded video content. – K1

LO 4.2.2 – Recall the requirements for alternatives to audio-only content. – K1

LO 4.2.3 – Recall the requirements for alternatives to video-only content. – K1

LO 4.2.4 – Recall the requirements for alternatives to live audio and video. – K1

LO 4.2.5 – Recall the requirements for audio descriptions or full text alternatives for synchronized media. – K1

LO 4.3.1 – Identify a meaningful and logical reading order of page content. – K3

LO 4.3.2 – Determine appropriate use of various HTML structural elements. – K3

LO 4.3.3 – Explain the WCAG success criterion for sensory characteristics and how it is tested. – K2

LO 4.3.4 – Explain the WCAG success criterion for identifying the purpose of input fields and how to expose this programmatically. – K2

LO 4.4.1 – Determine and evaluate contrast requirements for text and graphical content. – K3

LO 4.4.2 – Explain why information must not be conveyed through color alone and formulate appropriate recommendations for remediation. – K3

LO 4.5.1 – Explain the meaning of device independence in web accessibility. – K2

LO 4.5.2 – Recall the WCAG success criterion for keyboard accessibility. – K1

LO 4.5.3 – Recall the WCAG success criterion for avoiding keyboard traps. – K1

LO 4.5.4 – Recall the WCAG success criterion for keyboard shortcuts. – K1

LO 4.6.1 – Explain under which conditions time limits comply with accessibility requirements. – K2

LO 4.7.1 – Describe in which situations flashing content on websites may create an accessibility barrier. – K2

LO 4.8.1 – Evaluate what constitutes a meaningful and logical focus order for keyboard navigation. – K3

LO 4.8.2 – Explain the requirements for visible keyboard focus indicators. – K2

LO 4.8.3 – Determine when link texts are meaningful and explain the importance of this requirement. – K3

LO 4.8.4 – Explain the purpose and implementation of skip links for keyboard navigation. – K2

LO 4.9.1 – Explain the importance of ensuring the visible label is part of the accessible name. – K2

LO 4.9.2 – Recall the WCAG success criterion for pointer gestures. – K1

LO 4.9.3 – Recall the WCAG success criterion for pointer cancellation. – K1

LO 4.9.4 – Recall the WCAG success criterion for motion-activated inputs. – K1

LO 4.9.5 – Recall and apply the WCAG success criterion for target size. – K3

LO 4.10.1 – Explain the impact of missing or incorrect language declarations on web pages. – K2

LO 4.10.2 – Recall the WCAG success criterion for the language of text passages. – K1

LO 4.11.1 – List examples of unexpected context changes triggered by user input. – K1

LO 4.11.2 – Explain the principle and requirements for consistent navigation. – K2

LO 4.11.3 – Explain the principle and requirements for consistent identification of elements. – K2

LO 4.11.4 – Recall the WCAG success criterion for context changes triggered by focus changes. – K1

LO 4.12.1 – Explain how the requirements for labeling form elements can be fulfilled. – K2

LO 4.12.2 – Assess whether error identification and user guidance are implemented in an accessible and understandable manner. – K3

LO 4.12.3 – Recall and apply the WCAG success criterion for error prevention. – K3

LO 4.12.4 – Recall the WCAG success criterion for context-sensitive help. – K1

LO 4.13.1 – Formulate appropriate recommendations for implementing programmatically available status messages. – K3

LO 4.13.2 – Explain how semantics can be provided using HTML structural elements or WAI-ARIA and describe the role of correct syntax. – K2

About the W3C Recommendations: Web Content Accessibility Guidelines (WCAG)

The Web Content Accessibility Guidelines (WCAG) are published as official W3C Recommendations, meaning they have undergone thorough review and represent a stable, authoritative standard endorsed by the World Wide Web Consortium (W3C). As Recommendations, they are intended to be implemented widely and serve as a reliable basis for accessibility compliance.

WCAG Recommendations provide a set of testable success criteria organized under four core principles: content must be Perceivable, Operable, Understandable, and Robust (POUR). These criteria help developers, designers, and testers evaluate whether digital content is accessible to people with various disabilities.

4.1 Text Alternatives

Key Terms: alternative text, controls, CAPTCHAs, non-text content, alt attribute, WCAG 1.1.1

Definition:
Text alternatives are descriptive texts that convey the meaning and function of non-text content—such as images, icons, buttons, or multimedia elements—in clear and understandable language. They enable people with visual or cognitive impairments to perceive and interact with content using assistive technologies like screen readers.

4.1.1 Evaluating Text Alternatives

Test Criterion and Procedure
WCAG 2.2 – Success Criterion 1.1.1: Non-text Content
“All non-text content that is presented to the user has a text alternative that serves the equivalent purpose, except in specific situations (e.g., decorative content, sensory tests, or CAPTCHAs).”

Corresponding Evaluation Procedure:
Are text alternatives for graphics and objects present and meaningful?

Description
Success Criterion 1.1.1 requires that all relevant non-text content, such as images, icons, controls, or embedded objects, must include a meaningful text alternative. These alternatives allow assistive

technologies (like screen readers or Braille displays) to convey the information or functionality of the original content.

The Evaluation Includes the Following Steps:

1. **Presence**
   – Each relevant non-text element must have a text alternative, such as an alt attribute for <img> elements, text in ARIA labels, or accessible descriptions for SVG elements.
2. **Content Accuracy**
   – The text alternative must clearly, accurately, and purposefully describe the information or function of the element. It should not be vague or misleading.
3. **Purpose-specific Differentiation**
   – The type of description depends on the role of the image:
   • *Informative*: conveys content (e.g., charts, icons)
   • *Functional*: serves a purpose (e.g., buttons)
   • *Decorative*: adds visual appeal only (should use alt="")
   • *Text-based images*: reproduce visible text in the alternative text
4. **Technical Implementation**
   – The HTML or code structure must be correct and accessible. Automated tools can detect missing or empty alt attributes but cannot assess the appropriateness of the content.
5. **Assistive Technology Testing**
   – Use screen readers to test whether the alternative text is actually read aloud and whether it makes sense in the given context. This helps confirm usability for real users.

Examples

- **Informative graphic:**
  A chart labeled “Revenue increases significantly in 2024” → alt="Revenue increase in 2024"
- **Functional graphic (e.g., icon button):**
  A magnifying glass icon used for search → alt="Start search"
- **Decorative image:**
  A visual divider without meaning → alt=""
- **Text image (e.g., logo):**
  A logo showing “Example LLC” → alt="Example LLC"
- **Complex graphic:**
  A detailed flowchart or infographic → alt="Detailed description below the graphic"

Summary

Success Criterion 1.1.1 is essential for making digital content perceivable. To ensure that text alternatives are accessible, they must be:

- **present**,
- **accurately describe the content**,
- **tailored to the content's purpose**, and
- **implemented with correct technical markup**.

Only then can users of screen readers or other assistive technologies gain equal access to the information and functionality of non-text content.

4.1.2 Text Alternatives for Controls and Informative Graphics

Test Criterion and Procedure

WCAG 2.2 – Success Criterion 1.1.1: Non-text Content

All non-text content that is presented to the user must have a text alternative that serves the equivalent purpose, except in specific cases (e.g., decorative content or information that cannot be conveyed in text).

Corresponding Evaluation Procedure:

Are the text alternatives for controls and graphical links appropriate and functional?

This involves checking whether the alternative texts clearly and accurately describe the purpose and function of graphical user interface elements.

Description

Appropriate text alternatives must reflect the **purpose and function** of the graphical element. A distinction is made between:

- **Controls** such as icon buttons, symbols, or links that perform an action or lead to another page
- **Informative graphics** that convey content or messages visually

For **controls**, the alternative text should describe the **action or destination**, not the visual appearance (e.g., say "Go to next page," not "Right arrow"). For **informative graphics**, the alternative text must summarize the **main message** in a clear and concise way.

For **complex visual content** (e.g., large charts or infographics), a simple alt attribute is not sufficient. A detailed description should be provided in surrounding text or via a separate link.

Examples

1. Appropriate Text Alternatives for Graphical Controls – by Element Type

Function	Element Type	Recommended Accessible Implementation
Search icon (magnifying glass)	<img> within a link	<img src="search.png" alt="Start search">
Search icon (as button)	<button> with CSS icon or SVG	<button aria-label="Start search"></button>
Arrow icon to next page	<img> in a link	<img src="arrow.png" alt="Next page">
Arrow in button (SVG or CSS)	<button>	<button aria-label="Next page"></button>
Download icon (PDF)	<img> as link	<img src="download.png" alt="Download PDF">
Download button with icon	<button> with SVG	<button aria-label="Download PDF"></button>
Print icon in button	<button> with icon	<button aria-label="Print page"></button>

2. Informative Graphics

Function / Image Type	Appropriate Alternative Text	Inappropriate Alternative Text
Revenue chart	alt="Bar chart showing revenue increase from €2M in 2020 to €6M in 2023."	alt="Chart" or alt="Graphic"
Historical figure (portrait)	alt="Portrait of Marie Curie, Nobel laureate in Physics and Chemistry."	alt="Woman" or alt="Marie Curie"
Complex infographic	alt="Detailed description follows in the text below the graphic."	alt="Infographic" or no reference

Note: Controls without functionality (purely decorative) should use an empty alt="" or be included via CSS.

Summary

Text alternatives for graphical content must:

- **Clearly describe the function** (for controls) or **convey the main content** (for informative graphics)
- **Focus on meaning**, not appearance (e.g., describe the action, not the icon shape)
- **Reference longer descriptions** for complex content
- **Avoid redundancy** when the content is already described in surrounding text

This ensures that digital interfaces are fully understandable and usable for individuals who rely on assistive technologies.

4.1.3 Additional Elements

Test Criterion and Procedure
WCAG 2.2 – Success Criterion 1.1.1: Non-text Content
CAPTCHAs are considered non-text content and must include text alternatives that convey their purpose. Where non-accessible CAPTCHAs are used, an **accessible, equivalent alternative** must be provided to ensure usability for all user groups.

Corresponding Evaluation Procedures:

- Does the CAPTCHA contain a meaningful text alternative that explains its purpose and refers to any available accessible option?
- Is an **accessible alternative solution** provided that works for users with different types of disabilities?

Description
CAPTCHAs are used to differentiate human users from automated bots. However, image-based or audio-based CAPTCHAs often present significant accessibility barriers—especially for users who are blind, have low vision, or are deaf or hard of hearing.

To meet accessibility requirements, CAPTCHAs must:

- Provide a **clear text alternative** (e.g., using the alt attribute for images) that explains the purpose of the CAPTCHA.
- Include a **reference to an accessible alternative**, if one exists.
- Offer a **fully accessible and equivalent alternative**—such as a logic-based challenge, simple math problem, or two-factor authentication—that allows equal access.

Note: The visual clarity of characters in the CAPTCHA image is not part of accessibility evaluation and is not a WCAG requirement.

Examples

1. **Image-based CAPTCHA with alt text:**

<img src="captcha.jpg" alt="Enter the characters shown in the image. If you cannot read them, please use the accessible alternative below.">

2. **Accessible alternatives to CAPTCHA:**

- **Text link to alternative method:**
  "Use alternative verification method"
- **Logic-based alternative examples:**
  - A simple math question: "What is 3 plus 5?"
  - A checkbox with verification logic: "I am not a robot"

Summary
To implement CAPTCHAs in an accessible way, the following criteria must be met:

- **Image-based CAPTCHAs** must have a meaningful text alternative that clearly explains their purpose and refers to available alternatives.
- At least one **accessible alternative** must be provided to accommodate users with different impairments.
- The **accessible method** must be easy to locate, easy to use, and functionally equivalent to the original CAPTCHA.

This ensures that protected content remains accessible to all users, including those who rely on assistive technologies.

4.2 Time-Based Media

Key Terms: time-based media, audio-only, video-only, synchronized media, captions, transcripts, audio description

Definition
Time-based media, as defined by the WCAG, refers to any content whose information is dependent on a temporal sequence. This includes audio, video, or combined formats that may be pre-recorded or live-

streamed. Accessibility for such content is ensured through alternatives such as transcripts, captions, audio descriptions, or full text alternatives.

Evaluation of Time-Based Media According to WCAG 2.1

Time-based media includes content that unfolds over time, such as audio recordings, videos, or narrated presentations. To ensure accessibility, WCAG 2.1 defines specific requirements based on the type and format of the media. The following evaluation steps align with corresponding **WCAG Success Criteria**:

1. Audio-Only Content (Pre-Recorded)
WCAG Success Criterion 1.2.1: Audio-only and Video-only (Prerecorded)
Requirement: Provide a **full transcript** that captures all spoken content in written form.
Applies to: Podcasts, voice recordings without visual elements.

2. Video-Only Content (Pre-Recorded)
WCAG Success Criterion 1.2.1: Audio-only and Video-only (Prerecorded)
Requirement: Provide a **text alternative** describing the visual content.
Applies to: Videos with no audio track, such as silent recordings or surveillance footage.

3. Audio-Video Content (Pre-Recorded)
WCAG Success Criterion 1.2.2: Captions (Prerecorded)
Requirement: Provide **synchronized captions** that include dialogue, sound effects, and speaker identification.
Applies to: Tutorials, recorded lectures, instructional videos.

4. Audio-Video Content with Visual Information Beyond the Spoken Track (Pre-Recorded)
WCAG Success Criterion 1.2.3: Audio Description or Media Alternative (Prerecorded)
Requirement: Provide either an **audio description** or an **alternative version** that presents visual content conveyed during the video.
Applies to: Films, demonstrations, or educational content with significant visual elements.

5. Audio-Video Content (Live)
WCAG Success Criterion 1.2.4: Captions (Live)
Requirement: Provide **live captions** to ensure accessibility for deaf or hard-of-hearing users.
Applies to: Live conferences, streamed lectures, real-time webinars.

6. Complex Pre-Recorded Audio-Video Content Where Audio Description Is Not Feasible
WCAG Success Criterion 1.2.5: Audio Description (Prerecorded)
Requirement: Provide a **full text alternative** that includes both spoken and visual information.
Applies to: Technical or complex content where integrating audio description is impractical.

Summary Table of Requirements

Media Type	WCAG Success Criterion	Required Measure
Audio-only (pre-recorded)	1.2.1	Provide a full transcript
Video-only (pre-recorded)	1.2.1	Provide a text alternative for visual content
Audio-video (pre-recorded)	1.2.2	Provide synchronized captions
Audio-video (pre-recorded)	1.2.3	Provide audio description or alternative version
Audio-video (live)	1.2.4	Provide live captions
Audio-video (pre-recorded)	1.2.5	Provide a full text alternative if audio description is not possible

Important Note:
These accessibility requirements apply only to media that is essential for understanding or using the content. Purely decorative or irrelevant time-based media does not require a full accessibility solution.

Conclusion
Time-based media must be made accessible through appropriate alternatives depending on the format and context of use. These measures ensure that all users—including people who are blind, deaf, or have cognitive impairments—can fully engage with audio and video content.

4.2.1 Example: Detailed Evaluation of Captions

WCAG 2.2 – Success Criterion 1.2.2: Captions (Prerecorded)
This criterion requires that prerecorded videos with audio content provide **synchronized captions** that convey all spoken dialogue and relevant non-speech audio information.

Corresponding Evaluation Procedure:

- Are captions present for all time-based media with audio (e.g., tutorials, presentations)?
- Do the captions:

  - Include all spoken content and relevant sounds?
  - Appear in sync with the video?
  - Remain legible and appropriately placed on the screen?

Description

Captions are a key accessibility feature for making videos with audio content usable by individuals who are deaf or hard of hearing. They provide a written representation of:

- Spoken language
- Relevant background sounds (e.g., music, applause, ambient noise)
- Speaker identification when multiple people are talking

Captions also benefit a wider audience, including:

- Users in noisy or silent environments (e.g., public transportation, offices)
- People with limited language proficiency or learning difficulties
- Users who want to search or scan video content more efficiently

To meet accessibility requirements, captions must be **complete, synchronized with the video**, and **visually accessible** (sufficient contrast, appropriate font size, and readable placement).

Examples

Good captions include:

- **Spoken language:**
  “Welcome to today’s presentation.”
- **Non-speech audio:**
  “[soft music]”, “[door opens]”, “[applause]”
- **Speaker identification:**
  Presenter: “We will now move on to the next section.”

Inadequate captions might:

- Only cover part of the dialogue
- Omit important sounds
- Be out of sync with the video
- Be difficult to read due to poor contrast or small text size

Summary

Captions for prerecorded videos with audio are mandatory when the content is essential for understanding. To be considered accessible, captions must:

- Convey **all spoken dialogue and relevant audio cues**
- Be **synchronized** with the video timeline
- Be **clearly readable** (e.g., high contrast, legible font, adequate size and placement)

These requirements ensure that audiovisual content is fully accessible to users who are deaf, hard of hearing, or viewing in non-ideal listening conditions.

4.3 Adaptable Content

Key Terms: adaptable, meaningful sequence, HTML structural elements, sensory characteristics, input purpose, responsive design

Definition

In the context of WCAG, “adaptable” means that digital content must be designed in a way that allows it to be presented in different formats or contexts—without losing information or functionality. This ensures that users with diverse needs, preferences, or assistive technologies can access and interact with content effectively.

Adaptability supports various user experiences, including:

- Access across devices and screen sizes (responsive design)
- Content readability in high-contrast or zoom modes
- Understanding without relying solely on visual or sensory cues (e.g., color, shape, or position)

Though “adaptable” is not a standalone WCAG principle, it is addressed through multiple **success criteria** under the **Perceivable** principle. Relevant WCAG 2.2 criteria include:

- **1.3.1 Info and Relationships** – ensure structure is machine-readable
- **1.3.2 Meaningful Sequence** – preserve logical reading order
- **1.3.3 Sensory Characteristics** – avoid reliance on sensory-only cues
- **1.3.4 Orientation** – allow both portrait and landscape modes
- **1.3.5 Identify Input Purpose** – make input purposes programmatically detectable
- **1.4.4 Resize Text** – support up to 200% zoom without loss of content or functionality

Evaluation: Adaptable Content

Adaptability ensures that content can be perceived and used regardless of presentation style—whether via screen reader, Braille display, or zoomed interface. The following WCAG 2.2 success criteria are relevant to evaluating adaptability:

WCAG Success Criterion	What is Evaluated
1.3.1 Info and Relationships	Are headings, lists, and tables correctly marked up with HTML?
1.3.2 Meaningful Sequence	Is the reading order logical and consistent with the visual layout?
1.3.3 Sensory Characteristics	Can content be understood without relying only on color, shape, or location?
1.3.4 Orientation	Is content usable in both portrait and landscape orientations?
1.3.5 Identify Input Purpose	Are form field purposes exposed programmatically (e.g., via autocomplete)?

Description

WCAG Guideline 1.3 focuses on making content adaptable by ensuring that the structure and semantics are accessible to assistive technologies. This means:

- Content must use appropriate **semantic HTML tags** (e.g., <h1>–<h6>, <ul>, <table>)
- The **logical reading order** must match the visual structure
- Instructions must not depend **only on sensory characteristics** like color or position
- **Form input fields** must include machine-readable hints (e.g., autocomplete="email")

Examples

Structured HTML Content:

```
<h2>Contact Form</h2>
<label for="email">Email Address:</label>
<input type="email" id="email" autocomplete="email">
<ul>
  <li>First item</li>
  <li>Second item</li>
</ul>
```

Avoiding Sensory-Only Instructions:

- Incorrect: “Click the green button.”
- Correct: “Click the button labeled ‘Confirm’.”

Accessible Input Fields:

```
<input type="text" name="firstname" autocomplete="given-name">
<input type="text" name="postalcode" autocomplete="postal-code">
```

Summary

Adaptable content ensures that digital interfaces remain usable and understandable across devices, formats, and user needs. Accessibility checks focus on:

- Correct semantic markup (e.g., for headings and lists)
- Logical and programmatically readable sequence of content
- Alternative wording for sensory-based instructions
- Form field purposes that can be recognized by assistive technologies

These practices are critical to ensuring that users with screen readers, screen magnifiers, or alternative input methods can access content without barriers.

4.3.1 Example: Detailed Evaluation of "Meaningful Sequence"

Test Criteria and Procedures

WCAG 2.2 – Success Criterion 1.3.2: Meaningful Sequence

Content must be presented in a meaningful order that preserves its intended meaning and usability—regardless of how it is rendered (visually, via assistive technologies, or in alternative layouts).

WCAG 2.2 – Success Criterion 2.4.3: Focus Order

When navigating with a keyboard, the focus order must follow a sequence that preserves meaning and operability.

Steps for Identifying a Meaningful Sequence

1. **Check Heading Structure for Logical Order**
    - Does the page begin with a clear, overarching heading (<h1>)?
    - Are headings used hierarchically (<h1> → <h2> → <h3>)?
    - Does relevant content follow each heading logically?
2. **Simulate Reading Flow**
    - Can the content be read top-to-bottom in a logical order?
    - Does the page make sense when read linearly (as screen readers do)?

  - Is information presented in a sequence that matches user expectations and supports task completion?
3. **Evaluate Focus Order**
  - Use the Tab key to navigate: Does the focus move through interactive elements (e.g., navigation, forms, buttons) in a logical and expected order?
  - Are focusable elements placed in the DOM in a sequence that mirrors their visual arrangement?
4. **Use Assistive Technologies for Testing**
  - Test the page with a screen reader: Is content announced in a coherent and meaningful sequence?
  - Check if off-screen elements (e.g., "Skip to content" links) appear at appropriate points in the navigation order.
5. **Analyze Structure and Code**
  - Inspect the HTML structure (DOM) to verify that the order of elements reflects a logical content flow.
  - Identify layout techniques (e.g., CSS positioning or Flexbox/Grid) that might visually reorder elements without changing their DOM position—this can confuse assistive technologies.

Examples of Correct vs. Incorrect Sequence

Scenario	**Correct Implementation**	**Incorrect Implementation**
Page heading before intro text	<h1> appears before body content	Heading appears after several paragraphs of text
Menu structure	Navigation order in the source matches the visual layout	DOM order differs significantly from visual display
Form fields	Fields appear in the natural order of user interaction	Fields are coded out of logical sequence, making keyboard navigation confusing

Summary

A **meaningful sequence** in WCAG terms exists when the content is presented in a **logical, predictable order** that is understandable both visually and technically (in source code, keyboard navigation, and assistive technologies).

What should be checked:

- Headings follow a correct and consistent **hierarchy**
- Text, lists, and tables are **logically ordered** in the source code

- Navigation and content areas appear in the **same order** in the DOM and visually
- Form fields follow the **intended workflow**
- **No CSS-based layout tricks** (like absolute positioning) break the semantic or reading order

This ensures that all users—regardless of the device or assistive technology used—can navigate and understand the content effectively

4.3.2 Example: Detailed Evaluation of HTML Structural Elements

Test Criterion and Procedure
WCAG 2.2 – Success Criterion 1.3.1: Info and Relationships
Information, structure, and relationships conveyed through presentation must also be programmatically determined or available in text. This includes using semantic HTML elements to define the structure of the page, so that assistive technologies can interpret and navigate content effectively.

Description
This evaluation focuses on whether HTML5 structural elements such as <header>, <nav>, <main>, <section>, <article>, <aside>, and <footer> are used correctly and purposefully to represent the logical and semantic structure of a webpage. Proper use of these elements enhances orientation, navigation, and content understanding—particularly for users relying on assistive technologies.

What Is Evaluated?

- Are semantic elements used where appropriate?
- Do the elements match the actual function and purpose of the content?
- Are navigation regions marked with <nav> and clearly distinguishable using aria-label or aria-labelledby, if there are multiple?
- Is the main content enclosed in a single <main> element?
- Are self-contained content blocks marked up using <article> or <section>?
- Are page headers and footers properly defined using <header> and <footer>?

Correct vs. Incorrect Use

Element	Correct Usage Example	Incorrect Usage Example
<header>	Contains page header or introductory section	Used for styling or arbitrary layout dividers
<nav>	Wraps main or secondary navigation blocks	Navigation built only with <div> elements, lacking semantics
<article>	Used for standalone content like blog posts or news items	Used as generic container for layout or non-independent content
<main>	Used once per page to group primary content	Missing entirely or used more than once on a single page

Why This Matters

- Screen reader users can navigate between structural regions (e.g., from <main> to <nav>).
- The page structure is easier to understand for all users, improving usability.
- Consistent, semantic markup supports keyboard navigation and alternative output formats (e.g., Braille).

Steps to Determine Proper Use of HTML Structural Elements

1. **Analyze the Types of Content**
   - Use <h1> to <h6> for heading hierarchies
   - Use <p>, <ul>, <ol>, <li> for paragraphs and lists
   - Use <table>, <thead>, <tbody>, <th>, <tr>, <td> for tabular data
2. **Evaluate Semantic Appropriateness**
   - Is a list marked as a real list, not styled text?
   - Is a heading element used for actual headings, not just styled <div>?
   - Are tables only used for data, not for layout?
   - Are related form fields grouped using <fieldset> and <legend>?
3. **Check Consistency Between Visual and Semantic Structure**
   - Does the visual structure match the semantic HTML?
   - Are landmark elements like <nav>, <main>, and <footer> used consistently and appropriately?
4. **Test with Screen Readers or Code Inspection**
   - Are heading levels announced correctly and in logical order?
   - Are paragraphs, lists, and tables read in full and with appropriate structure?

Examples of Correct vs. Incorrect Implementation

Content Type	**Correct Implementation**	**Incorrect Implementation**
Main heading	Accessible Website	Accessible Website
List	- First item - Second item	Styled list using dashes and line breaks in plain text
Table	Semantic <table> with headers and rows	Layout created with nested <div> elements
Navigation	<nav>...navigation items...</nav>	Navigation structure using unmarked <div> containers

Summary

The proper use of semantic HTML structural elements is essential for meeting **WCAG 2.2 Success Criterion 1.3.1**. It ensures that content is:

- **Programmatically understandable**
- **Semantically meaningful**
- **Usable across assistive technologies and alternative interfaces**

This contributes significantly to the overall accessibility and usability of a website for all users

4.4 Distinguishable

Key Terms: distinguishable, color contrast, color differentiation, color coding, focus indicator, text scaling

Based on WCAG 2.2 – Principle 1: Perceivable, Guideline 1.4: Distinguishable

Definition

The term *"distinguishable"* refers to the requirement that digital content must be designed so that users can clearly perceive and visually differentiate all information. This applies to text, visual elements, and controls, which must be easy to recognize through sufficient contrast, clear structure, and non-reliance on color alone. Content must remain legible and must not overlap or blur when resized or accessed in alternative viewing modes.

Key Areas and WCAG Success Criteria

Area	Requirement	WCAG Success Criterion
Text color contrast	Minimum 4.5:1 for standard text, 3:1 for large text	1.4.3 Contrast (Minimum) (AA)
Graphical user interface	At least 3:1 contrast with adjacent colors for icons, buttons, etc.	1.4.11 Non-Text Contrast (AA)
Use of color	Color must not be the only means of conveying information	1.4.1 Use of Color (A)
Text resizing	Text must scale up to 200% without loss of content or functionality	1.4.4 Resize Text (AA)
Background clarity	Text must remain readable against patterns, gradients, or images	1.4.3 & 1.4.5 Images of Text (AA)
Visible focus indicator	Focus must be clearly visible for keyboard users	2.4.7 Focus Visible (AA)
Text spacing and readability	Support for adjusting line, word, and character spacing	1.4.12 Text Spacing (AA, WCAG 2.1)

Practical Examples of Applying "Distinguishable"

To ensure clarity and usability, especially for users with visual impairments or in challenging environments, the following practices should be applied:

- Maintain a **minimum contrast ratio** of:
  - 4.5:1 for standard text
  - 3:1 for large text or graphical elements
- Ensure **interactive elements** (buttons, icons, form fields) are visually distinct from the background.
- Make the **keyboard focus** clearly visible, using outlines or visual highlights.
- Do not rely solely on **color to convey meaning** (e.g., "errors shown in red" must also have a text or icon).
- Avoid placing text over **low-contrast or patterned backgrounds**.
- Use **large, easily recognizable controls** for better usability with low vision.

- Enable **text scaling up to 200%** without losing layout integrity or functionality.
- Maintain **consistent text structure**, adequate font size, and logical visual grouping of content.

Summary

“Distinguishable” means that all content, controls, and visual elements must be designed in a way that:

- Is **clearly perceivable**, even with limited vision or color perception
- Uses **adequate contrast and spacing**
- Communicates information through **multiple visual cues** (not just color)
- **Supports text scaling and focus visibility**

These requirements, defined in **WCAG Guideline 1.4**, are essential for ensuring that digital interfaces are readable, navigable, and functional for a wide range of users—including those with low vision, color blindness, or cognitive processing differences.

4.4.1 Details on Contrast Requirements for Text and Graphic Content

Based on WCAG 2.2 – Success Criteria 1.4.3 (Contrast – Minimum) and 1.4.11 (Non-text Contrast)

Purpose

This section explains how to evaluate whether text and graphical elements provide **sufficient color contrast** to ensure readability for all users, including those with visual impairments.

Contrast Requirements by Content Type

Content Type	**Minimum Contrast Ratio Requirement**
Standard text (up to 18 pt or 24 px)	4.5:1 against background
Large text (18 pt or 14 pt bold and larger)	3:1 against background
Graphical elements (e.g., icons, buttons)	3:1 against adjacent or background colors
Logos and brand names	Exempt from contrast requirements (for design flexibility)

Relevant WCAG Success Criteria

- **1.4.3 – Contrast (Minimum)**: Ensures that foreground text is readable against its background.
- **1.4.11 – Non-text Contrast**: Applies to active UI components and visual indicators.

What to Evaluate

- All **visible text content**, including:
  - Body text and headings
  - Button labels and form instructions
  - Navigation links and labels
- All **graphical UI components**, including:
  - Icons, interactive elements, and visual controls
  - Status indicators, infographics, and charts

Exemptions:

- Decorative elements
- Logos and brand names
- Hidden or non-functional text

Transparent backgrounds, gradients, and background images must also be considered—what matters is the **actual visual contrast** in the rendered layout.

Example Evaluations

Example	Contrast Ratio	Result
Black text on white background	21:1	Pass
Light gray text (#AAAAAA) on white	1.8:1	Fail
White text on dark blue (#003366)	12:1	Pass

Contrast Testing in Practice

1. **Identify all text and graphic elements**
   - Include headings, paragraphs, labels, links, form inputs, buttons, etc.
   - Verify text size to apply the correct contrast threshold.
2. **Determine background color or image**
   - Consider any gradient, transparency, or background image.
3. **Get color values**
   - Use browser developer tools or color pickers to extract HEX or RGB values.

4. **Calculate contrast ratio**
   - Use standardized tools such as:
     - WebAIM Contrast Checker
     - Colour Contrast Analyser (TPGi)
     - axe DevTools
     - WAVE (WebAIM Evaluation Tool)
     - Google Lighthouse
5. **Assess against WCAG thresholds**
   - 4.5:1 for standard text
   - 3:1 for large text and graphical elements
   - If contrast is too low, adjust colors accordingly.
6. **Evaluate graphical elements separately**
   - Check that **icons, charts, and status indicators** meet at least 3:1 contrast.
   - Confirm that color is **not the only** means of conveying meaning—use shapes, labels, or symbols as well.

Summary

WCAG Success Criteria **1.4.3** and **1.4.11** require that:

- All **visible text** and **interactive graphic elements** meet **minimum contrast ratios** for readability.
- These criteria ensure that content remains accessible to users with **low vision or color blindness**.
- Contrast ratios are **quantifiable** and should be tested using reliable tools.
- If minimum values are not met, **color adjustments** (e.g., darker backgrounds, brighter text) are required.

Meeting contrast requirements is a measurable and essential aspect of **perceivable and inclusive web design**.

4.4.2 Limiting Reliance on Color

Key Terms: distinguishable, color contrast, color differentiation, color coding, focus indicator, text scaling

Based on WCAG 2.2 – Success Criterion 1.4.1: Use of Color

Definition

According to WCAG, color must **not be the only means** of conveying information, indicating an action, prompting a response, or distinguishing a visual element. This ensures that users with color vision deficiencies or complete color blindness can access content equally.

WCAG Success Criterion 1.4.1 – Use of Color (Level A) states:

Color must not be used as the only visual means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.

What to Check

1. **Color-coded information**
   - Are form errors, required fields, or status messages **indicated only by color** (e.g., red borders)?
   - Is there accompanying **text, icon, or pattern** to clarify the meaning?
2. **Form fields**
   - Do error states include **textual explanations** (e.g., "Please fill out this field") in addition to red highlights?
3. **Charts and infographics**
   - Are data lines or regions **only color-coded**, or are they also **distinguished by labels, textures, shapes, or patterns**?
4. **Buttons and interactive states**
   - Are active, focused, or selected elements marked only by color?
   - Are additional indicators like **borders, shadows, icons**, or text used?

Correct vs. Incorrect Implementation

Use Case	Accessible Example	Inaccessible Example
Required form field	Red border **plus** text "Required" or symbol (e.g., *)	Only red border, no text or icon
Error message	Highlighted field **plus** text "Please correct this" and icon	Only red border, no explanatory text
Chart visualization	Color-coded lines **plus** labels, patterns, or legends	Lines differ only by color, no labels or legend
Button focus	Focused state shown by **color + outline or shape change**	Focus indicated only through color

Use Case	Accessible Example	Inaccessible Example
Navigation indicator	Active link shown with **underline or bold style + color**	Active link shown with color only

Best Practices

1. **Add textual support**
    - Pair color indicators (e.g., red for errors) with **text explanations**.
    - Example: Show "* Required field" instead of relying solely on red styling.
2. **Use symbols and visual markers**
    - Add **icons** (e.g., ⚠ or ✔), **borders**, or **shapes** to reinforce meaning.
    - Use **patterns or line styles** in charts and graphics.
3. **Explain visual indicators**
    - If symbols (e.g., *) are used, include a **legend** (e.g., "* = Required field").
4. **Avoid color-only focus indicators**
    - Use visible outlines, background changes, or icons in addition to color.
5. **Check implementation in CSS**
    - Ensure CSS does not rely solely on color to differentiate interactive or state-based elements.
6. **Test with color-blind simulators and grayscale settings**
    - Tools:
        - **Color Oracle**, **Coblis**, **NoCoffee Vision Simulator**
        - Browser dev tools with grayscale filter
    - Manual check: View content in grayscale and ensure it remains understandable.

Summary

WCAG Success Criterion **1.4.1: Use of Color** requires that **color must never be the only method** for conveying information. Any color-coded element must include **at least one additional indicator**, such as:

- Text labels
- Icons or symbols
- Patterns or visual shapes

This ensures users with color vision deficiencies can access the same information and functionality as all other users. Meeting this criterion is essential for providing **perceivable and inclusive digital content**.

4.5 Keyboard Accessible

Key terms: keyboard accessibility, focus order, keyboard focus, keyboard traps, device independence, input modalities

Keyboard accessibility means that all functions and interactive elements of a website or application must be fully operable **without using a mouse**. Users must be able to access, control, and activate all content **exclusively via the keyboard**. This includes navigating between focusable elements using the **Tab key**, activating links and buttons using the **Enter** or **Space** key, and relying on a **visible focus indicator** to show which element is currently focused. For certain components - such as menus, dropdown lists, or carousels - **arrow keys** must also be usable for navigation within the element.

4.5.1 Device-Independent Operability of Web Pages

A core requirement for accessible websites is the **complete and reliable operability of all content and functions regardless of the input device used**, especially by keyboard alone.

Success Criterion and Evaluation Approach

- **WCAG Success Criterion 2.1.1: Keyboard (Level A)**

This criterion ensures that **all interactive elements** on a website or web application are **fully reachable and usable via keyboard only**, with **no mouse or gesture interaction required**. It confirms compliance with WCAG requirements for device independence.

Description
Device-independent operability means that all content and functionality of a website must be usable **regardless of the input or output technology**. This includes:

- Keyboard-only navigation
- Touch input
- Voice control
- Screen readers

- Braille displays

Core requirements include:

- All content must be usable **without special hardware**
- No function must rely solely on mouse, touch, or gesture input
- The **focus order** must be **logical and consistent**
- The **keyboard focus** must always be **visible and controllable**

Importance under WCAG:
Keyboard accessibility is essential for users with motor or visual disabilities. WCAG requires that **every function** of a web page be **operable using only the keyboard**.

What does keyboard accessibility include?

- **Navigation**: Through all links, buttons, and form fields using **Tab** and **Shift + Tab**
- **Activation**:
  – Buttons with **Enter** or **Space**
  – Links with **Enter** only
- **Visible focus**: Clearly visible on every element (minimum contrast ratio 3:1)
- **Dialogs and modals**: Must be fully operable and closable with the keyboard
- **No keyboard traps**: Focus must not become “stuck” in any component
- **Groups and states**: Menus, radio button groups, and similar elements must support **arrow key navigation**

Relevant WCAG Success Criteria

Criterion	Name	Level	Description
2.1.1	Keyboard	A	All functionality is operable via keyboard
2.1.2	No Keyboard Trap	A	Focus can always be moved away via keyboard
2.1.4	Character Key Shortcuts	A	Single-character shortcuts must be cancellable
2.4.3	Focus Order	A	Tab order must be logical and meaningful
2.4.7	Focus Visible	AA	Focus must be clearly visible
2.4.11	Focus Appearance (WCAG 2.2)	AA	Focus indicator must meet contrast and size requirements
3.2.1	On Focus	A	Focus must not trigger unexpected changes

Criterion	Name	Level	Description
3.2.3	Consistent Navigation	AA	Navigation mechanisms must be consistent across pages

Examples of evaluation steps

1. Put aside the mouse and avoid using it
2. Use **Tab** to navigate through the page
3. Check that all interactive elements are **focusable and visible**
4. Use **Enter** or **Space** to activate them
5. Use **Shift + Tab** to move backward
6. Open modal dialogs and close them with **Escape** or an appropriate button

Well-implemented examples

Use Case	Meets Requirements
Tab navigation through menu	All items are focusable in a logical order
Button activation	Buttons respond to **Enter** and **Space**
Modal dialog interaction	Dialog can be fully operated and closed via keyboard
Focus visibility	Focus is clearly indicated (e.g., high-contrast outline)

Common issues

Problem	Effect
Elements only operable by mouse	Users without a mouse cannot access features
Invisible or disappearing focus	Users lose orientation
Illogical focus order	Navigation becomes confusing and frustrating
No way to exit dialog via keyboard	Users cannot leave modal windows without a mouse

Evaluation Summary

WCAG SC 2.1.1 ensures that all interactive content and functionality are **fully operable via keyboard**, **independent of other input methods**.

The following must be tested:

- All elements are **focusable and operable**

- The **focus order is logical**
- There are **no keyboard traps**
- **Focus visibility is maintained**
- **Complex components** (e.g., menus, dialogs, sliders) are keyboard accessible

Recommended tools

- WAVE (WebAIM Accessibility Evaluation Tool)
- axe DevTools
- Browser dev tools with keyboard inspection features

Recommended remediation measures

- Ensure **all elements** are operable via standard keyboard inputs
- Maintain **logical tab order** in the HTML structure (avoid tabindex > 0; use tabindex="0" for additional elements, tabindex="-1" for programmatic focus)
- Do not suppress focus outlines in CSS (avoid outline: none)
- Use appropriate **ARIA roles and keyboard interaction support** for complex widgets
- Provide **keyboard-accessible exit options** (e.g., **Escape** for modals)

Conclusion
Device-independent access is a fundamental criterion of digital accessibility. Websites and applications must be designed so that all users—regardless of their input device—can interact equally. Full keyboard operability is a foundational requirement for accessibility evaluation.

4.6 Sufficient Time

Key terms: time limit, timeout, extension option, real-time simulation

The principle of **Sufficient Time** is based on **WCAG Success Criterion 2.2.1 Timing Adjustable**, along with related criteria. It means that users must be given **enough time** to read content and complete tasks without being hindered by automatic timeouts or forced interruptions.

If a time-limited activity lasts longer than 20 hours, it is considered an exception under WCAG. In that case, there is no requirement to provide an option to extend, disable, or control the time limit.

What is required?
If there are time restrictions (e.g., automatic session timeouts, expiring forms, inactivity logouts), **at least one** of the following options must be provided:

- The ability to **turn off** the time limit.
- The ability to **extend** the time limit.
- A **warning** before the time expires and an option to **request more time**.

Additionally:

Content must not change or disappear so quickly that it cannot be consumed at the user's own pace (e.g., **auto-advancing carousels**).

Examples of sufficient time in practice

Example	Implementation
Online form with timeout	A message appears before timeout allowing users to **extend the session**.
Auto-advancing slideshow	Users can **pause or manually control** the slide rotation.
Online banking session expiration	A warning is displayed before timeout with an option to **extend the session**.

Summary
Sufficient Time means that all users can interact with digital content **at their own pace**. Time limits must **not be unexpected** and should be **controllable or extendable**. This ensures that users are not excluded due to processing speed, cognitive load, or use of assistive technologies.

4.6.1 Time Limits and Accessibility Requirements

When does a time limit implementation meet accessibility requirements?

A time limit implementation complies with accessibility requirements according to **WCAG 2.2 Success Criterion 2.2.1 (Timing Adjustable)** when at least one of the following conditions is met:

1. Disabling the Time Limit
Users can completely **turn off** the time limit, allowing unlimited time to read or interact with content.

2. Extending the Time Limit

Users can **adjust or extend** the time limit themselves (e.g., “Extend by 5 minutes”).

3. Warning Before Expiration

A **warning message** is shown before the time expires, giving users a chance to extend time or complete their task.

4. Justified Time Limits

If a time limit is **technically or legally required** (e.g., security in financial transactions), a warning or extension must still be provided where possible.

5. Exception Cases

This requirement does **not apply** if:

- The time limit is an **essential part of a real-time simulation** (e.g., live competitions),
- The time limit is **inherent to the activity** and cannot be changed (e.g., auctions),
- The time limit is so long that it is **unlikely to cause barriers** (e.g., 20 hours or more).

WCAG-Compliant Examples

Use Case	**Meets Requirement If...**
Online form with timeout	A message appears before timeout: “Your session will expire in 1 minute. Extend?”
Online test or survey	Users can select or adjust the time limit at the beginning or during the session.
Auto-closing pop-up	A countdown is visible with a “Extend” button to delay closure.

Summary

A time limit is accessible when **users remain in control**: it must be **disablable**, **extendable**, or paired with a **clear warning**. Only **justified and unavoidable** exceptions are permitted.

Corresponding WCAG Evaluation Guidance

Success Criterion 2.2.1: Timing Adjustable

This criterion ensures that websites and applications with time-dependent content provide users **enough time** to perceive content and complete interactions.

What to Evaluate

- Are there **time limits** (e.g., auto-logouts, session timeouts, timed forms, rotating content)?

- Are users given the ability to:
    - **Turn off** the time limit,
    - **Extend** the time, or
    - Receive a **warning** before time runs out?
- Can auto-rotating or updating content (e.g., **slideshows**) be paused or manually controlled?

Typical Testing Procedure

Step	**Procedure**
Identify time limits	Does the page include automatic time-based actions?
Trigger and observe time limits	Is a warning shown? Can users extend the time?
Check rotating content	Can users pause or manually control carousels or sliders?
Evaluate timed forms	Is there a notification and option to continue editing before timeout?

Common Accessibility Failures

- Sudden session expiration with **no warning** or **extension** option.
- Auto-updating content that **cannot be paused**.
- Lack of user control over **rotating banners** or **pop-ups**.

Summary

This evaluation step verifies compliance with **WCAG 2.2.1** by checking that users have **sufficient time** to interact with all time-limited content. Websites must ensure that timeouts are **disablable**, **extendable**, or accompanied by **advance notice**. This supports users who need additional time due to cognitive, physical, or technological constraints.

.

4.7 Seizures

Terms: Flickering, flashing, epilepsy, visual stimuli, photosensitivity, PEAT

The term “seizures” in the context of digital accessibility refers to the risk that certain visual content—especially rapidly flashing, blinking, or flickering elements—can trigger epileptic seizures in individuals with specific neurological sensitivities, particularly those affected by photosensitive epilepsy.

4.7.1 Testing for Flickering Content on Websites

Evaluation Criteria and Testing Requirement

• WCAG Success Criterion 2.3.1: Three Flashes or Below Threshold (Level A)

• Testing Requirement: *Check whether any flashing content exceeds the flash threshold and poses a risk for users prone to seizures.*

Description

Flickering on websites becomes a significant accessibility barrier when it exceeds certain thresholds or occurs without user control. This is especially critical for people with photosensitive epilepsy, but also affects individuals sensitive to intense visual stimulation.

Critical Conditions for Flicker-Related Barriers

1. **Rapid Flashing (More than Three Flashes Per Second)**
   – Content flashing more than three times per second can trigger epileptic seizures in sensitive individuals and may also cause nausea, dizziness, headaches, concentration issues, or sensory overload.
2. **High Contrast Effects**
   – Bright flashes on a dark background or highly saturated colors (especially red) significantly increase the risk.
3. **Large Display Areas**
   – The larger the flashing element, the higher the potential risk.
4. **Lack of User Control**
   – If users cannot pause or stop the flashing, it becomes a persistent barrier.
5. **Unexpected or Unpredictable Effects**
   – Sudden blinking or unanticipated animations can lead not only to seizures but also to anxiety, stress, or cognitive overload.

Typical Testing Procedure

Test Step	Method
Visual inspection	Look for banners, animations, or videos with flickering effects.

Test Step	Method
Count flashes	Check if the flashing occurs more than 3 times per second.
Evaluate size and contrast	Assess how large and intense the flickering elements are.
Check for user control	Confirm whether the content can be paused, stopped, or hidden.
Assess element behavior	Does the flashing start suddenly without warning?

Common Violations

- Automatically starting animated banners with strobe-like effects
- Advertisements or pop-ups that flash more than 3 times per second
- Videos with rapid light transitions and no warning or stop option

Technical Tools for Flicker Testing

1. **Automated Evaluation Tools**

Tool	Function
WAVE	Flags flashing content and known visual risks
axe DevTools	Detects risky animations and flicker effects
Google Lighthouse	Chrome-integrated testing for animated content

2. **Specialized Flicker Detection Tools**

Tool	Function
PEAT (Photosensitive Epilepsy Analysis Tool)	Analyzes flash frequency according to WCAG thresholds
Trace Center Toolkits	Includes checkers for GIFs, videos, and embedded animations

3. **Browser Extensions**

Tool	Function
NoBlink (Firefox)	Detects and blocks flashing content in the browser
Adblockers with filters	Often block flashing ads and animated pop-ups

4. **Manual Tools**

- Slow-motion screen recorders: For manually counting flashes
- Grayscale mode (OS feature): Makes flicker contrast easier to detect

Recommended Testing Workflow

1. Fully load the web page
2. Run automated testing using tools like PEAT or axe DevTools
3. Conduct manual visual inspection for blinking banners, videos, and ads
4. If necessary, use screen recording to analyze flash frequency
5. Simulate blocked flashing with ad blockers to assess content usability

Conclusion
Fast-flashing content poses a serious risk to users with epilepsy and can negatively impact many others. Websites must comply with WCAG guidelines regarding flash frequency, size, contrast, and user control. Testing for flashing content should include both automated and manual steps and is a critical part of any accessibility audit.

4.8 Navigable

Key Terms: Navigability, Focus Navigation, Skip Link, Page Structure, Logical Order

“Navigable” means that users can easily locate, understand, and access the content of a website or application.

All page content should be structured, labeled, and organized so that:

1. **Navigation is predictable and consistent**
    - Menus, link positions, and interactive elements remain in consistent locations and follow a logical structure.

      Relevant WCAG Success Criteria:
    - 3.2.3 Consistent Navigation
    - 3.2.4 Consistent Identification
2. **The order of content is meaningful**
    - The tab order for keyboard users follows the logical reading order and visual layout.

      Relevant WCAG Success Criteria:
    - 1.3.2 Meaningful Sequence
    - 2.4.3 Focus Order
3. **Headings, labels, and links are clear and descriptive**
    - Users always know where they are and what content is behind a link or button.

      Relevant WCAG Success Criteria:
    - 2.4.4 Link Purpose (In Context)
    - 2.4.6 Headings and Labels
    - 4.1.2 Name, Role, Value
4. **Structural elements are used**
    - Semantic structural elements (e.g., <nav>, <header>, <main>, <footer>) help users quickly grasp the content.

      Relevant WCAG Success Criteria:
    - 1.3.1 Info and Relationships
    - 4.1.1 Parsing
5. **Orientation aids are provided**
    - For example: breadcrumbs, tables of contents, “skip to main content” links.

      Relevant WCAG Success Criteria:
    - 2.4.1 Bypass Blocks
    - 2.4.8 Location

Summary:
Navigable means that all content and functions are organized in a way that users can easily find, logically understand, and access them without barriers — regardless of input method or technology used.

4.8.1 Clear and Logical Focus Order for Keyboard Operation

Success Criterion and Evaluation Guidance

• WCAG Success Criterion 2.4.3: Focus Order (Level A)

• Evaluation Recommendation: *“Verify that the focus order on the webpage follows a logical and consistent sequence.”*

This ensures that users who rely on keyboard navigation can reach all content in a meaningful, predictable order — without unexpected jumps or disorienting behavior.

Description

A clear and logical focus order for keyboard operation means that users can access all content and interactive elements using the Tab key in a traceable, linear sequence. This order should reflect both the **visual layout** and **functional structure** of the page.

Criteria for Logical Focus Order:

1. **Reflect the Visual Structure**
   – Focus follows the logical reading direction: top to bottom, left to right
   – Content is navigated in the order it is visually presented
2. **Respect Functional Priorities**
   – Key functions such as navigation, search, and main content come first
   – Secondary content like sidebars or footers follows afterward
3. **Avoid Jumps and Inconsistencies**
   – Focus should not jump unpredictably between sections
4. **Ensure Complete Accessibility**
   – No interactive element may be skipped in the Tab order
5. **Prioritize Accessibility Features**
   – Skip links (e.g., “Skip to main content”) or accessibility settings appear at the beginning
6. **Handle Modal Dialogs Correctly**
   – When a dialog opens, the focus stays inside it
   – After closing, the focus returns to a logical position

Examples

Positive Implementations:

Page Element	Keyboard Navigation Order
Standard Web Page	Logo → Main Navigation → Search → Main Content → Sidebar → Footer

Page Element	Keyboard Navigation Order
Contact Form	Name → Email Address → Message → Submit Button
Modal Dialog	Focus jumps into the dialog on open and returns to a logical position on close

Common Failures:

• Focus jumps from the menu to the footer and then back to the main content

• Visible buttons or links are not reachable with Tab

• Form fields do not follow a logical input sequence

• After closing a modal dialog, the focus lands in an illogical position

Evaluation Methodology:

Step	Procedure
Test Focus Order	Use the Tab key to move through all interactive elements
Check Visibility	Each element must show a visible focus indicator
Test Reverse Navigation	Use Shift+Tab to verify the reverse order
Check Modal Dialogs	Ensure focus remains within the dialog and returns logically after closing

Summary

WCAG Success Criterion 2.4.3 ensures that web pages have a clearly structured and predictable Tab order. Focus navigation must follow the page's **visual hierarchy** and **functional structure** so that users relying on keyboards can operate the content comprehensively, logically, and without confusion.

A correct focus order is achieved when:

• The focus follows the page's visual and logical layout

• No interactive elements are skipped

• Dialogs manage focus correctly

• No sudden or illogical jumps occur

This guarantees that people with motor disabilities or without access to a mouse can use websites effectively and without barriers.

4.8.2 Visibility of Keyboard Focus

Success Criterion and Evaluation Guidance

• WCAG Success Criterion 2.4.7: Focus Visible (Level AA)

• Evaluation Recommendation: *“Check whether the keyboard focus is clearly visible to ensure accessibility.”*

This criterion requires that the focus indicator is always visually perceptible during keyboard navigation, providing users with clear orientation.

Description

A visible focus indicator is essential for navigating a website using the keyboard. It shows which element is currently active and ready to receive input. If the focus is not visible or is hard to perceive, it creates a barrier for users with motor or visual impairments.

Detailed Requirements:

1. **Focus must be visually discernible**
   – When using the Tab key, it must always be clear which element is currently focused
   – Applies to all interactive elements: links, buttons, forms, widgets, menus
2. **Focus must have sufficient contrast**
   – The focus indicator (e.g., border, color change) must clearly stand out from the background
   – Thin lines or subtle color changes are insufficient
   – Recommended minimum contrast ratio with adjacent content: 3:1
3. **Focus must not be removed**
   – CSS rules like outline: none; are not allowed unless replaced with an equally visible style
4. **Focus must persist in all states**
   – Focus should remain visible in open dropdowns, forms, menus, modal dialogs, etc.
5. **Focus must not jump or disappear**
   – The focus path must remain logical and continuous — visual focus should not be lost due to transitions or effects

Examples

Positive Implementations:

• A clear, high-contrast outline around the focused element

• A noticeable background color change when an element receives focus

• Additional visual cues such as underlining focused links

Common Failures:

• The focus indicator is not visible or barely noticeable
• Focus outline is too thin or low-contrast
• Focus disappears when opening dropdowns or modal dialogs
• Focus is suppressed or removed by CSS
• Focus visibility is inconsistent, e.g., only appears on hover

Evaluation Methodology:

Step	**Procedure**
Perform Tab Test	Navigate the website using the Tab key
Observe Focus	Ensure that each active element is visually highlighted
Check Color Combinations	Verify that the focus indicator has adequate contrast with the background
Test Complex Components	Dropdowns, dialogs, sliders: focus must remain visible and operable

Summary

WCAG Success Criterion 2.4.7 ensures that the keyboard focus is clearly and visibly indicated on all interactive elements. Focus visibility is a fundamental requirement for accessible keyboard navigation.

To allow users who rely solely on the keyboard to stay oriented and interact reliably, the focus must:

• Be constantly visible and recognizable
• Stand out clearly from the background
• Never be lost or removed through CSS
• Function reliably in all component states (e.g., dropdowns, dialogs)

With a reliable and clearly visible focus indicator, full use of a website without a mouse becomes possible — a core principle of digital accessibility.

4.8.3 Descriptive Link Texts

Success Criteria and Evaluation Guidance

• WCAG Success Criterion 2.4.4: Link Purpose (In Context) – Level A
• WCAG Success Criterion 2.4.9: Link Purpose (Link Only) – Level AAA
• Evaluation Recommendation: *“Check whether link texts provide sufficient information and are*

understandable for screen reader users."

This ensures that the purpose of a link is clear through its text alone or within its surrounding context — even when links are accessed in isolation by screen reader users.

Description

A link text is considered descriptive when it clearly communicates the destination or function of the link. The link text must:

• Be clear, concise, and self-explanatory
• Be understandable on its own (e.g., in screen reader link lists)
• Avoid generic terms like "here," "click," or "more"
• Be consistent for identical destinations and distinct for different targets
• Avoid redundancy (e.g., do not begin with "Link to: ...")

Examples

Descriptive vs. Non-descriptive Link Texts:

Context	Good (Descriptive)	Poor (Unclear)
Link to contact page	"Contact Us"	"Click here"
PDF download	"2023 Annual Report (PDF)"	"Download"
Further information	"More about Data Protection"	"Learn more"
Navigation	"Home"	"Back" or "Here"

Evaluation Methodology:

Step	Procedure
Screen Reader Link List	Check if link texts make sense when read in isolation
Manual Reading in Page Context	Determine whether the link's purpose is clear from surrounding content
Navigation and Repeated Links	Verify consistency and clarity of link texts across identical or varied destinations

Common Failures:

• Link texts consist only of "here," "more," "click," or "continue"
• Same link text leads to different pages
• Different link texts lead to the same destination
• Link labels are incomplete or cut off

Why Are Descriptive Link Texts Important?

• **Accessibility:**
Screen reader users often browse lists of links without surrounding context. Vague texts lead to disorientation.

• **Comprehension:**
Users can immediately understand the function or target of the link.

• **Consistency and Orientation:**
Uniformly worded links help users recognize destinations and navigate more easily.

• **Avoiding Frustration:**
Non-specific links force users to click unnecessarily just to understand their purpose.

Summary
WCAG Success Criteria 2.4.4 and 2.4.9 require that link texts clearly, precisely, and understandably describe the purpose of the link — both in-context and in isolation, such as in screen reader link lists.

Descriptive link texts are crucial for making web content accessible. They support all users — especially those using screen readers, or those with cognitive or motor disabilities — in navigating efficiently, safely, and without confusion.

4.9 Input Modalities

Key Terms: Input Modalities, Pointer Gestures, Pointer Cancellation, Motion Activation, Target Size, Accessible Name

Input modalities refer to the different ways users can interact with digital content. This includes all forms of input across various devices or assistive technologies.

Examples of input modalities:
• Keyboard
• Mouse
• Touchscreen
• Voice control

• Gesture control
• Assistive technologies such as screen readers, braille displays, or eye tracking

Relevance in WCAG (Guideline 2.5: Input Modalities):
WCAG requires that all content and functionality must be accessible regardless of the input modality. This means:

1. **No function may depend on a single input method**
   – All interactive elements must be operable using a keyboard.
   (WCAG 2.1 Success Criterion 2.1.1 – Keyboard, Level A)
2. **Controls must be large enough and clearly identifiable**
   – Controls must be at least 24 by 24 CSS pixels to ensure reliable operation via touch, voice commands, or other alternative inputs.
   (WCAG 2.2 Success Criterion 2.5.8 – Target Size, Level AA)
3. **No complex gestures without alternatives**
   – If content relies on multi-point gestures (e.g., swiping, pinch-to-zoom), there must be a simple alternative method of operation.
   (WCAG 2.1 Success Criterion 2.5.1 – Pointer Gestures, Level A)
4. **Support for voice and motion-based input**
   – Users must be able to operate elements through voice control or motion sensors (e.g., device tilt).
   (WCAG 2.1 Success Criteria 2.5.2 – Pointer Cancellation, 2.5.4 – Motion Actuation)

Summary:
Input modalities include all forms of interaction with digital content. Accessible content must be designed so that it can be used regardless of the chosen input method — whether via keyboard, mouse, touch, voice, or assistive technology.

4.9.1 Visible Labels as Part of the Accessible Name

Success Criteria and Evaluation Guidance
• WCAG Success Criterion 2.5.3: Label in Name – Level A
• WCAG Success Criterion 3.3.2: Labels or Instructions – Level A
• Evaluation Recommendation: *“Check whether form elements are clearly labeled and understandable for all users.”*

This criterion ensures that interactive elements (e.g., buttons, form fields, links) have clear and specific labels, and—critically—that the visible text is included in the accessible name used by assistive technologies.

Description

In digital accessibility, the *accessible name* refers to the text string used by screen readers or voice control systems to identify an element. It can be derived from HTML elements such as <label>, aria-label, title, or other attributes.

WCAG Success Criterion 2.5.3 “Label in Name” requires that the visible label of an interactive element must also be part of its accessible name.

Why is this important?

1. **Consistency between visual and spoken information**
   – If the visible text (e.g., “Submit”) is not part of the accessible name, screen reader users may hear something different from what sighted users see. This leads to confusion and errors.
2. **Reliable voice control**
   – Voice control systems like Dragon NaturallySpeaking only work when users speak the visible text. If that text is not part of the accessible name, the system won’t recognize the command.
3. **Trust and predictability**
   – Consistent labeling between what users see and what is spoken or recognized builds trust and ensures a predictable user experience.
4. **Accessible navigation and form handling**
   – Mismatches between visible and accessible labels in form fields, buttons, or links can cause disorientation or input errors, especially for screen reader users.

Examples

Visible on Button	**Accessible Name (Screen Reader / Voice Control)**	**Result**
Submit	Submit	Clear and correct – both match
Submit	Send the form	Confusing – mismatch between visual and spoken
Submit	Submit the form now	Acceptable – visible text is fully included

Evaluation Methodology

Step	Procedure
Visual Inspection	Check if each form element has a visible, understandable label
Source Code Review	Verify that <label> is correctly linked to the input using the for attribute
Screen Reader Test	Use a screen reader to confirm the visible label is accurately read aloud
Form Input Testing	Check whether required fields are clearly identified without relying on visual cues

Common Failures

• Form elements without visible or programmatically associated labels
• <label> not properly linked (missing for or id)
• Labels that are too vague, such as “Input” or “Data”
• Error messages not directly associated with the corresponding field
• Mismatches between visible text and accessible name (e.g., due to incorrect aria-label)

Summary
WCAG Success Criteria 2.5.3 and 3.3.2 ensure that all form elements are labeled clearly, unambiguously, and consistently—both visually and programmatically.
It is especially critical that the **visible label is included in the accessible name**, so that screen readers and voice control systems can interpret and interact with content correctly.

This guarantees an accessible, intuitive, and trustworthy user experience for all users.

4.10 Readable

Key Terms: Readability, Primary Language, Language of Text Passages, Language Switching

WCAG Reference: Guideline 3.1 – Readable

Goal: Content must be designed and marked up so that it is linguistically understandable, correctly identified, and easy to process for all users — regardless of assistive technology or cognitive ability.

Description
“Readable” refers to the linguistic clarity and correct language markup of digital content. These requirements are particularly important for users with learning disabilities or cognitive limitations, as well as users of screen readers and translation tools.

Core Requirements for Readability (WCAG 3.1):

1. **Specify the page's primary language**
   – The default language of the page must be correctly set in the HTML code (e.g., <html lang="en"> for English).
   – This ensures that screen readers and translation tools can render content correctly.
   → *WCAG Success Criterion 3.1.1 Language of Page*
2. **Mark changes in language within the content**
   – Words, sentences, or paragraphs in a different language must be marked using the lang attribute (e.g., bonjour).
   – This improves pronunciation accuracy and avoids confusion.
   → *WCAG Success Criterion 3.1.2 Language of Parts*
3. **Use plain and clear language (Level AAA)**
   – Use short, active sentences wherever possible.
   – Avoid foreign terms or explain them when used.
   – Structure text logically and use simple sentence constructions.
   → *WCAG Success Criterion 3.1.5 Reading Level*
   → Optional: 3.1.3 Unusual Words / 3.1.4 Abbreviations / 3.1.6 Pronunciation
4. **Provide simplified or plain language versions (optional, AAA)**
   – For audiences with limited reading comprehension, offering a simplified version of the content can be helpful or necessary.
   – This is recommended at WCAG Level AAA.
   → *In addition to 3.1.5 Reading Level*

Note:
The following points are sometimes mistakenly considered part of readability but are formally covered under **WCAG Guideline 1.4 – Distinguishable (Principle 1: Perceivable):**
• Color contrast between text and background (1.4.3 Contrast – Minimum)
• Ability to resize text without loss of content or functionality (1.4.4, 1.4.10 Reflow)
• Use of legible fonts and adequate spacing (1.4.12 Text Spacing)
• Avoiding visual distraction from flashing content (2.2.2 / 2.3.1)
These requirements support visual readability but are technically and methodologically separate.

Examples of Requirements from Guideline 3.1 – Readable

Requirement	Example / Implementation
Primary language correctly set	<html lang="en">

Requirement	Example / Implementation
Language change correctly marked	“Download Terms of Use”
Understandable language	“Please enter your email address.” instead of “Kindly insert your electronic contact data into the respective input field.”
Plain language (optional)	Providing a version in plain language or with symbol support

Summary

WCAG Guideline 3.1 Readable ensures that content is linguistically clear, correctly marked, and understandable for all users. It focuses on the correct use of language attributes in code and the application of simple, user-friendly language.

Visual aspects like contrast or font size, while supporting readability, fall under **Guideline 1.4 “Distinguishable”** and are addressed separately.

4.10.1 Verifying the Primary Language of Web Pages

Success Criterion and Evaluation Guidance

• WCAG Success Criterion 3.1.1 – Language of Page
• Evaluation Recommendation: *Is the primary language of the page correctly identified?*

Description

Specifying the primary language of a webpage in the source code (e.g., <html lang="en">) is a fundamental requirement for digital accessibility. It enables screen readers, translation tools, and other assistive technologies to interpret and present content correctly.

If this language attribute is missing or incorrectly set, it can significantly impair comprehension, navigation, and usability of the page.

What should be tested?

• Does the <html> element include the correct lang attribute for the page's main language?
Example: <html lang="en"> for an English-language page.
• Does the declared language match the actual content?
• Are foreign language terms or quotes within the main content also marked with a lang attribute?
Example: Datenschutz

• Is a language defined even on graphic-heavy pages or applications when navigational elements are present?

Common Issues with Missing or Incorrect Language Declaration

1. **Problems for screen reader users**
   – The lang attribute determines correct pronunciation, syllabification, and intonation.
   – Without it, screen readers default to the system language, which may result in incomprehensible output.
2. **Difficult navigation for users with learning or language challenges**
   – If, for example, English content is pronounced using a German voice, this creates extra cognitive load or confusion.
3. **Errors with automated translation tools**
   – Browsers and translation services rely on language declarations to identify text accurately. Without it, they may produce incorrect or faulty translations.
4. **Negative impact on search engine optimization (SEO)**
   – Search engines use the language setting to index content properly. Missing or incorrect language settings can reduce discoverability.
5. **Barriers in multilingual content**
   – If language switches within a text (e.g., quotes or terminology) are not marked, assistive technologies will also mispronounce or misinterpret those sections.

Evaluation Methodology

Step	Procedure
Check the source code	Inspect the <html> tag for a correctly set lang attribute
Visual content review	Ensure the declared language matches the visible main content
Screen reader testing	Verify that text is read aloud in the correct language and is intelligible

Examples

Incorrect Implementation	Why It's Problematic	Correct Implementation
<html> without lang attribute	Screen reader uses system default — incorrect pronunciation	<html lang="en"> for English pages

Incorrect Implementation	Why It's Problematic	Correct Implementation
<html lang="en"> on a German page	German content is read with English pronunciation	<html lang="de">
Foreign words unmarked (e.g., "Download")	Word is mispronounced by screen reader	Download
Language switch in paragraph not marked	Assistive tech fails to detect the switch – wrong pronunciation	Please click Absenden.

Summary

WCAG Success Criterion 3.1.1 requires that the primary language of a webpage be correctly defined in the source code. This ensures that:

• Screen readers pronounce content accurately
• Users can understand linguistic content
• Translation tools work effectively
• The page remains accessible in multilingual and SEO contexts

A missing or incorrect language declaration is considered a **serious accessibility barrier**.

4.11 Predictable

Key Terms: Predictability, Context Change, Navigation, Identification

The term *"predictable"* originates from **WCAG Principle 3: Understandable**, specifically **Guideline 3.2 (Predictable)**.

Predictable means that websites and web applications must be designed so that their functionality, navigation, and behavior of interactive elements are consistent and can be expected by all users.

Core Requirements for Predictability:

1. **No unexpected changes on focus or input**
   – Focusing on or inputting data into an element must not cause automatic content changes (e.g., no instant page reloads when selecting a dropdown option).
2. **Consistent navigation**
   – Navigation menus must appear in the same order and location on all pages.

3. **Consistent behavior of interactive elements**
   – Buttons, links, and controls must behave uniformly across the page or site.
4. **Predictable page behavior**
   – Pages must not launch unexpected modals or autoplay content without user intent or control.
5. **Clear and consistent labeling**
   – Repeating functions must always use the same labels.

Examples of Predictability:

Positive Example	Negative Example (Violation)
Clicking “Proceed to Checkout” reliably leads to checkout.	Clicking a link unexpectedly opens a new window without warning.
Navigation remains consistent across all subpages.	Navigation order or position changes between pages.
Dropdown selection requires confirmation via a button.	Merely focusing on a dropdown option triggers an immediate page change.

Why Is Predictability Important?

• Users with cognitive or learning disabilities may struggle with unexpected changes.
• Keyboard and screen reader users rely on stable and predictable interactions.
• Predictability reduces confusion and enhances usability and accessibility for all users.

Summary:
Predictable means that, in accordance with WCAG, websites and applications behave consistently, transparently, and in a controllable way — ensuring that users always know what will happen and are not surprised by unexpected changes.

4.11.1 Unexpected Context Changes on Input

Success Criterion and Evaluation Guidance
• WCAG Success Criterion 3.2.2 – On Input (Level A)
• Evaluation Recommendation: *Test whether user input is handled consistently and whether unexpected changes occur.*

Description:
WCAG Success Criterion 3.2.2 requires that selecting an option, entering input, or shifting focus must not cause unexpected changes to the page's content, navigation, or layout — unless the change is predictable and initiated by the user through a clear, intentional action (e.g., pressing a button). This applies especially to forms, dropdowns, filters, and interactive controls.

What should be tested?
• Does selecting or entering a value immediately trigger a page reload or content change without confirmation?
• Does the page context change just by moving focus (e.g., via Tab navigation)?
• Do new windows or tabs open without prior notice?
• Does the page scroll or shift layout automatically?

Examples of Unexpected Context Changes on Input:

Example	Description
Dropdown triggers page reload	Selecting a "Product Category" instantly loads a new page without a submit button.
Checkbox triggers navigation	Ticking "Yes, I want to learn more" immediately redirects to another page.
Form field changes layout on focus	Focusing an input field causes other content to disappear or the focus to jump.
Auto-submit on form change	Changing a field value causes the form to submit automatically.
Filter field loads content on focus	Tabbing into a filter triggers dynamic updates without user confirmation.
Link opens new window without notice	A "Learn more" link opens a new browser tab without warning.
Input changes scroll position	Changing a field causes the page to jump unexpectedly to another location.

Evaluation Methodology:

Step	Procedure
Check dropdowns and forms	Does content or the page reload automatically without user confirmation?

Step	Procedure
Test keyboard navigation	Does content change when focus moves via Tab without a triggered action?
Test behavior of links	Are new windows or tabs only opened with prior indication?
Observe dynamic content	Are contents auto-loaded on selection, hover, or focus without interaction?

Common Violations:

• Auto-submit triggered by selecting a dropdown value without a confirmation button
• Context changes triggered by simply focusing a field
• Page reload or navigation caused by checking a checkbox
• New windows or tabs open without visual or textual indication

Summary:
WCAG Success Criterion 3.2.2 ensures that **user input or focus does not result in unexpected context changes**. Users must always confirm their actions before page content changes or new windows open.

This rule is especially important for people with cognitive disabilities, screen reader users, and keyboard-only users, as it protects them from **unpredictable interface behavior** that could cause confusion or disrupt navigation.

4.12 Input Assistance

Key Terms: Input Assistance, Error Message, Error Prevention, User Support

Input assistance means that users receive clear, helpful guidance when interacting with forms and input fields, enabling them to complete entries correctly and accessibly.

These requirements are based on the following WCAG success criteria:

• **WCAG 3.3.1 Error Identification**
If an input error is detected, it must be clearly and understandably described — in direct relation to the field concerned.

• **WCAG 3.3.2 Labels or Instructions**

Forms must be clearly labeled and, where needed, provide instructions on how to enter data correctly.

• **WCAG 3.3.3 Error Suggestions**

Users must be able to understand any errors and, ideally, receive suggestions for how to correct them.

• **WCAG 3.3.4 Error Prevention**

For critical interactions (e.g., legally binding or financial transactions), mechanisms must be in place to prevent errors, such as confirmation dialogs, previews, or undo features.

Overview of Required Input Assistance:

1. **Clear labels**
   – Every input field must have a clear, understandable, and visible label.
2. **Explanatory guidance (placeholders, help texts, tooltips)**
   – Fields requiring specific formats (e.g., “MM/DD/YYYY” for dates) must communicate these clearly.
   – Optional or required fields must be clearly identified.
3. **Proper error messages**
   – When input is invalid, an explicit message must explain what needs to be corrected.
   – Error notifications must be accessible to screen readers (e.g., ARIA live regions or programmatic associations with the field).
4. **Error prevention through confirmation options**
   – Before submitting sensitive data (e.g., purchases, legally binding forms), a confirmation screen should be shown.
5. **Automatic validation and suggestions**
   – Where feasible, features such as input suggestions or autocomplete should be provided to ease data entry.
6. **No time limits without extension options**
   – Users must be given sufficient time to complete entries.

Examples of Good Input Assistance:

Problem Area	Good Solution
Phone number input	Example provided near the field: “Please use the format +1 123 4567890.”
Required fields	Clear note: “* Required field” and explanation at the start of the form
Error message	“Please enter a valid email address (e.g., name@domain.com).”

Problem Area	Good Solution
Forms with legal impact	Display a confirmation page before final submission

Summary:
Input assistance means that all input fields and forms are supported by clear labels, easy-to-understand instructions, and meaningful error messages. This enables users to complete inputs **safely and without barriers**, regardless of their abilities or the technology they use.

4.12.1 Labeling of Form Elements

Reference:
• WCAG Success Criterion 3.3.2 – Labels or Instructions
• Evaluation Recommendation: *Verify that form element labels are accessible and clear.*

Description:
This evaluation step outlines the WCAG 2.1 requirements for the proper and accessible labeling of form elements. The goal is to ensure that all users — including those with visual, motor, or cognitive impairments — can complete forms independently, correctly, and without barriers.

What should be evaluated?
• Every form element (e.g., input fields, select boxes, checkboxes, radio buttons, buttons) must have a **visible and understandable label**.
• The label must be **programmatically associated** with the input element (e.g., via a correct <label for="...">).
• **Screen readers** must be able to read the associated label correctly.
• Information about input formats (e.g., date), required fields, or special requirements must be **present and understandable**.
• **Placeholder text must not be used as the sole label**.
• Vague or redundant terms like “Input,” “Data,” or “Field” should be **avoided**.

Examples of Good Practice:

Requirement	Good Practice Example
Email input	<label for="email">Email Address *</label> + hint: “We need your email to contact you.”
Date input	Label “Date of Birth” + help text “Format: MM/DD/YYYY”

Requirement	Good Practice Example
Required field indication	Asterisk () *with introductory text: “* Required field” at the top of the form
Error message	Error appears directly below the relevant field and is programmatically linked
Code association	<label for="lastname">Last Name</label> + <input id="lastname" type="text">

Common Failures:

Faulty Implementation	Why It’s Problematic
Field without a label or unclear label (“Input”)	Users don’t know what information is expected
Label not connected via for attribute	Screen readers cannot detect the label; label is not read aloud
Placeholder used as only label	Text disappears when typing — users lose orientation
No explanation for required format	Users are unable to enter data correctly (e.g., for birth date)

Summary:
WCAG Success Criterion 3.3.2 ensures that **every form element is clearly, unambiguously, and accessibly labeled**. Labels must be visible, understandable, and properly linked to their corresponding input fields.

Additionally, users must be informed about **required fields**, **input format expectations**, and potential **error messages**. This ensures that form usage is accessible and understandable for all user groups.

4.12.2 Accessible Error Identification and Helpful Feedback

Success Criteria and Evaluation Guidance
• **WCAG Success Criterion 3.3.1 – Error Identification (Level A)**
Forms must inform users when input errors occur and clearly indicate which field is affected.

• **WCAG Success Criterion 3.3.3 – Error Suggestion (Level AA)**
If an input error is detected and correction is possible, helpful suggestions should be provided.

Description

Accessible error identification means that users receive clear, understandable feedback when errors occur in form input — allowing them to correct mistakes independently, regardless of whether they use a mouse, keyboard, or assistive technologies such as screen readers.

Key Requirements:

1. **Errors are automatically detected**
   – For example, when required fields are left blank, or invalid formats or values are entered.
2. **Error messages are visible and accessible**
   – Messages should appear near the affected field or be listed at the top of the page (for complex forms), be visually and textually emphasized, and programmatically associated with the field (e.g., using aria-describedby).
3. **Error messages are clear and include correction hints**
   – Vague messages like “Error” or “Invalid input” are insufficient. Messages must explain what went wrong and how to fix it — e.g., “Please enter a date in the format MM/DD/YYYY.”
4. **Errors are not indicated by color alone**
   – Visual cues such as a red outline must be supplemented with text or symbols to ensure accessibility for users with color vision deficiencies.
5. **Focus handling is meaningful**
   – After submitting a form with errors, focus must move to the error message or the affected field so keyboard users can respond quickly.

Typical Evaluation Process

Step	**Procedure**
1. Submit form with errors	Intentionally leave fields blank or enter incorrect data; check for error detection.
2. Visual inspection of messages	Are the messages clearly visible? Do they include text, not just color?
3. Screen reader test	Navigate the form using a screen reader — are errors read aloud? Are they associated with the correct fields?
4. Code inspection	Check whether attributes like aria-describedby are correctly used.
5. Input guidance check	Are instructions provided before input (e.g., required fields, formats)?
6. Focus behavior test	Does focus remain on the error message or move to the field after submission?

Step	Procedure
7. Accessibility of correction	Can the form be corrected using only keyboard and screen reader?

Examples

Situation	Implementation	Evaluation
Required field left blank	"Please fill out this field." — shown below the input, with red warning symbol, linked via aria-describedby	Accessible
Invalid email format	"Please enter a valid email address in the format name@domain.com."	Clear & helpful
Only red border, no text	Screen reader cannot detect error; unclear for all other users	Not accessible
Error list at top of form	"3 errors found — click to jump to the fields."	Useful for long forms

Summary

Accessible error detection and guidance in forms is essential for inclusive design. The following must be ensured:

- Automatic error detection
- Visible, screen reader-accessible error messages
- Correct association between message and input field
- Clear explanations and guidance for correction
- Error indication not based on color alone
- Helpful focus management toward error locations

This ensures that all users — regardless of impairments or assistive technologies — can use forms **completely, understandably, and without stress**.

4.13 Compatible

Key Terms: Compatibility, Status Messages, WAI-ARIA, Semantic Code

The term *"Compatible"* comes from **WCAG Principle 4: Robust**, and refers to the requirement that websites and web applications must be developed in a way that ensures reliable functionality and correct interpretation across different browsers, devices, and assistive technologies (e.g., screen readers, speech output, braille displays).

Core Requirements for Compatibility:

1. **Clean, valid source code**
   – HTML and CSS must be written according to established standards.
   – Syntax errors, incorrectly nested elements, or missing tags can negatively affect compatibility with screen readers and other assistive technologies.
2. **Use of standard HTML elements and ARIA attributes**
   – Controls should be built using native HTML elements (e.g., <button>, <label>).
   – Use ARIA roles and attributes only when necessary — never as a substitute for clean HTML.
3. **Proper use of IDs and associations**
   – Labels must be programmatically linked to input fields using unique IDs.
   – ARIA attributes such as aria-labelledby or aria-describedby must correctly reference valid elements.
4. **Avoid reliance on specific technologies**
   – Content and functionality must remain accessible even when certain scripts, plugins, or stylesheets are disabled.
5. **Testing across devices and assistive tools**
   – Functionality should be tested using different browsers, keyboard navigation, screen readers, and mobile devices.

Why Compatibility Matters:
• People with disabilities use a variety of assistive tools that depend on well-structured, standards-compliant content.
• Incompatible content can result in controls not being announced, forms becoming unusable, or navigation breaking entirely.
• Compatibility is also key for long-term stability, updates, and future accessibility improvements.

Summary:
In the context of WCAG, **compatible** means that websites must be technically implemented to ensure **flawless and reliable operation** across common browsers, devices, and assistive technologies — enabling all users to access content correctly and without barriers.

4.13.1 Programmatically Available Status Messages

Reference to Success Criterion and Evaluation Guidance

• WCAG Success Criterion 4.1.3 – Status Messages (Level AA)

Status messages must be programmatically available so they can be detected by assistive technologies and announced without requiring a change of focus.

Description

Status messages (e.g., loading indicators, success confirmations, or error alerts) inform users about changes in system state or provide feedback after actions. To ensure that blind or visually impaired users using screen readers or other assistive tools are also aware of these messages, they must be made programmatically available.

Key Requirements:

1. **Use ARIA live regions**
   Example:
   <div aria-live="polite">…</div>
   For urgent alerts (e.g., error messages), aria-live="assertive" may be used.
2. **Apply appropriate ARIA roles**
   To define the nature of the message, the following roles are recommended:
   • role="status" for general feedback or success confirmations
   • role="alert" for error messages or warnings
   These roles trigger automatic screen reader announcements without requiring focus shifts.
3. **Ensure visual clarity**
   Messages must be clearly visible through text, color contrast, or icons.
4. **Avoid automatic dismissal**
   Status messages should remain visible long enough or be dismissible by the user, not removed automatically too quickly.
5. **Do not rely on color alone**
   Color coding must be supported by accompanying text or symbols to be perceivable by users with color vision deficiencies.
6. **Logical placement in layout**
   Messages should appear near the action that triggered them, or prominently within the main content area.
7. **Manage focus appropriately**
   For important messages, visual cues or optional focus indication may help, but forced focus shifts should be avoided.

Examples of Good Implementation

Scenario	Implementation Example
Error message in form	<div role="alert">Please enter your email address.</div>
Success message after submit	<div role="status">Your message has been sent successfully.</div>
Dynamic loading notification	<div aria-live="polite">Loading data…</div>

Typical Evaluation Process

Step	Procedure
Trigger status message	For example, submit a form, apply a filter, update content
Visual inspection	Is the message clearly visible (text, color, position, icon)?
Screen reader test	Is the status message read aloud automatically and correctly?
Code inspection	Are aria-live, role="status", or role="alert" properly implemented?
Observe focus behavior	Does focus remain where it was, or is it shifted unexpectedly?

Common Failures

• Status messages are visible but not programmatically detectable
• Missing aria-live, role="status", or role="alert"
• Messages disappear too quickly (e.g., after 2 seconds)
• Messages appear outside the visible viewport
• Screen readers do not announce the message after user actions

Summary

WCAG Success Criterion 4.1.3 requires that status messages be accessible and understandable for users of assistive technologies — without additional interaction. This is achieved through:

• The use of aria-live regions
• Appropriate roles like status or alert
• A combination of visual presentation and technical accessibility
• Avoiding color-only indicators
• Ensuring messages are not automatically dismissed too quickly

This ensures that feedback such as errors, confirmations, or loading states is **accessible to all users**, regardless of the tools or methods they use to interact with digital content.

4.13.2 Semantics Through HTML Structural Elements and WAI-ARIA

Reference to Success Criterion and Evaluation Guidance

WCAG Success Criterion 1.3.1 – Info and Relationships (Level A)

Content, structure, and relationships must be conveyed using semantic HTML elements or programmatic means.

Description

In web accessibility, *semantics* means that content is marked up using appropriate HTML structural elements or ARIA roles so that its function and meaning are understandable to assistive technologies such as screen readers. This is particularly important for logical structure, navigation, and correct role assignment.

1. Semantics Through Native HTML Structural Elements

The basic rule: HTML offers many elements that already carry semantic meaning — these should be used by default. Examples include:

HTML Element	Meaning
`<header>`	Page or section header
`<nav>`	Navigation block
`<main>`	Main content area
`<section>`	Standalone section of content
`<article>`	Independent article or post
`<aside>`	Complementary content
`<footer>`	Page or section footer
`<h1>–<h6>`	Headings in logical hierarchy
`<ul>, <ol>, <li>`	Lists
`<table>, <th>, <td>`	Table structure

These elements help screen readers and other technologies interpret the page structure and the relevance of content correctly.

2. Supplementary Semantics with WAI-ARIA

When complex interactive content or custom components lack an HTML equivalent, WAI-ARIA (Accessible Rich Internet Applications) can be used in addition:

ARIA Role / Attribute	Purpose
role="navigation"	Identify navigation areas
role="main"	Define the main content (if <main> is not used)
role="complementary"	Indicate complementary content, such as <aside>
aria-label / aria-labelledby	Provide accessible labeling for elements
role="region" + aria-label	Identify regions with descriptive labels

Important: ARIA must only be used as a *supplement* — never as a replacement for semantic HTML.

3. The Role of Syntax

Error-free and logically nested HTML is essential for assistive technologies to interpret semantics correctly:

• Misplaced elements like <li> outside a list, or missing id references in aria-labelledby, make content inaccessible.

• Valid HTML improves compatibility with browsers and reliability across assistive tools.

Examples

Use Case	Implementation	Evaluation
Main navigation	<nav aria-label="Main navigation">…</nav>	Correct
Article structure	<article> … <section>…</section></article>	Semantically sound
Incorrect use	<div class="main">…</div> instead of <main>	Inadequate semantics
Unnecessary ARIA	<div role="navigation"> instead of <nav>	Redundant use of ARIA

Tools for Reviewing HTML Semantics and ARIA Roles

1. **WAVE (WebAIM)**
   Analyzes HTML structure, heading hierarchy, and ARIA usage. Marks errors directly on the page.
   [Online: wave.webaim.org](https://wave.webaim.org)
2. **axe DevTools (Deque)**
   Browser extension (Chrome, Firefox) for detecting semantic and ARIA violations based on WCAG.
   [Download: deque.com/axe/devtools](https://deque.com/axe/devtools)

3. **Google Lighthouse**
   Integrated in Chrome DevTools. Checks semantic structure, landmarks, and ARIA usage.
   Access via: Right-click > Inspect > “Lighthouse” tab
4. **W3C HTML Validator**
   Validates HTML syntax and detects structural or semantic issues.
   Online: validator.w3.org
5. **Accessibility Insights (Microsoft)**
   Chrome/Edge extension to analyze semantics, landmarks, ARIA, and keyboard paths.
   Download: accessibilityinsights.io
6. **HeadingsMap**
   Browser extension for Chrome and Firefox to visualize heading hierarchy — helpful for checking structural logic.

Note on Tool Use

Automated tools detect many typical issues such as missing semantic elements, improper or redundant ARIA roles, and incorrect heading structures. However, **they cannot detect everything** — especially whether roles or landmarks are used meaningfully. Therefore, **manual review** (e.g., via source code analysis and screen reader testing) remains essential.

Summary

The evaluation of semantics through HTML structural elements and WAI-ARIA ensures that content is **structured, understandable, and programmatically accessible** — a foundation for effective assistive technology support.

Requirements for semantic accessibility include:
• Using HTML elements according to their intended meaning
• Adding ARIA roles only where necessary and correct
• Maintaining valid syntax and logical structure
• Avoiding non-semantic tags like <div> or <span> when better alternatives exist

This guarantees that content is accessible, well-structured, and comprehensible for all user groups.

5 Strategies for Successfully Implementing Accessibility

⏱ Estimated Instruction Time: approx. 120 minutes

🎯 Learning Objectives of the Chapter

LO 5.1.1 – Explain the importance of accessibility testing for the successful implementation of digital accessibility. – K2
LO 5.2.1 – Recall strategies for integrating accessibility into development processes. – K1
LO 5.2.2 – Explain the concept of “shift-left” in the context of digital accessibility. – K2
LO 5.2.3 – Explain the need for continuous accessibility monitoring after release. – K2
LO 5.2.4 – Identify the benefits of early integration of accessibility in the development process. – K1
LO 5.3.1 – Explain the importance of structured test preparation for successful accessibility testing. – K2
LO 5.3.2 – Recall the key steps in preparing for an accessibility test. – K1
LO 5.3.3 – Define test objectives and scope for an accessibility evaluation. – K3
LO 5.3.4 – List criteria for selecting appropriate testing methods, tools, and assistive technologies for accessibility testing. – K1
LO 5.3.5 – Explain the importance of defining success criteria and test cases for accessibility test execution. – K2
LO 5.3.6 – Prepare a structure for documenting and reporting accessibility test results. – K3
LO 5.3.7 – Identify the challenge that not all pages of a website can be tested. – K1
LO 5.3.8 – Recall the criteria for selecting representative pages for an accessibility evaluation. – K1
LO 5.3.9 – List important page types for accessibility testing and justify their relevance. – K3

Systematic preparation for accessibility testing is a key component of an inclusive development process. This section introduces strategies for meaningfully and proactively integrating accessibility into digital projects to ensure long-term quality and accessibility.

The objective is to develop an understanding of the importance of regular testing and ongoing monitoring, and to become familiar with specific steps to consider when planning an accessibility test. These steps include selecting appropriate testing methods, defining a clear testing strategy, involving relevant stakeholders, and choosing suitable testing environments and representative pages of a website.

Early consideration of accessibility during the development process not only improves overall quality but also supports the efficient fulfillment of legal and normative requirements.

.

5.1 The Importance of Accessibility Testing

Key Terms: Accessibility Testing, Test Types, Evaluation Methods

Summary of the Importance of Accessibility Testing

1. **Ensure usability for all people**
   Accessibility tests verify whether digital content can be used regardless of disabilities or assistive technologies—for example, for people with visual, auditory, cognitive, or motor impairments.
2. **Meet legal requirements**
   In Germany, accessible websites are mandatory for public sector entities (BITV 2.0), and from June 2025, also for certain private providers (European Accessibility Act). Regular testing is necessary to ensure compliance.
3. **Improve usability for everyone**
   Accessible design also benefits users without disabilities—through clear navigation, visible focus indicators, understandable error messages, and better mobile usability.
4. **Prevent exclusion**
   A lack of accessibility excludes people from using digital services. Testing helps identify and remove unintended barriers early on.
5. **Ensure quality**
   Accessibility testing demonstrates professional quality. It uncovers structural, technical, or design-related flaws and encourages thoughtful, inclusive solutions.
6. **Reduce long-term costs**
   Early testing prevents costly fixes later. It supports sustainable development and reduces effort during future updates.

Conclusion:
Accessibility testing is critical for inclusive, legally compliant, and user-friendly web development. It promotes quality, prevents exclusion, and ensures digital services are accessible to everyone.

Why Regular Accessibility Testing Is Necessary

Accessibility is an ongoing process
Websites and applications are constantly evolving—with new content, features, or design updates. Each change can unintentionally introduce new barriers (e.g., missing alt text, poor color contrast, unlabeled form fields). Only regular testing ensures continued accessibility.

Essential in agile development processes

In agile methods like Scrum or Kanban, new releases ("sprints") are created continuously. Accessibility must be considered from the start—as part of the “Definition of Done” for every task, not only at the end of a project.

Continuous monitoring: Automated and manual

Automated tests can be integrated into CI/CD pipelines and allow rapid detection of common accessibility issues during development. However, manual testing is essential—especially for keyboard navigation, logical focus order, clarity of link text, and screen reader usability. Each new version of a website or application should be explicitly tested for accessibility to catch and fix barriers early.

Tools and Strategies for Ongoing Monitoring

Tool / Strategy	**Function**
Browser Developer Tools (e.g., Chrome DevTools)	Check contrast ratios (partly manually), assess semantics via DOM inspector, trace keyboard focus
Bookmarklets / WAVE Add-on	Quick visual accessibility analysis in the browser (WAVE also works on local sites)
axe DevTools (extension)	Automatically detects WCAG violations, e.g., missing alt text, improper ARIA use, contrast issues
Google Lighthouse (in Chrome DevTools)	Automated analysis based on selected WCAG criteria (not complete); focuses on performance, SEO, and accessibility
Color Simulators (e.g., Color Oracle)	Simulate color vision deficiencies to assess visual perceptibility
Screen Reader Testing (e.g., NVDA, VoiceOver)	Manual testing of usability and clarity using assistive technologies

Risks of Not Monitoring Accessibility

If new content is published without accessibility testing, digital barriers may be introduced unintentionally. Users—especially people with disabilities—may encounter difficulties, leading to frustration and a loss of trust in the digital service. In addition, legal obligations (e.g., BITV or the German Accessibility Strengthening Act [BFSG]) may be violated. If issues are only discovered later, remediation is often significantly more complex and costly than early detection during development.

Summary:

Regular accessibility testing is essential—especially in continuous development. It must be firmly integrated into the development process. Automated tools assist with initial checks, but manual testing remains indispensable for real-world usability.

5.2 Strategies for Integrating Accessibility into Development Processes

Key Terms: Integration into development processes, accessibility strategy, responsibilities

This section presents strategies for embedding accessibility into the development of digital products from the very beginning. Participants will learn that accessibility must not be treated as an afterthought but as an integral part of the development process.

1. **Consider accessibility early ("Shift Left")**

- Accessibility should be addressed in the planning and conceptual phases—not just tested at the end of the project.
- Requirements can be described through user stories that reflect specific accessibility needs:
    - *Example:* "As a user relying on a screen reader, I want to be able to fully operate the contact form."

2. **Include accessibility in the Definition of Done**

- In agile projects, accessibility should be part of the acceptance criteria.
- A feature is only considered complete if, for example:
    - Keyboard operation works reliably
    - Contrast requirements are met
    - All images have alternative text
    - Semantic HTML structure is correctly implemented

3. **Incorporate checklists into development and review processes**

- Accessibility checklists should be used during code reviews:
    - Are interactive elements properly labeled?
    - Do forms work with both screen readers and keyboard?
    - Do visible labels match accessible names?
    - Are status messages provided using aria-live?

Sample checklist items:

- Semantic HTML elements are used correctly
- ARIA is only used as a supplement, not a replacement for HTML
- Links and buttons are clearly labeled
- Form inputs are labeled and error messages are accessible

4. **Integrate automated accessibility testing into CI/CD**

- Tools like axe-core, Pa11y, or Lighthouse can run automatically in each build
- These tools check for syntax, color contrast, alt text, landmarks, and basic ARIA issues

5. **Conduct regular manual testing**

- Automated tools do not detect all issues—manual testing is essential in every sprint:
    - Test keyboard navigation and logical focus order
    - Check compatibility with screen readers
    - Ensure usability on mobile devices
- Responsibility for testing should be clearly assigned within the team

6. **Provide documentation and guidelines**

- Developers and designers should have access to practical resources:
    - Best practices for accessible HTML and ARIA
    - Templates and checklists for forms, link texts, and visual design

7. **Promote training and awareness within the team**

- Offer regular training for all team roles (e.g., developers, designers, product owners)
- Sharing experiences about common mistakes and successful implementations supports accessibility competence in daily work

Conclusion:
Accessibility can only be implemented sustainably when it is embedded into the development process. This includes clear requirements, automated and manual tests, documented standards, and trained teams. The result is digital products that are accessible to all people—from the start.

Benefits of Early Integration of Accessibility into Development Processes

Early integration of accessibility offers many benefits: it reduces long-term costs, improves the quality of digital products, and ensures access for all. Organizations benefit not only technically, but also legally, economically, and socially.

1. **Reduced rework and lower costs**
   When accessibility is considered from the start, problems can be identified and resolved early—before they become expensive. Retrofitting after a product is complete is largely avoided.
2. **Avoidance of design flaws and rework**
   Structural barriers that would require a complete redesign never occur. This saves development time and reduces the risk of having to rethink core functionality.
3. **Improved code quality and clean structure**
   Accessible code is based on semantically correct HTML and a clear structure. This makes maintenance easier, increases stability, and improves team collaboration.
4. **Better usability for everyone**
   What is accessible is usually also user-friendly. Clear navigation, understandable content, and well-visible controls benefit all users—regardless of disability.
5. **Legal compliance without last-minute pressure**
   Early implementation allows organizations to meet legal requirements like the German Accessibility Strengthening Act (BFSG) or the Ordinance on Accessible Information Technology (BITV) on time—without rushing at the end of a project.
6. **Sustainable and future-proof development**
   Accessible solutions are more resilient to technological change. They work reliably across browsers, on mobile devices, and with assistive technologies.
7. **Wider reach through inclusive design**
   Accessible digital products reach a broader audience—including older users, people with disabilities, and those with low digital literacy—thereby increasing usage rates.
8. **Positive image and social responsibility**
   Organizations that proactively implement accessibility demonstrate responsibility, promote inclusion, and position themselves as modern, forward-thinking, and user-focused.

Conclusion:
Early integration of accessibility is an investment in quality, sustainability, and inclusion. It creates value for everyone—technically, economically, and socially.

5.3 Preparing for an Accessibility Test

Key Terms: Test preparation, page selection, evaluation criteria

This section focuses on the preparation phase of an accessibility test. It covers the systematic planning and organization of all necessary steps to effectively evaluate the accessibility of digital content and applications for people with various impairments. This includes selecting appropriate testing methods, tools, and assistive technologies, defining test objectives, and incorporating relevant standards and guidelines such as WCAG or EN 301 549. Thorough preparation ensures that the test can be conducted in a structured, efficient, and meaningful way.

5.3.1 Steps for Preparing an Accessibility Test

A systematically prepared accessibility test is essential for generating meaningful results and implementing targeted improvements. The following steps support a well-structured and efficient testing process:

1. **Define test objectives and scope**
   Identify which pages, features, and content should be tested—for example: homepage, main navigation, forms, media content, or specific applications.
2. **Establish accessibility requirements**
   Determine which standards and levels of conformance will be evaluated:
   – e.g., WCAG 2.1 or 2.2
   – Level A, AA, or AAA
   – optionally, requirements from the BITV or other relevant regulations
3. **Assemble the test team and assign roles**
   Form an interdisciplinary team (e.g., developers, QA, designers, content authors). Assign responsibilities for:
   • Automated testing
   • Manual keyboard testing
   • Screen reader testing
   • Documentation and evaluation
4. **Set up the test environment**
   Ensure that all necessary technical conditions are met:
   • Up-to-date browsers (e.g., Chrome, Firefox, Edge)
   • Test devices (PC, tablet, smartphone)
   • Installed assistive technologies (e.g., NVDA, VoiceOver, TalkBack)

5. **Select and configure testing tools**
   Install and configure automated testing tools such as:
   • axe DevTools
   • WAVE
   • Google Lighthouse
   • Pa11y
   Optionally: Integrate these tools into your continuous integration (CI/CD) process.
6. **Define manual test scenarios**
   Plan targeted tests for:
   • Keyboard accessibility
   • Visible and logical focus management
   • Screen reader compatibility
   • Form input and error messages
   • Modal dialogs, sliders, and other interactive components
7. **Provide checklists**
   Use structured checklists for reviewing:
   • Form elements
   • Alternative text for images
   • Color contrast
   • Focus indicators
   • Heading structure
   • Link text
   • ARIA usage
8. **Prepare test data**
   Provide realistic example content, user data, and typical input values to enable practical testing scenarios.
9. **Create a structure for test documentation**
   Prepare a consistent format for documenting results, including:
   • Tested criterion
   • Evaluation (compliant / non-compliant)
   • Screenshot or code sample
   • Recommendation for improvement
   Ensure traceable versioning and proper archiving of test results.

Conclusion:
Good preparation makes accessibility testing more efficient and meaningful. With clearly defined objectives, appropriate tools, realistic testing scenarios, and structured documentation, all team members can actively contribute to improving digital accessibility.

5.3.2 Selecting Individual Pages of a Website for Accessibility Testing

Key Terms: Page selection, website structure, test coverage

In practice, it is rarely feasible to test every subpage of a website in full. Therefore, it is essential to strategically select pages that represent typical functions, content, and structural elements. A well-considered selection ensures that all relevant accessibility aspects are adequately covered.

1. **Homepage**
   The homepage is often the first point of contact for users.
   It typically includes key elements such as navigation, teasers, sliders, or announcements—making it ideal for testing structure, keyboard operability, and visual design.
2. **Main navigation or category page**
   These pages represent the internal structure and navigation of a website, with numerous links and menu components.
   They are critical for testing focus order, keyboard accessibility, and the clarity of link text.
3. **Example page with standard content (text and media)**
   A typical content page with text, headings, images, videos, or links should always be included.
   This allows for testing of semantic structure, presence of alternative text, sufficient contrast, and logical reading order.
4. **Form page**
   At least one page with a form—such as for contact, registration, or search—must be included in the test scope.
   Key aspects to evaluate: correct labeling, understandable error messages, keyboard operability, and status message accessibility.
5. **Page with complex widgets**
   If pages include interactive components like dropdown menus, accordions, sliders, or modal dialogs, these must be tested specifically.
   Important criteria include correct ARIA implementation, logical focus behavior, and compatibility with assistive technologies.
6. **Download or media library page**
   These pages often contain PDF downloads, embedded videos, or audio content.
   Accessibility testing should include checking for accessible alternatives: accessible documents, transcripts, captions, or audio descriptions.
7. **Page with language switch or foreign-language content**
   If foreign languages are used, verify that language changes are correctly marked using the lang attribute—both for entire sections and individual words.

8. **Error pages or search results pages**
   Pages like "404 – Page not found" or search result pages provide essential user feedback. Check whether the messages are clearly worded, navigation can be continued, and focus is managed appropriately.

Summary:
To ensure a meaningful accessibility test, page selection should include all functional and structural types of a website:
Homepage, navigation, content pages, forms, interactive components, media content, and pages with special considerations such as language switching or error messaging.
Only then can the full range of potential accessibility barriers be identified and evaluated effectively.

6 Automated Evaluation Tools in Accessibility Testing

⏱ **Estimated Instruction Time:** approx. 30 minutes

🎯 **Learning Objectives of the Chapter**

- LO 6.1.1 – Identify the advantages of using automated evaluation tools in accessibility testing. – K1
- LO 6.2.1 – Explain why automated evaluation tools cannot fully replace manual accessibility testing. – K2
- LO 6.3.1 – Describe the functional strengths of automated evaluation tools in improving the efficiency of accessibility testing. – K1
- LO 6.4.1 – Recognize the limitations of automated evaluation methods and assess their impact on comprehensive accessibility evaluation. – K2
- LO 6.5.1 – Compare the functionality and use cases of selected evaluation tools and select appropriate tools for specific accessibility testing needs. – K3

Automated evaluation tools are a key resource in digital accessibility testing. They enable developers to quickly check websites for common errors, thereby improving quality and compliance with standards such as the Web Content Accessibility Guidelines (WCAG) and relevant legal frameworks. While these tools are highly useful, they also have limitations. Therefore, their use should always be understood as part of a comprehensive evaluation process.

6.1 Use Cases for Automated Tools

Key Terms: Automated Tools, Use Cases

Automated tools are used at various stages of the development process – from early design to technical implementation and quality assurance. They primarily assess technical aspects of accessibility, such as contrast ratios, alternative text, semantically correct HTML structures, and the use of ARIA attributes. Many tools can be integrated into development environments, build processes, or content management systems.

6.2 Role in the Evaluation Process According to WCAG and Legal Standards in the U.S. and U.K

Key Terms: Evaluation Process, WCAG, Legal Standards, Test Integration

The Web Content Accessibility Guidelines (WCAG 2.2) define three conformance levels: A, AA, and AAA. Most automated tools focus on Level A and AA criteria, such as alternative text (1.1.1), structure (1.3.1), contrast (1.4.3), and element roles (4.1.2).

In the U.S., WCAG 2.1 AA is the standard for Section 508 and ADA compliance. In the U.K., public services must meet WCAG 2.1 AA under accessibility regulations.

Automated accessibility tools are recommended to support conformance checks, especially for quickly identifying frequent technical issues. However, they are never considered a substitute for manual testing. Comprehensive accessibility evaluations must include manual reviews—particularly for aspects like keyboard navigation, meaningful link text, or screen reader usability—to ensure full compliance and user-centered accessibility.

6.3 Benefits of Automated Evaluation Tools

Key terms: Benefits of automated tools, efficiency, repeatability, scalability, traceability

Automated evaluation tools are a key support mechanism in assessing digital accessibility. They allow for fast, objective, and repeatable analysis of many technical requirements from the WCAG. Especially in early development phases, they help identify common issues efficiently and enable immediate corrections in code or interface. While they cannot replace manual testing, they offer high scalability and provide valuable input for ongoing improvement.

- **Time savings:** Common accessibility issues—such as missing alternative text, insufficient contrast, or absent ARIA attributes—can be detected in seconds and addressed immediately. This significantly speeds up the development process.
- **Integration:** These tools can be seamlessly embedded into existing build, test, and deployment pipelines. This enables continuous accessibility checks during development, identifying issues early (e.g., in CI/CD workflows).
- **Visualization:** Detected issues are shown directly within the user interface, often through browser extensions or developer tools. This supports clear understanding of problem areas and facilitates targeted corrections.
- **Repeatability:** Automated tests are consistent and yield the same results under the same configuration. This makes accessibility evaluations reliable and repeatable—regardless of who runs them.
- **Training value:** Many tools provide built-in explanations and suggestions for fixing problems. Developers, designers, and content editors receive immediate, practical feedback that promotes awareness and skills in accessibility.

- **Scalability:** Large websites or complex applications can be scanned efficiently—even across hundreds of pages. This allows wide-ranging weaknesses to be uncovered without manual review of each page.
- **Traceability:** Most tools generate automatic reports (e.g., in JSON or CSV formats), supporting project tracking, documentation, and communication with stakeholders or compliance bodies.

6.4 Risks and Limitations

Key terms: risks of automated testing, limitations, misinterpretation

Despite their many benefits, automated tools cover only a portion of accessibility criteria—typically between 20% and 40%, depending on the tool. They cannot, for example, assess meaningful focus order, detect language-related barriers, or evaluate the quality of alternative text content.

Another risk lies in the false sense of security: a green result in an automated tool does not guarantee full accessibility. Particularly complex or dynamic web applications require additional manual testing by qualified professionals or users of assistive technologies.

6.5 Selected Evaluation Tools: Features and Special Characteristics

Key terms: ANDI, Axe DevTools, WAVE, Lighthouse, Siteimprove, Tingtun Checker, Accessibility Insights, NoCoffee Vision Simulator

ANDI (Accessible Name & Description Inspector – SSA)

A lightweight, browser-based accessibility inspection tool developed by the U.S. Social Security Administration. ANDI runs as a bookmarklet and works directly in the browser without installation. It highlights one element at a time and displays its accessible name, role, state, and keyboard focusability, showing how it would be interpreted by assistive technologies. ANDI includes specialized modules for links, images, forms, tables, and headings, making it useful for manual inspection during WCAG conformance evaluations.
It is a recommended tool in Trusted Tester training and certification. It is often integrated into Section 508 compliance (US) workflows, especially for web-based public services. Some state governments and contractors also use ANDI when working on federally funded digital projects.

Limitation: Does not perform full-page automated scans or generate downloadable reports. It is not intended for large-scale automated testing, which limits its appeal in commercial environments or CI/CD pipelines. It is designed to support manual testing and is especially suited for Trusted Tester (US) workflows rather than large-scale automated testing.

Axe DevTools (Deque)

A widely used open-source tool with a commercial extension. It can be integrated as a browser extension in Chrome, Edge, or Firefox. Axe automatically checks a wide range of WCAG success criteria, including alternative text, ARIA implementation, color contrast, and semantic HTML structure. The commercial version offers enhanced reporting features, team collaboration tools, and integration into CI/CD pipelines (e.g., GitHub, Jenkins).
Limitation: Cannot assess visual or linguistic aspects, such as layout comprehension or clarity of instructions.

WAVE (WebAIM)

A user-friendly tool available as a web interface and browser add-on. It visually highlights accessibility issues and explains them clearly, making it ideal for editors and content creators checking for accessible content structure and contrast.
Limitation: Does not detect dynamic content changes or elements inside Shadow DOMs. May become difficult to use on complex websites with many elements.

Lighthouse (Google)

Included in Chrome DevTools, this tool assesses not only accessibility but also performance, SEO, and best practices. It provides a numeric accessibility score and detailed suggestions for improvements.
Limitation: Detects only basic accessibility issues and lacks depth for comprehensive WCAG testing.

Siteimprove Accessibility Checker

A professional tool suited for large websites or organizations. The free browser extension evaluates individual pages, while the full version offers site-wide monitoring, detailed reports, policy management, and workflow integration with CMS or issue trackers.
Limitation: Limited functionality in the free version; full access requires a paid subscription.

Tingtun Checker

A European tool designed in alignment with public sector monitoring requirements under the EU Web Accessibility Directive. It systematically checks HTML structure, ARIA usage, and formal WCAG aspects based on defined rulesets.
Limitation: Technically oriented interface with minimal visual guidance; best suited for advanced users and developers.

Accessibility Insights (Microsoft)

Provides both automated and guided assessments. The “Assessment Mode” is especially valuable, enabling structured, step-by-step evaluation of WCAG success criteria with explanations and validation checks.
Limitation: Available only as an add-on for Chrome and Edge; cannot scan entire websites automatically.

NoCoffee Vision Simulator (Firefox Add-on)

A simulation tool that visually replicates various visual impairments such as color blindness, cataracts, central or peripheral vision loss. It helps teams understand how users with low vision perceive web content.
Limitation: Available only for Firefox; does not detect accessibility violations or test against WCAG—it is purely for visual simulation and empathy-building.

Pa11y

An open-source command-line tool for automated accessibility testing. Pa11y checks web pages against WCAG success criteria using a headless browser. It reports issues such as missing alternative text, insufficient color contrast, improper heading structure, and ARIA misuse. Pa11y can be integrated into development workflows or used with complementary tools like Pa11y Dashboard or Pa11y Webservice for continuous monitoring and visualization.

Limitation: Cannot evaluate visual or content-related aspects, such as the understandability of language, layout clarity, or user experience with assistive technologies.

Conclusion

Automated evaluation tools are an essential part of modern accessibility strategies. They streamline development, help identify systematic issues early, and raise awareness among developers and designers. However, manual testing, especially by trained evaluators and users with disabilities, remains irreplaceable. Only the combination of automated and manual methods enables a reliable, high-quality assessment of digital accessibility in line with WCAG, the Americans with Disabilities Act (ADA), and the European Accessibility Act.

.

7 Learning Objectives and Cognitive Learning Levels

Learning Objectives and Cognitive Learning Levels

The original Bloom's Taxonomy was developed in 1956 by Benjamin Bloom and his research team to systematically classify educational goals within the cognitive domain and to establish a shared language for defining learning objectives. In 2001, the taxonomy was comprehensively revised by a group led by Lorin W. Anderson (a former student of Bloom) and David R. Krathwohl. The revision aimed to align the structure more closely with modern learning and thinking processes. As part of the revision, the level names were transformed into action-oriented verbs, and the hierarchy was adjusted: the highest level is now "Create" instead of "Evaluate."

Today, the revised version is widely used in education, higher education didactics, and professional training.

The revised Bloom's Taxonomy classifies cognitive learning objectives into six hierarchical levels, referred to as K1 to K6. Each level represents a specific cognitive performance — from recalling simple facts (K1), to understanding and applying (K2 and K3), to analyzing (K4), evaluating (K5), and ultimately creating new concepts or products (K6). The revision by Anderson and Krathwohl emphasizes active learning processes and the practical application of knowledge in complex contexts.

Level	Name	Description	Typical Verbs	Example Tasks
K1	Remember	Recall and reproduce facts, terms, and information.	List, identify, recall, name	"List the four principles of the WCAG." "Name common barriers for screen reader users."
K2	Understand	Grasp, interpret, and explain meanings.	Explain, describe, summarize, illustrate	"Explain why contrast ratios are important in accessible design." "Describe the role of alternative text in digital accessibility."
K3	Apply	Use and implement knowledge in specific situations.	Apply, execute, demonstrate, use	"Apply WCAG criteria to make a form accessible." "Demonstrate how to optimize navigation for keyboard use."

Level	Name	Description	Typical Verbs	Example Tasks
K4	Analyze	Break down information, recognize structures and relationships.	Analyze, differentiate, examine, compare	“Analyze the accessibility issues on a given website.” “Compare WCAG requirements to those of the BITV.”
K5	Evaluate	Make judgments and critically assess arguments or methods.	Evaluate, justify, critique, verify	“Evaluate the accessibility of an online shop based on the WCAG guidelines.” “Critique the accessibility implementation in a PDF document.”
K6	Create	Develop new products, theories, or approaches.	Design, plan, develop, generate	“Plan an accessible navigation concept for a complex website.” “Design a training session to raise awareness for digital accessibility.”

8 Recommended Reading and Standards

TECHNICAL STANDARDS AND GUIDELINES

- **W3C: Web Content Accessibility Guidelines (WCAG) 2.2**
  W3C Recommendation, October 2023
  https://www.w3.org/TR/WCAG22
  Core specification for implementing digital accessibility.
- **WCAG-EM (W3C Evaluation Methodology)**
  https://www.w3.org/WAI/test-evaluate/conformance/wcag-em/
  W3C methodology for systematically evaluating websites against WCAG 2.1. Widely used in the EU for monitoring public sector accessibility under Directive 2016/2102.
- **W3C: WAI-ARIA Authoring Practices 1.2**
  https://www.w3.org/TR/wai-aria-practices/
  Practical guidance for accessible UI components and design patterns.
- **W3C: Accessible Rich Internet Applications (WAI-ARIA) 1.2**
  https://www.w3.org/TR/wai-aria-1.2/
  Specification for making dynamic content accessible to assistive technologies.
- **EN 301 549 V3.2.1 (2021-03)**
  https://www.etsi.org/deliver/etsi_en/301500_301599/301549
  European standard for accessibility requirements in public procurement of ICT products and services.
- **ISO 9241-171:2008 – Ergonomics of Human-System Interaction, Part 171**
  Guidance on software accessibility
  International standard offering best practices for accessible software design.
- **Section 508 Standards (U.S. Rehabilitation Act)**
  https://www.access-board.gov/ict/
  U.S. federal accessibility standards aligned with WCAG 2.0.
- **Accessibility Regulations 2018 (UK)**
  https://www.gov.uk/guidance/accessibility-requirements-for-public-sector-websites-and-apps
  UK law requiring public sector websites and apps to meet WCAG 2.1 AA.

TEXTBOOKS AND REFERENCE BOOKS

- **Shawn Lawton Henry (W3C): *Just Ask: Integrating Accessibility Throughout Design***
  https://www.uiaccess.com/accessucd/
  Free e-book for inclusive UX design practices.

- **Sarah Horton & Whitney Quesenbery: *A Web for Everyone: Designing Accessible User Experiences* (Rosenfeld Media)**
  Practical book combining usability, UX, and accessibility.
- **Heydon Pickering: *Inclusive Design Patterns* (Smashing Magazine)**
  Focused on ARIA, semantic HTML, and accessible component design.
- **Jonathan Lazar et al.: *Research Methods in Human-Computer Interaction***
  Includes chapters on accessibility testing and inclusive user research.
- **Lainey Feingold: *Structured Negotiation: A Winning Alternative to Lawsuits***
  Legal perspective on achieving accessibility without litigation.

PRACTICAL GUIDES AND STUDIES

- **WebAIM: Introduction to Web Accessibility**
  https://webaim.org/intro/
  Foundational overview and practical guide for educators and developers.
- **WebAIM Million Report (2024)**
  https://webaim.org/projects/million/
  Annual large-scale analysis of accessibility on the web.
- **W3C: WAI Tutorials**
  https://www.w3.org/WAI/tutorials/
  Hands-on tutorials for forms, images, navigation, tables, etc.
- **TeachAccess (U.S. initiative): Curricula and Teaching Materials**
  https://teachaccess.org
  Free curriculum resources for universities and tech training.
- **Deque University**
  https://dequeuniversity.com
  Industry-standard training platform for accessibility testing and development.
- **The A11Y Project**
  https://www.a11yproject.com
  Community-driven resource hub with checklists, tools, and code patterns.

SPECIAL TOPICS AND INTERNATIONAL PERSPECTIVES

- **G3ict (2023): CRPD ICT Accessibility Progress Report**
  https://g3ict.org
  Global study tracking digital inclusion in line with the UN CRPD.

- **UNESCO (2022): *Digital Inclusion of Persons with Disabilities***
  https://unesdoc.unesco.org
  Strategic paper addressing policy and education.
- **AbilityNet (UK): Accessibility Resources and Case Studies**
  https://abilitynet.org.uk
  Guidance and audits from a UK accessibility organization.
- **National Center on Accessible Educational Materials (AEM Center, U.S.)**
  https://aem.cast.org
  Focus on accessible content in education and publishing.