Manual on Digital Accessibility

Table of Contents

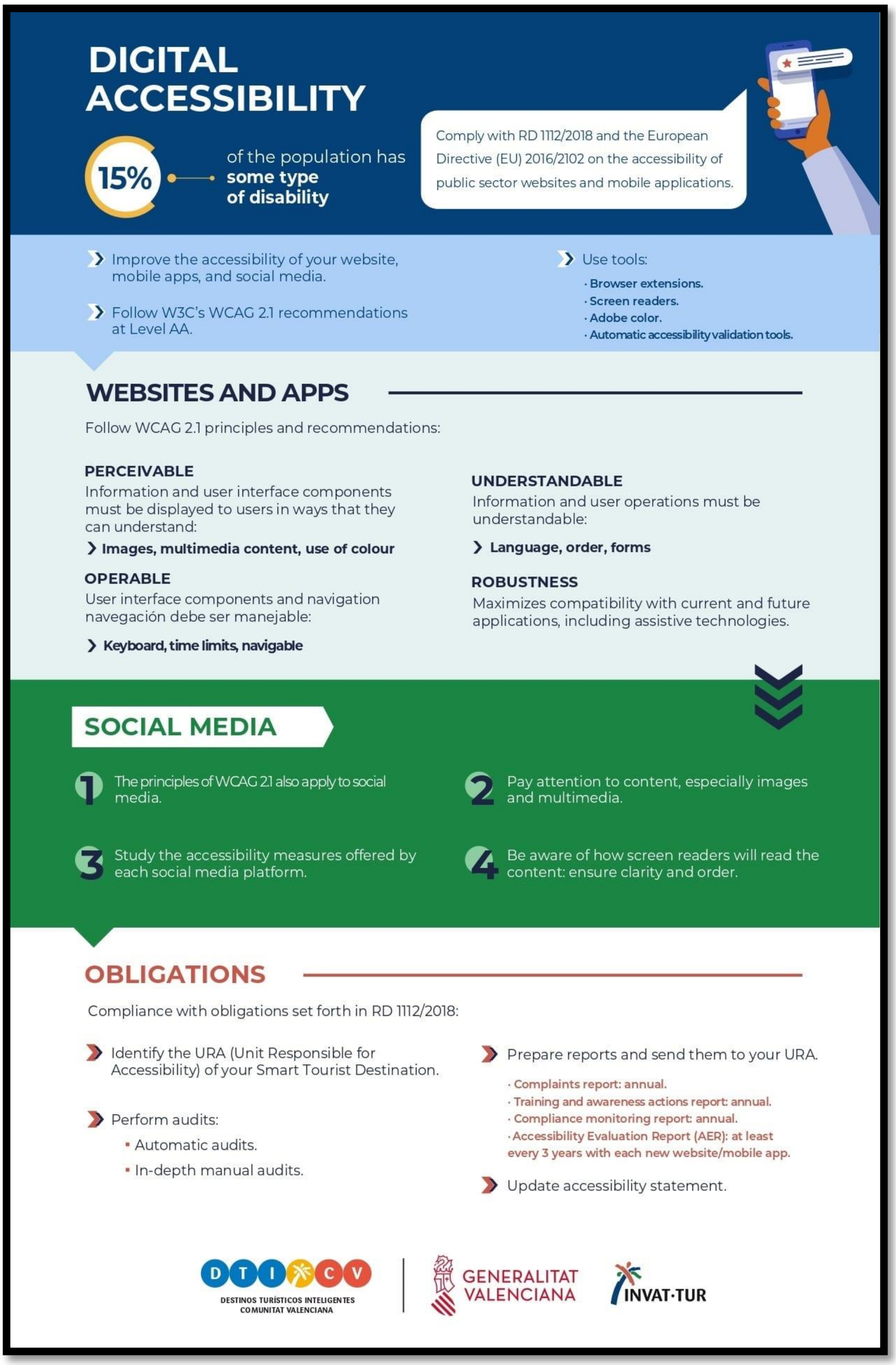
1. Introduction

This manual on digital accessibility provides valuable information and tools for technical managers to improve the digital accessibility of websites.

The manual focuses on ensuring compliance with accessibility regulations. Its aim is to support managers in meeting their accessibility obligations and responsibilities towards people with disabilities and creating inclusive digital experiences.

By following the guidelines presented in this manual, technical managers will gain insights into best practices for designing, developing, and maintaining accessible digital assets. It provides a holistic approach, covering various aspects of digital accessibility, including website structure, content, navigation, multimedia, forms, and more. Furthermore, the manual equips managers with the necessary knowledge to conduct accessibility audits, implement remediation strategies, and establish ongoing accessibility testing processes.

1.1. Executive Summary



DIGITAL ACCESSIBILITY

15% of the population has **some type of disability**

Comply with RD 1112/2018 and the European Directive (EU) 2016/2102 on the accessibility of public sector websites and mobile applications.

- Improve the accessibility of your website, mobile apps, and social media.
- Follow W3C's WCAG 2.1 recommendations at Level AA.
- Use tools:
  - · **Browser extensions.**
  - · **Screen readers.**
  - · **Adobe color.**
  - · **Automatic accessibility validation tools.**

WEBSITES AND APPS

Follow WCAG 2.1 principles and recommendations:

PERCEIVABLE
Information and user interface components must be displayed to users in ways that they can understand:

- **Images, multimedia content, use of colour**

OPERABLE
User interface components and navigation navegación debe ser manejable:

- **Keyboard, time limits, navigable**

UNDERSTANDABLE
Information and user operations must be understandable:

- **Language, order, forms**

ROBUSTNESS
Maximizes compatibility with current and future applications, including assistive technologies.

SOCIAL MEDIA

1. The principles of WCAG 2.1 also apply to social media.
2. Pay attention to content, especially images and multimedia.
3. Study the accessibility measures offered by each social media platform.
4. Be aware of how screen readers will read the content: ensure clarity and order.

OBLIGATIONS

Compliance with obligations set forth in RD 1112/2018:

- Identify the URA (Unit Responsible for Accessibility) of your Smart Tourist Destination.
- Perform audits:
  - Automatic audits.
  - In-depth manual audits.
- Prepare reports and send them to your URA.
  - · **Complaints report: annual.**
  - · **Training and awareness actions report: annual.**
  - · **Compliance monitoring report: annual.**
  - · **Accessibility Evaluation Report (AER): at least every 3 years with each new website/mobile app.**
- Update accessibility statement.

DTI CV
DESTINOS TURÍSTICOS INTELIGENTES
COMUNITAT VALENCIANA

GENERALITAT VALENCIANA

INVAT·TUR

Illustration 1. Executive summary of the Digital Accessibility Manual

1.2. Aim of this Guide

Have you ever thought about how many people may not understand a video posted by your organization on its Facebook page? Is a blind person able to navigate around your website? Has your mobile app been designed so that its texts can be understood by people with low vision?

People commonly believe that Internet access is universal. However, having an Internet connection and a computer or smartphone aren't the only things you need for access to web content: they are of little use if content is presented in a format that many people can't understand.

According to data from the World Health Organization (WHO), 15% of the world's population experience some form of disability. This means that every effort we make in improving accessibility will also improve the lives of many people, enabling them to enjoy their fundamental rights.

Disability comes in many forms and, therefore, people's needs may differ greatly. The idea is simple: everybody should be able to access web content easily.

It is important to first determine the types of disabilities that exist, and the user needs associated to each:

- Total or partial blindness: screen readers or braille displays are needed.
- Disabilities that affect vision or colour distinction: in this case, content needs to have adequate contrast and, more importantly, if this information is conveyed in colour, it is supported by a complementary method. For these users, being able to modify the size and contrast of the elements they see on the screen is vital.
- Deafness of varying degrees: in these cases, audio content needs to be accessible in the form of text (e.g., captions).
- Attention deficit disorders, hyperactivity, and dyslexia: layout and clarity of content and navigation elements in particular improve online experience for these users
- Motor disabilities (ranging from missing limbs to arthritic-type pain etc.): These users may not be able to operate a mouse or touch screens, and therefore require alternative means of input.
- Problems with tremors and precision in movement: as in the previous case, these individuals may present limitations in the use of the mouse or significant difficulties in using touch screens. The fundamental issue is that they lack the precise movements needed to use the controls properly.
- Users with intellectual disabilities: these individuals often find it difficult to understand the content and may need visual and written descriptions and assistance. Inconsistency of website or mobile app elements can cause confusion.

In addition, elderly people may experience two or more disabilities at the same time, and so we also need to consider the special needs of this group.

By understanding the above disability types and putting ourselves in the shoes of a person experiencing them, we will be taking an important step forward in the knowledge that not all people perceive the world as we do. This represents a good starting point, because in the future, whenever you generate a PDF, upload a photo to Instagram, or choose a colour for a text on your website, you will probably ask yourself: Will everyone be able to understand the message I'm trying to convey?

If you can adapt your work methods to this idea of reaching all users, you will have achieved your goal: that of improving digital accessibility.

It is also important to remember that digital accessibility not only benefits people with disabilities, but it also makes our communication channels much more usable for everyone.

This manual offers guidance and technical information on the design and implementation of strategies and good practices that will help you develop content that is more accessible and inclusive.

1.3. What is universal accessibility?

Broadly speaking, digital accessibility is a basic feature of digital environments, allowing universal access to web pages and mobile apps for all citizens, with special emphasis being placed on the needs of people with disabilities and the elderly.

The objective is to promote and raise awareness about the need to implement changes in websites and mobile apps that will ensure that people with disabilities can perceive, understand, navigate, and interact with these environments.

When designing, maintaining, or updating a website or app, you need to comply with a series of principles and techniques to ensure good web accessibility.

Yet we must never lose sight of the main objective: to ensure that everyone can access and use information under conditions of equality and can do so independently.

1.4. The WCAG 2.1 Accessibility Guidelines of the W3C

The W3C (World Wide Web Consortium) is an international community whose main objective is to implement uniform technologies in the use and development of the Internet.

One of the initiatives developed by W3C is its Web Accessibility Initiative (WAI), which aims to safeguard and work towards web accessibility. Out of this initiative came the Web Content Accessibility Guidelines (WCAG), a set of internationally recognized guidelines and technical standards to help create accessible web content. In addition to their use for websites, these principles also apply to social media, applications, and digital environments.

This guide is based on WCAG 2.1 which is currently in force, although these guidelines are updated frequently.

The guidelines are divided into four principles:

- **Perceivable:** Can the user perceive the content in question?
- **Operable**: Can the user navigate, input data, or otherwise interact with the content?
- **Understandable:** Can a user process and understand the displayed content?
- **Robust**: Is the content available as intended in a wide range of browsers, including legacy and emerging browsing environments?

Based on the above criteria, there are three levels of conformance that represent the degree to which requirements are met: basic (Level A), medium (Level AA or Double-A) and high (Level AAA or Triple-A). the most common situation is that websites and mobile apps must meet medium level accessibility requirements, that is, Level AA.

1.5. WCAG-EM evaluation methodology

This methodology is designed for anyone who wants to follow a common approach for evaluating the conformance of websites to WCAG 2.0, This is the internationally recognized standard explaining how to make web content more accessible to people with disabilities.

Terms and Definitions

For the purposes of this document, the following terms and definitions apply:

Complete processes

When a web page is one of a series of web pages presenting a process (i.e., a sequence of steps that need to be completed in order to accomplish an activity), all web pages in the process conform at the specified level or better. (Conformance is not possible at a particular level if any page in the process does not conform at that level or better.)

Conformance

Satisfying all the requirements of a given standard, guideline or specification

Common web pages

Web pages and *web page states* that are relevant to the entire website. This includes the homepage, login page, and other entry pages, and, where applicable, the sitemap, contacts page, site help, legal information, and similar web pages that are typically linked from all other web pages (usually from the header, footer, or navigation menu of a web page).

Essential functionality

Functionality of a website that, if removed, fundamentally changes the use or purpose of the website for users. This includes information that users of a website refer to and tasks that they carry out to perform this functionality.

Note: Examples of essential functionality include "selecting and purchasing a product from the shop area of the website", "completing and submitting a form provided on the website", and "registering for an account on the website".

Note: Other functionality is not excluded from the scope of evaluation. The term "essential functionality" is intended to help identify critical web pages and include them among others in an evaluation.

Evaluator

The person, team of people, organization, in-house department, or other entity responsible for carrying out the evaluation.

Evaluation commissioner

The person, team of people, organization, in-house department, or other entity that commissioned the evaluation.

Note: In many cases the evaluation commissioner may be the website owner or website developer, in other cases it may be another entity such as a procurer or an accessibility monitoring survey owner.

Relied upon (Technologies)

The content would not conform if that technology is turned off or is not supported

Templates

Content patterns that are filled in by authors or the authoring tool to produce web content for end users (e.g., document templates, content management templates, presentation themes). Often templates will pre-specify at least some authoring decisions.

Website

A coherent collection of one or more related web pages that together provide common use or functionality. It includes static web pages, dynamically generated web pages, and mobile websites and applications.

Note: The focus of this methodology is on full, self-enclosed websites. Websites may be composed of smaller sub-sites, each of which can be considered to be an individual website. For example, a website may include an online shop, an area for each department within the organization, a blog area, and other areas that may each be considered to be a website.

Website developer

The person, team of people, organization, in-house department, or other entity that is involved in the website development process including but not limited to content authors, designers, front-end developers, back-end programmers, quality assurance testers, and project managers.

Website owner

The person, team of people, organization, in-house department, or other entity that is responsible for the website.

Web page

A non-embedded resource obtained from a single URI using HTTP plus any other resources that are used in the rendering or intended to be rendered together with it by a user agent

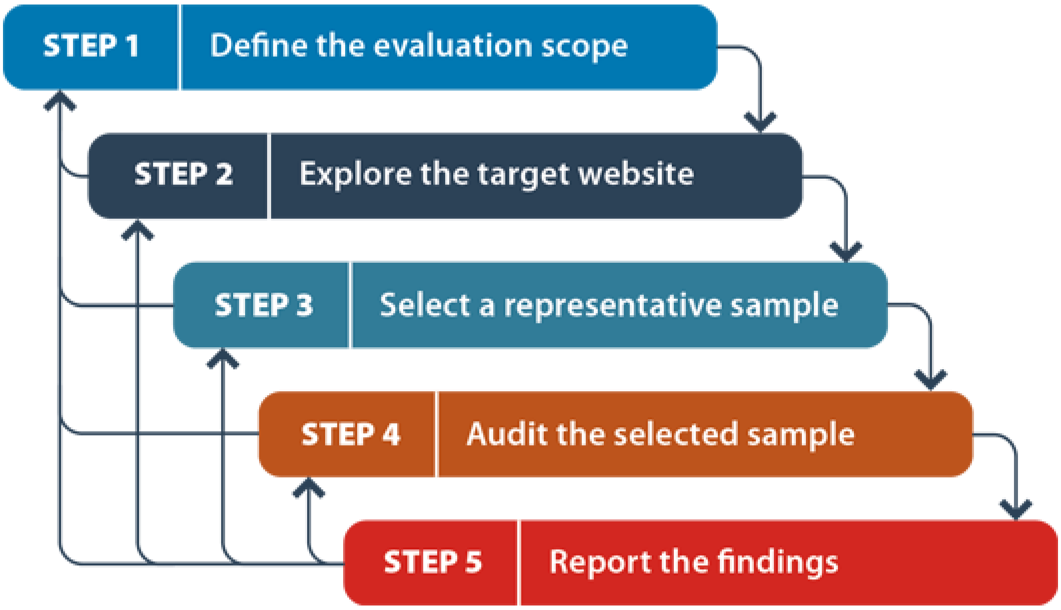
Note: Web pages may include multimedia content, interactive components, and rich and mobile web applications. Web pages are not limited to HTML and can be PDF documents and any other format.

Web page states

Dynamically generated web pages sometimes provide significantly different content, functionality, and appearance depending on the user, interaction, device, and other parameters. In the context of this methodology such web page states can be treated as ancillary to web pages (recorded as an additional state of a web page in a web page sample) or as individual web pages.

Note: Examples of web page states are the individual pages of a multi-part online form that are dynamically generated depending on the user's input. These individual states may not have unique URIs and may need to be identified by describing the settings, input, and actions required to generate them.

Next is a description of the stages and activities of an evaluation procedure. The stages are not necessarily sequential. Also, the exact sequence of the activities carried out during the evaluation stages depends on the type of website, the purpose of the evaluation, and the process used by the evaluator. Some of the activities can overlap or may be carried out in parallel. The following diagram illustrates the iterations between the stages defined in this section:


STEP 1
Define the evaluation scope
STEP 2
Explore the target website
STEP 3
Select a representative sample
STEP 4
Audit the selected sample
STEP 5
Report the findings

The workflow diagram above depicts five sequential steps: 1. Define the evaluation scope; 2. Explore the target website; 3. Select a representative sample; 4. Audit the selected sample and 5. Report the findings. Each step has an arrow to the next step, and arrows back to all prior steps. This illustrates how evaluators proceed from one step to the next and may return to any preceding step in the process as new information is revealed to them during the evaluation process.

- **Step 1: Define the Evaluation Scope**

Methodology Requirement 1: Define the evaluation scope according to Methodology Requirement 1.a, Methodology Requirement 1.b, and Methodology Requirement 1.c, and optionally Methodology Requirement 1.d.

During this step the overall scope of the evaluation is defined. It is a fundamental step that affects the subsequent steps in the evaluation procedure. It is ideally carried out in consultation with the evaluation commissioner (who may or may *not* be the website owner) to ensure common expectations about the scope of the evaluation. Initial exploration of the target website during this step may be necessary to better known specifics of the website and the required evaluation. Detailed exploration of the website is carried out in Step 2: Explore the Target Website.

Step 1.a: Define the Scope of the Website

Methodology Requirement 1.a: Define the target website according to Scope of Applicability, so that for each web page it is unambiguous whether it is within the scope of evaluation or not.

During this step the target website (the web pages and states of web pages that are in scope of the evaluation) is defined. This scope of the website is defined according to the terms established in the section Scope of Applicability.

To avoid later mismatches of expectations between the evaluator, evaluation commissioner, and readers of the resulting evaluation report, it is important to define the target website so that it is unambiguous that a web page is within its scope. Using formalizations including regular expressions and listings of web addresses (URIs) is recommended where possible.

It is also important to document any aspects of the target website to support its identification. This includes:

- Use of third-party content and services;

- Mobile and language versions of the website;

- Parts of the website, especially those that may not be easily identifiable as such, for example, an online shop that has a different web address but is still considered to be part of the target website.

Step 1.b: Define the Conformance Target

Methodology Requirement 1.b: Select a target WCAG 2.0 conformance level ("A", "AA", or "AAA") for the evaluation.

Part of initiating the evaluation process is to define the target WCAG 2.0 conformance level ("A", "AA", or "AAA") for evaluation. WCAG 2.0 Level AA is the generally accepted and recommended target.

Note: It is often useful to evaluate beyond the conformance target of the website to get a more complete picture of its accessibility performance. For example, while a website might not fully meet a particular conformance level, it might meet individual requirements from a higher conformance level. Having this information can help plan future improvements more effectively.

Step 1.c: Define an Accessibility Support Baseline

Methodology Requirement 1.c: Define the web browser, assistive technologies and other user agents for which features provided on the website are to be accessibility supported.

Particularly for new web technologies it is not always possible to ensure that every accessibility feature provided on a website, such as a 'show captions' function in a media player, is supported by every possible combination of operating system, web browser, assistive technology, and other user agents. WCAG 2.0 does not pre-define which combinations of features and technologies must be supported as this depends on the particular context of the website, including its language, the web technologies that are used to create the content, and the user agents currently available. Understanding Accessibility Support provides more guidance on the WCAG 2.0 concept of *accessibility support*.

During this step the evaluator determines the minimum set of combinations of operating systems, web browsers, assistive technologies, and other user agents that the website is expected to work with, and that is in-line with the WCAG 2.0 guidance on accessibility support (linked above). This step is carried out in consultation with the evaluation commissioner to ensure common expectation for the targeted level of accessibility support. The website owner and website developer may also have such a list of combinations that the website was designed to support, which could be a starting point for this step. Depending on the purpose of the evaluation such a list may need to be updated, for example to assess how well the website works with more current browsers.

Note: This initial definition of the baseline does not limit the evaluator from using additional operating systems, web browsers, assistive technologies and other user agents at a later point, for example to evaluate content that was not identified at this early stage of the evaluation process. In this case the baseline is extended with the additional tools that were used.

Note: For some websites in closed networks, such as an intranet website, where both the users and the computers used to access the website are known, this baseline may be limited to the operating systems, web browsers and assistive technologies used within this closed network. However, in most cases this baseline is ideally broader to cover most current user agents used by people with disabilities in any applicable particular geographic region and language community.

Step 1.d: Define Additional Evaluation Requirements (Optional)

Methodology Requirement 1.d: Define any additional evaluation requirements agreed by the evaluator and evaluation commissioner (Optional).

An evaluation commissioner may be interested in additional information beyond what is needed to evaluate the extent of conformance of the target website to WCAG 2.0. For example, an evaluation commissioner might be interested in:

- Evaluation of additional web pages beyond what is needed to form a representative sample from the target website;
- Reports of all occurrences of issues rather than representative examples of the types of issues on the target website;
- Analysis of particular use cases, situations, and user groups for interacting with the target website;
- Description of possible solutions to the issues encountered beyond the scope of the evaluation;
- Evaluation involving users with disabilities;
- Adherence to specific documentation or reporting templates.

Such additional evaluation requirements that are agreed on with the evaluator need to be clarified early on and documented. This also needs to be reflected in the resulting report, for example, to clarify how the selection of the sample was carried out.

Step 2: Explore the Target Website

Methodology Requirement 2: Explore the website to be evaluated according to Methodology Requirement 2.a, Methodology Requirement 2.b, Methodology Requirement 2.c, Methodology Requirement 2.d, and Methodology Requirement 2.e.

During this step the evaluator explores the target website to be evaluated, to develop an initial understanding of the website and its use, purpose, and functionality. Much of this will not be immediately apparent to evaluators, in particular to those from outside the development team. In some cases, it is also not possible to exhaustively identify and list all functionality, types of web pages, and technologies used to realize the website and its applications. The initial exploration carried out in this step is typically refined in the later steps Step 3: Select a Representative Sample and Step 4: Audit the Selected

Sample, as the evaluator learns more about the target website. Involvement of website owners and website developers can help evaluators make their explorations more effective.

Note: Carrying out initial cursory checks during this step helps identify web pages that are relevant for more detailed evaluation later on. For example, an evaluator may identify web pages that seem to be lacking color contrast, document structure, or consistent navigation, and note them down for more detailed evaluation later on.

Note: To carry out this step it is critical that the evaluator has access to all the relevant parts of the website. For example, it may be necessary to create accounts or otherwise provide access to restricted areas of a website that are part of the evaluation. Granting evaluators such access may require particular security and privacy precautions.

Step 2.a: Identify Common Web Pages of the Website

Methodology Requirement 2.a: Identify the common web pages, which may be web page states, of the target website.

Explore the target website to identify its common web pages, which may also be web page states in web applications. Typically, these are linked directly from the main entry point (home page) of the target website, and often linked from the header, navigation, and footer sections of other web pages. The outcome of this step is a list of all common web pages of the target website.

Step 2.b: Identify Essential Functionality of the Website

Methodology Requirement 2.b: Identify an initial list of essential functionalities of the target website.

Explore the target website to identify its essential functionality. While some functionality will be easy to identify, others will need more deliberate discovery. For example, it may be easier to identify the functionality for purchasing products in an online shop than the functionality provided for vendors to sell products through the shop. The outcome of this step is a list of functionalities that users can perform on the website. This list will be used in the following steps to help select representative web page instances for evaluation.

Note: The purpose of this step is not to exhaustively identify all functionality of a website but to determine those that are essential to the purpose and goal of the target website. This will inform later selection of web pages and their evaluation. Other functionality will also be included in the evaluation but through other selection mechanisms.

Examples of Website Functionality

Some examples of website functionality include:

- Selecting and purchasing products from the web shop;
- Completing and submitting the survey forms;
- Registering for an account on the website.

Step 2.c: Identify the Variety of Web Page Types

Methodology Requirement 2.c: Identify the *types* of web pages and web page states.

Web pages and web page states with varying styles, layouts, structures, and functionality often have varying support for accessibility. They are often generated by different templates and scripts, or authored by different people. They may appear differently, behave differently, and contain different content depending on the particular website user and context.

During this step the evaluator explores the target website to identify the different **types** of web pages and web page states. The outcome of this step is a list of descriptions of the types of content identified, rather than specific instances of web pages and web page states. This list will be used in the following steps to help select representative web page instances for evaluation.

Examples of Web Page Types

Some examples of different types of web pages and web page states that evaluators can look for include those:

- …with varying styles, layout, structure, navigation, interaction, and visual design;
- …with varying types of content such as forms, tables, lists, headings, multimedia, and scripting;
- …with varying functional components such as date picker, lightbox, slider, and others;
- …using varying technologies such as HTML, CSS, JavaScript, WAI-ARIA, PDF, etc.;
- …from varying areas of the website (home page, web shop, departments, etc.) including any applications;
- …with varying coding styles and created using varying templates (if this is known to the evaluator);
- …authored by varying people, departments, and other entities (if this is known to the evaluator);
- …that change appearance and behavior depending on the user, device, browser, context, and settings;
- …with dynamic content, error messages, dialog-boxes, pop-up windows, and other interaction.

Step 2.d: Identify Web Technologies Relied Upon

Methodology Requirement 2.d: Identify the web technologies relied upon to provide the website.

During this step, the web technologies relied upon for conformance are identified. This includes base web technologies such as HTML and CSS, auxiliary web technologies such as JavaScript and WAI-ARIA, as well as specific web technologies such as SMIL, SVG and PDF. The outcome of this step is a list of technologies that are relied upon according to WCAG 2.0. This list will be used in the following steps to help select representative web page instances for evaluation.

Note: Where possible, it is often also useful to identify any content management system, version, and configuration as it may be relevant to explain the evaluation results. Also, any libraries and components used to create the website, such as Dojo, jQuery, and others may be relevant. Particularly for web applications, much of the accessibility support is built into libraries and components, and evaluation can become more effective and efficient when these are identified.

Step 2.e: Identify Other Relevant Web Pages

Methodology Requirement 2.e: Identify other web pages and web page states that are relevant to people with disabilities and to accessibility of the website.

Some websites include web pages and web page states that are specifically relevant for people with disabilities and the accessibility of the website. The outcome of this step is a list of such web pages and web page states, if they have not already been identified as part of Step 2.a: Identify Common Web Pages of the Website.

Examples of Other Relevant Web Pages

Examples of other relevant web pages and web page states include those:

- …explaining the accessibility features of the website;
- …with information and help on the use of the website;
- …explaining settings, preferences, options, shortcuts, etc.;
- …with contact information, directions, and support instructions.

Step 3: Select a Representative Sample

Methodology Requirement 3: Select a representative sample of web pages from the website according to Methodology Requirement 3.a, Methodology Requirement 3.b, and Methodology Requirement 3.c.

During this step the evaluator selects a sample of web pages and web page states that is representative of the target website to be evaluated. The purpose of this selection is to ensure that the evaluation results reflect the accessibility performance of the website with reasonable confidence. In cases where it is feasible to evaluate all web pages and web page states of a website, which is highly recommended, this sampling procedure can be skipped and the "selected sample" in the remaining steps of this evaluation process is the entire website. In some cases, such as for small websites, this sampling procedure may result in selecting all web pages and web page states of a website.

The actual size of the sample of web pages and web page states needed to evaluate a website depends on many factors including:

- **Size of the website** — websites with more web pages typically require a larger sample to evaluate.

- **Age of the website** — older websites tend to have more (often not easy to find) content with different levels of complexity, consistency, and design and development processes, so that a larger sample is typically required to evaluate.

- **Complexity of the website** — higher complexity requires a larger sample to evaluate; consider the following:

    - **How interactive the content is** —websites with content that is rich in interaction require larger samples to cover the functions provided by a website and the different states that individual web pages can have;

    - **How the content is generated** — websites with content that is aggregated from different sources or that is processed as it is served (at runtime) typically require larger samples to cover the combinations of content that can be generated;

    - **How the content is implemented** — websites that are available in different versions, are served according to users and their preferences, or adapt to access devices require larger samples to cover these different situations.

- **Consistency of the website** — lower consistency requires a larger sample to evaluate; consider the following:

    - **Variety of web page types** — websites with a broader variety of web page types (see Step 2.c: Identify the Variety of Web Page Types) require larger samples to evaluate;

    - **Variety of functionality** — websites with a broader variety of functionality (see Step 2.b: Identify Essential Functionality of the Website), in particular different types of applications, require larger samples to evaluate;

    - **Variety of technologies** — websites with a broader variety of web technologies in use (see Step 2.d: Identify Web Technologies Relied Upon) require larger samples to evaluate;

    - **Variety of coding styles** — websites with a broader variety of coding styles (typically these are from different scripts that generate the code, templates, and web page authors) require larger samples to evaluate.

- **Adherence to development processes** — lower adherence requires a larger sample to evaluate; consider the following:

    - **Formalization of the process** — websites with formalized development and quality assurance processes tend to show more consistency in the coding and quality of the web pages so that they typically require smaller samples to evaluate;

    - **Training for the developers** — websites with designers, developers, and content authors that receive regular training tend to have more consistent accessibility performance so that they typically require smaller samples to evaluate;

    - **Development tools being used** — websites that are developed and maintained using a consistent set of tools such as a content management system (CMS) also tend to be more consistent and require smaller samples to evaluate;

    - **Number of web page authors** — websites that are developed and maintained by a more confined set of web page authors, including content editors, tend to be more consistent and require smaller samples to evaluate.

- **Required level of confidence** — higher confidence in the evaluation results often requires evaluation of a larger sample.

- **Availability of prior evaluation findings** — smaller samples may be required when evaluators have access to prior evaluation findings, including test results from manual and automated accessibility testing.

The selection carried out during this step relies initially on the exploration carried out in Step 2: Explore the Target Website. The selection is also continually refined during the following Step 4: Audit the Selected Sample, as the evaluator learns more about the particular implementation aspects of the target website.

Step 3.a: Include a Structured Sample

Methodology Requirement 3.a: Select web pages and web page states that reflect all identified (1) common web pages, (2) essential functionality, (3) types of web pages, (4) web technologies relied upon, and (5) other relevant web pages.

Select a sample of web pages and web page states that includes:

1. All common web pages and web page states that were identified in Step 2.a: Identify Common Web Pages of the Website:

2. All other relevant web pages and web page states that were identified in Step 2.e: Identify Other Relevant Web Pages;

3. If not already reflected in the previous steps, select additional web pages and web page states with:

    a. Content from each essential functionality identified in Step 2.b: Identify Essential Functionality of the Website;

    b. Content from the different types of web pages identified in Step 2.c: Identify the Variety of Web Page Types;

    c. Content provided using the web technologies identified in Step 2.d: Identify Web Technologies Relied Upon.

Note: An individual web page or web page state may reflect more than one of each of the criteria listed above. For example, a single web page may be representative of a particular design layout, functionality, and web technologies used. The purpose of this step is to have representation of the different types of web pages and web page states, functionality, and web technologies that occur on the website. Careful selection of these representative instances can significantly reduce the required sample size while maintaining appropriate representation of the entire website. The number of required instances of web pages and web page states depends on the particular aspects of the website explained in the previous section, factors influencing the sample size.

Step 3.b: Include a Randomly Selected Sample

Methodology Requirement 3.b: Select a random sample of web pages and web page states, and include them for auditing.

A randomly selected sample of web pages and web page states acts as an indicator to verify that the structured sample selected through the previous steps is sufficiently representative of the content provided on the website. Confidence in the overall evaluation outcome increases when the evaluation results from both selection approaches correlate.

The number of web pages and web page states to randomly select is **10% of the structured sample** selected through the previous steps. For example, if the structured sample selected for a website resulted in 80 web pages and web page states, then the random sample size is 8 web pages and web page states. (Note: The size of the structured sample is different than the size of the website.)

To perform this selection, randomly select unique instances of web pages and web page states from the target website that are not already part of the structured sample selected through the previous steps. Depending on the type of website and the access that an evaluator has for it there are different techniques that may need to be used for this selection. This may include:

- Using a tool that will traverse the website and propose a list of randomly selected web pages and web page states;
- Using a script that will generate a list of all web pages and web page states available on a website, to select from;
- Using server logs, search engines, blind-folded colleagues, and other creative methods to get to a random sample.

Document the web pages and web page states that were randomly selected as these will need to be compared to the remaining structured sample in Step 4.c: Compare Structured and Random Samples.

Note: While the random sample need not be selected according to strictly scientific criteria, the scope of the selection needs to span the entire scope of the website (any web page and web page state on the website may be selected), and the selection of individual web page and web page states does not follow a predictable pattern. Recording the method used to generate the random sample is important for replicability and reliability of the results.

Step 3.c: Include Complete Processes

Methodology Requirement 3.c: Include all web pages and web page states that are part of a complete process in the selected sample.

The selected sample has to include all web pages and web page states that belong to a series presenting a complete process. No web page or web page state in the selected sample may be part of a process, unless all other web pages and web page states that are part of that process are included in the selected sample.

Use the following steps to include the necessary web pages and web page states in the sample:

1. For each web page and web page state selected through Step 3.a: Include a Structured Sample and Step 3.b: Include a Randomly Selected Sample that is part of a process, locate the starting point (web page or web page state) for the process and include it in the selected sample;
2. For each starting point for a process, identify and record at least the default sequence of web pages and web page states to complete the process. Add these web pages and web page states into the selected sample. **Note:** The default sequence follows the standard use case, describing the default path through the complete process. It assumes that there are no user input errors and no selection of additional options. For example, for a web shop application, the user would proceed to checkout, confirm the default payment option, provide all required payment details correctly, and complete the purchase, without changing the contents of the shopping cart, using a stored user profile, selecting alternative options for payment or shipping address, providing erroneous input, and so forth.

3. For each process, identify and record the branch sequences of web pages and web page states that are commonly accessed and critical for the successful completion of the process. Add these web pages and web page states into the selected sample.
   Note: Branch sequences may terminate where they re-enter the default branch of the process. For example, adding a new shipping address will be registered as a critical alternative branch that leads back to the default branch of the process.

Note: In most cases it is necessary to record and specify the actions needed to proceed from one web page and web page state to the next in a sequence to complete a process so that they can be replicated later. An example of such action could be "fill out name and address, and select the 'Submit' button". In most cases the web address (URI will not be sufficient to identify the web page and web page state in a complete process. It is also useful to clearly record when web pages and web page states are part of a process so that evaluators can focus their effort on the relevant changes such as elements that were added, modified, or made visible.

- **Step 4: Audit the Selected Sample**

Methodology Requirement 4: Audit the selected sample of web pages according to Methodology Requirement 4.a, Methodology Requirement 4.b, and Methodology Requirement 4.c.

During this step the evaluator audits (detailed evaluation of) all of the web pages and web page states selected in Step 3: Select a Representative Sample, and compares the structured sample to the randomly selected sample. The audit is carried out according to the five WCAG 2.0 conformance requirements at the target conformance level defined in Step 1.b: Define the Conformance Target.

The five WCAG 2.0 conformance requirements are:

1. Conformance Level

2. Full pages

3. Complete processes

4. Only Accessibility-Supported Ways of Using Technologies

5. Non-Interference

Further guidance on evaluating to these conformance requirements is provided in the following sections. The WCAG 2.0 Layers of Guidance and Understanding Conformance provide more background and guidance on the WCAG 2.0 conformance requirements, which is beyond the scope of this document.

Note: Carrying out this step requires deep understanding of the WCAG 2.0 conformance requirements and the expertise described in section Required Expertise.

Step 4.a: Check All Initial Web Pages

Methodology Requirement 4.a: Check that each web page and web page state in the selected sample that is not within or the end of a complete process conforms to each of the five WCAG 2.0 conformance requirements at the target conformance level.

For each web page and web page state in the sample selected in Step 3: Select a Representative Sample that is not within or the end of a complete process, check its conformance with each of the five WCAG 2.0 conformance requirements, with the target conformance level defined in Step 1.b: Define the Conformance Target. This includes all components of the web page or web page state without activating any functions, entering any data, or otherwise initiating a process. Such functionality and interaction, including web pages and web page states that are within or the end of a complete process, will be evaluated in the subsequent step.

Note: Many web pages and web page states in the sample will have components, such as the header, navigation bars, search form, and others that occur repeatedly. While the requirement is to check full pages, typically these components do not need to be re-evaluated on each occurrence unless they appear or behave differently, or when additional evaluation requirements are defined in Step 1.d: Define Additional Evaluation Requirements (Optional).

WCAG 2.0 Success Criteria

There are typically several ways to determine whether WCAG 2.0 Success Criteria have been met or not met. W3C/WAI provides one set of (non-normative) Techniques for WCAG 2.0, which documents ways of meeting particular WCAG 2.0 Success Criteria. It also includes documented *common failures*, which are known ways in which content does not meet particular WCAG 2.0 Success Criteria. Understanding Techniques for WCAG Success Criteria provides more guidance on the WCAG 2.0 concept of *techniques*.

Evaluators can use such documented guidance to check whether particular web content meets or fails to meet WCAG 2.0 Success Criteria. Documented techniques and failures can also be useful background in evaluation reports. However, it is not required to use the particular set of techniques and failures documented by W3C/WAI. In fact, evaluators do not need to follow any techniques and failures at all. Evaluators might use other approaches to evaluate whether WCAG 2.0 Success Criteria have been met or not met. For example, evaluators may utilize specific testing instructions and protocols that meet the requirements for sufficient techniques, and that may be publicly documented or only available to the evaluators. More guidance on the use of techniques is provided in the previously linked Understanding Techniques for WCAG Success Criteria.

Note: WCAG 2.0 Success Criteria are each formulated as a "*testable statement that will be either true or false when applied to specific web content*". When there is no content presented to the user that relates to specific Success Criteria (for example, no video on the web page), then the Success Criteria are "satisfied" according to WCAG 2.0. Optionally, an evaluation report can specifically indicate Success Criteria for which there is no relevant content, for example, with "not present". Understanding Conformance provides more background and guidance.

Conforming Alternate Versions

Content on a web page or web page state might have alternate versions. For example, video content may be provided in a version with and without captions. In some cases, an entire web page or web page state (or series of them) may be provided as an alternate version to an initial web page or web page state. Conformance to WCAG 2.0 can be achieved with the help of alternate versions that meet the requirements listed in the WCAG 2.0 definition for conforming alternate version. For example, a web page with video content without captions could still meet WCAG 2.0 by providing an alternate version for the video that qualifies to be a *conforming alternate version*. Understanding Conforming Alternate Versions provides further guidance on conforming alternate versions that is beyond the scope of this document.

Note: Alternate versions are not considered to be separate web pages or web page states but part of the content. Web pages and web page states are evaluated together with their alternate versions as one unit (full page).

Accessibility Support

Content on a web page or web page state needs to be provided in a way that is *accessibility supported* (either directly or through an alternate version). For example, the captions for a video need to be provided in a way that they can be displayed to users. The WCAG 2.0 definition for accessibility supported defines specific requirements for the use of web content technologies to qualify as accessibility-supported. Understanding Accessibility-Supported Web Technology Uses provides further guidance on accessibility support that is beyond the scope of this document. However, WCAG 2.0 does not define a particular threshold or set of software that a website needs to support for accessibility. The definition of such a baseline depends on several parameters including the purpose, target audience, and language of website. The baseline used to evaluate a particular website is defined in Step 1.c: Define an Accessibility Support Baseline.

Non-Interference

Content on a web page or web page state may not conform to WCAG 2.0, even though the web page or web page state as a whole might still conform to WCAG 2.0. For example, information and functionality may be provided using web content technologies that are not yet widely supported by assistive technologies or in a way that is not supported by assistive technologies, accompanied by a conforming alternate version for the information and functionality that is accessibility supported. In this case the non-conforming content must not negatively interfere with the conforming content so that the web page or web page state can conform to WCAG 2.0. The WCAG 2.0 conformance requirement for non-interference defines specific requirements for content to qualify as non-interfering. Understanding Requirement 5 provides further guidance on non-interference that is beyond the scope of this document.

Step 4.b: Check All Complete Processes

Methodology Requirement 4.b: Check that all interaction for each web page and web page state along a complete process conforms to each of the five WCAG 2.0 conformance requirements at the target conformance level.

For each complete process identified in Step 3.c: Include Complete Processes, follow the identified default and branch sequences of web pages and web page states, and evaluate each according to Step 4.a: Check All Initial Web Pages. However, in this case it is not necessary to evaluate all content but only the content that changes along the process.

Functionality, entering data, notifications, and other interaction is part of this check. In particular it includes:

- Interaction with forms, input elements, dialog boxes, and other web page components;

- Confirmations for input, error messages, and other feedback from user interaction;

- Behavior using different settings, preferences, devices, and interaction parameters.

Step 4.c: Compare Structured and Random Samples

Methodology Requirement 4.c: Check that each web page and each web page state in the randomly selected sample does not show types of content and outcomes that are not represented in the structured sample.

While the individual occurrences of WCAG 2.0 Success Criteria will vary between the structured and randomly selected samples, the randomly selected sample should not show new *types* of content not present in the structured sample. Also, the outcomes from evaluating the randomly selected sample should not show new findings to those of the structured sample. If the randomly selected sample shows new types of content or new evaluation findings then it is an indication that the structured sample was not sufficiently representative of the content provided on the website. In this case evaluators need to go back to Step 3: Select a Representative Sample to select additional web pages and web page states that reflect the newly identified types of content and findings. Also, the findings of Step 2: Explore the Target Website might need to be adjusted accordingly. This step is repeated until the structured sample is adequately representative of the content provided on the website.

- **Step 5: Report the Evaluation Findings**

Methodology Requirement 5: Report the evaluation findings according to Methodology Requirement 5.a and optionally Methodology Requirement 5.b, Methodology Requirement 5.c, Methodology Requirement 5.d, and Methodology Requirement 5.e.

While evaluation findings are reported at the end of the process, documenting them is carried out throughout the evaluation process to ensure verifiable outcomes. The documentation typically has varying levels of confidentiality. For example, documenting the specific methods used to evaluate individual requirements might remain limited to the evaluator while reports about the outcomes from these checks are typically made available to the evaluation commissioner. Website owners might further choose to make public statements about the outcomes from evaluation according to this methodology.

Step 5.a: Document the Outcomes of Each Step

Methodology Requirement 5.a: Document each outcome of the steps defined in Step 1: Define the Evaluation Scope, Step 2: Explore the Target Website, Step 3: Select a Representative Sample, and Step 4: Audit the Selected Sample.

Documenting the outcomes for each of the previous steps (including all sub-sections) is essential to ensure transparency of the evaluation process, replicability of the evaluation results, and justification for any statements made based on this evaluation. This **documentation does not need to be public**; the level of confidentiality is usually determined by the evaluation commissioner.

Documenting the outcomes for each step includes at least the following:

- **About the Evaluation**
  - Name of the evaluator
  - Name of the evaluation commissioner
  - Date for the evaluation (completion date or duration period)
- **Evaluation Scope**
  - Scope of the website defined in Step 1.a: Define the Scope of the Website
  - Conformance target defined in Step 1.b. Define the Conformance Target
  - Accessibility support baseline defined in Step 1.c: Define an Accessibility Support Baseline
  - Additional requirements, if any, defined in Step 1.d: Define Additional Evaluation Requirements (Optional)
- **Website Exploration**
  - Web technologies relied upon identified in Step 2.d: Identify Web Technologies Relied Upon
  - Optional: Common web pages identified in Step 2.a: Identify Common Web Pages of the Website
  - Optional: Essential functionality identified in Step 2.b: Identify Essential Functionality of the Website
  - Optional: Variety of web page types identified in Step 2.c: Identify the Variety of Web Page Types
  - Optional: Other relevant web pages identified in Step 2.e: Identify Other Relevant Web Pages
- **Representative Sample**
  - Web pages selected through structured sampling in Step 3.a: Include a Structured Sample
  - Randomly selected web pages and selection method used in Step 3.b: Include a Randomly Selected Sample
  - Complete processes selected in Step 3.c: Include Complete Processes
- **Sample Audited**
  - Evaluation outcomes from Step 4.a: Check All Initial Web Pages
  - Evaluation outcomes from Step 4.b: Check All Complete Processes
  - Evaluation outcomes from Step 4.c: Compare Structured and Random Samples

Note: Depending on the desired granularity of the report documentation, the outcomes of Step 4: Audit the Selected Sample may be provided for each evaluated web page and web page state, or aggregated over the entire sample. Reports should include at least one example for each conformance requirement and WCAG 2.0 Success Criterion not met. It is also good practice for evaluators to indicate issues that occur repeatedly.

Reports may also include additional information depending on any additional evaluation requirements defined in Step 1.d: Define Additional Evaluation Requirements (Optional). For example, an evaluation commissioner may request a report indicating every failure occurrence for every web page and web page state in the selected sample, more information about the nature and the causes of the identified failures, or repair suggestions to remedy the failures.

Step 5.b: Record the Evaluation Specifics (Optional)

Methodology Requirement 5.b: Archive the web pages and web page states audited, and record the evaluation tools, web browsers, assistive technologies, other software, and methods used to audit them (Optional).

While optional, it is good practice for evaluators to keep record of the evaluation specifics, for example to support conflict resolution in the case of dispute. This includes archiving the web pages and web page states audited, and recording the evaluation tools, web browsers, assistive technologies, other software, and methods used to audit them. This recording is typically kept internal and not shared by the evaluator unless otherwise agreed on in Step 1.d: Define Additional Evaluation Requirements (Optional).

Records of the evaluation specifics could include any of the following:

- Copies of the files and resources of the web pages and web page states;
  Note: Some tools can save the dynamically generated or modified content (DOM) as displayed during the evaluation rather than the initial content of the files and resources, which is often different;

- Screenshots (screen grabs) of the web pages and web page states;

- Description of the path to locate the web pages and web page states, especially when they are part of a process;

- Description of the settings, input, and actions used to generate or navigate to the web pages and web page states. Specific test credentials (user-IDs, etc.) required to replicate a unique data set or workflow;

- Names and versions of the evaluation tools, web browsers and add-ons, assistive technology, and other software used;

- The methods, procedures, and techniques used to evaluate conformance to WCAG 2.0.

This recording may apply globally for the entire evaluation, to individual web pages, or to individual checks carried out within the audited web pages and web page states. A table or grid may be useful to record what was used for the different web pages and web page states audited.

Note: Records of the evaluation specifics may include sensitive information such as internal code, passwords, and copies of data. They may need particular security and privacy precautions.

Step 5.c: Provide an Evaluation Statement (Optional)

Methodology Requirement 5.c: Provide a statement describing the outcomes of the conformance evaluation (Optional).

Reminder: In the majority of situations, using this methodology alone does not result in WCAG 2.0 conformance claims for the target websites; see Relation to WCAG 2.0 Conformance Claims for more background.

Website owners may wish to make public statements about the outcomes from evaluations following this methodology. This can be done when at least every non-optional methodology requirement is satisfied, the conformance target defined in Step 1.b. Define the Conformance Target is satisfied by all web pages and web page states audited (in Step 4: Audit the Selected Sample), and the website owner commits to ensuring the validity and maintaining the accuracy of the evaluation statement made.

An evaluation statement according to this methodology includes at least the following information:

1. **Date** of when the evaluation statement was issued;
2. **Guidelines, title, version and URI:** "Web Content Accessibility Guidelines 2.0 at http://www.w3.org/TR/WCAG20/";
3. **Conformance level evaluated**: Level A, AA or AAA, as defined in Step 1.b. Define the Conformance Target;
4. **Definition of the website** as defined in Step 1.a: Define the Scope of the Website;
5. **Web technologies relied upon** as identified in Step 2.d: Identify Web Technologies Relied Upon;
6. **Accessibility support baseline** as defined in Step 1.c: Define an Accessibility Support Baseline.

Evaluation statements according to this methodology can also be made when only partial conformance to WCAG 2.0 has been achieved. In such cases the evaluation statements also include the following information:

7. **Website areas** that do not conform to WCAG 2.0;
8. **Reason for not conforming to WCAG 2.0:** "third-party content" or "lack of accessibility support for languages".

1.6. Roles involved

Building a website is not an easy task: you need to select content, define a structure, make it look good and, finally, write the code that will make all of this a reality.

In general, there are three well-defined roles involved in website creation:

- **Content writers/editors**: they choose the information to be conveyed, as well as write or select website text, images, and other content.
- **Layout artists**: they distribute and arrange the above material as attractively as possible, choose the templates for the content, select the colours, and give form to website elements in order to give the website its final identity.
- **Developers**: They write the code that is read by the browser, which in turn displays the web pages as the designer intended.

None of these roles is independent from the other. Everyone needs to work together, and, in some cases, one person may take on all of these roles. What is important to understand is that different roles will have widely different accessibility considerations. This guide is geared toward the first of these, the people responsible for content.

This is based on the consideration that most of the decisions regarding website appearance and corporate image are made by content editors/writers which in turn has a significant bearing on the work of website designers. Web accessibility aspects associated with basic website layout will also be dealt with as part of their work.

1.7. Tools

At this point some questions may arise about where to start or how to take into account the needs of all disabilities. There are numerous tools on the market, both free and paid, that can help improve web accessibility of your website or your app.

For those in charge of the content, there are numerous tools that can be used for reviewing and generating content.

Browser Extensions

Extensions (also known as plug-ins or add-ons) for browsers add functions that facilitate website development or enable the use of Internet services.

All browsers support extensions, but browsers based on the Chromium operating system (Chrome, Edge, Brave, etc.) have a greater variety of extensions, due to their wider use.

To install an extension or add-on, you need to access the Chrome Web Store. Here, you can search for your extension of choice. To add it to your browser, click on the "Add to Chrome" button.

The following extensions are helpful in detecting some accessibility issues:

Web Developer Extension

This extension is widely recommended by many accessibility specialists, including the Observatory itself. Although it was not developed for this purpose alone, it does offer many tools enabling you to see how web pages behave in this regard. Among its features you will find the image handling tool, which is especially useful as it allows you to hide images, show their alternative text, replace images with alt text, etc.

It can be downloaded **here**.

Kontrast

This simple extension offers a dropper feature to enable you to select element and background colours to measure contrast between them, indicating whether or not they meet the minimum requirements of WCAG.

Screen readers

Strictly speaking, the use of a screen reader is not necessary, yet by doing so developers can browse a web page much in the same way a visually impaired person would, and thus improve the results of their work.

All platforms today provide a screen reader of some kind (Windows offers its Narrator tool, Android has Talkback and iOS offers VoiceOver to its users). For the desktop, the most commonly used tools among blind users are JAWS or NVDA.

JAWS (*Job Access With Speech*)

Designed for the blind and the visually impaired, this screen reader software makes computers more accessible for people with visual disabilities. Its operation is simple: the application converts the content displayed on the screen into sound, enabling the user to access or browse this information without needing to see it.

✓ It offers powerful speech synthesizers, PDF reader, image recognition, optimized navigation for blind users, braille display support, and is compatible with many Windows and Office applications.

✗ The main drawback is its cost, which can be high for many users. Supports only 7 languages.

NVDA (NonVisual Desktop Access)

This is another widely used tool that offers the further advantage of being open source – this means it can be downloaded and used for free. While it is not particularly easy to use, it still provides a very rough idea of what users perceive.

✓ Free software, ideal for beginners, good speech synthesizers (eSpeak, Sapi 4/5), braille display support, supports multiple languages.

✗ Only speech synthesizers are free, navigation is somewhat cumbersome.

The application can be downloaded at https://nvda.es/.

Narrator

This screen reading app is included with Windows, so nothing needs to be downloaded or installed. This tool is the ideal starting point for those new to screen readers.

✓ Simple text reading, easy navigation with keyboard commands, native integration with Windows.

✗ It has fewer navigation options than other screen readers such as NVDA or JAWS.

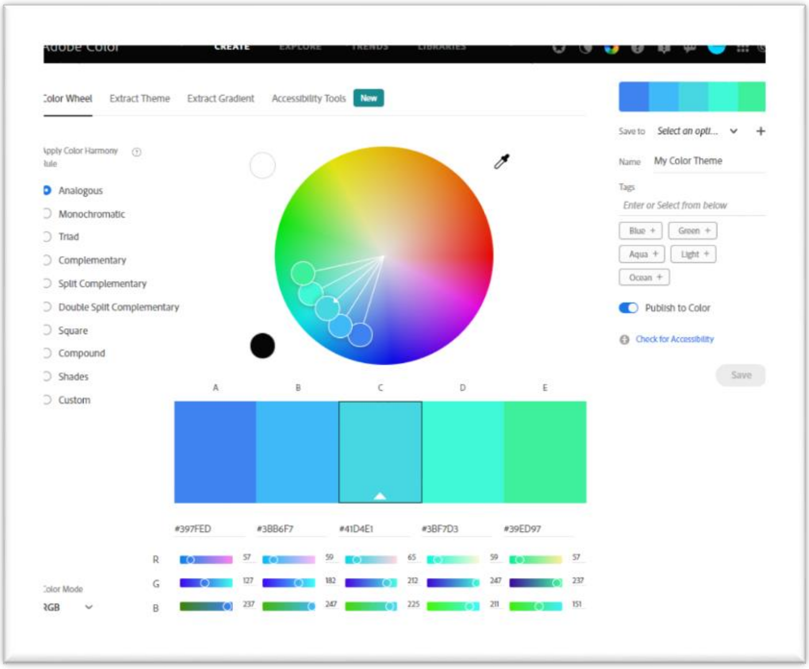
On a Windows computer, search for Narrator and you can start trying it out right away. Support is available here.

Other tools

Adobe Color

This is a very useful tool found on the Adobe web page which allows you to create a colour theme for use on your website based on a few colours.

It generates complementary colours, colours within the same range, etc. It offers a variety of options for the creation of different colour combinations, but it also has a section which helps create accessible colour themes:



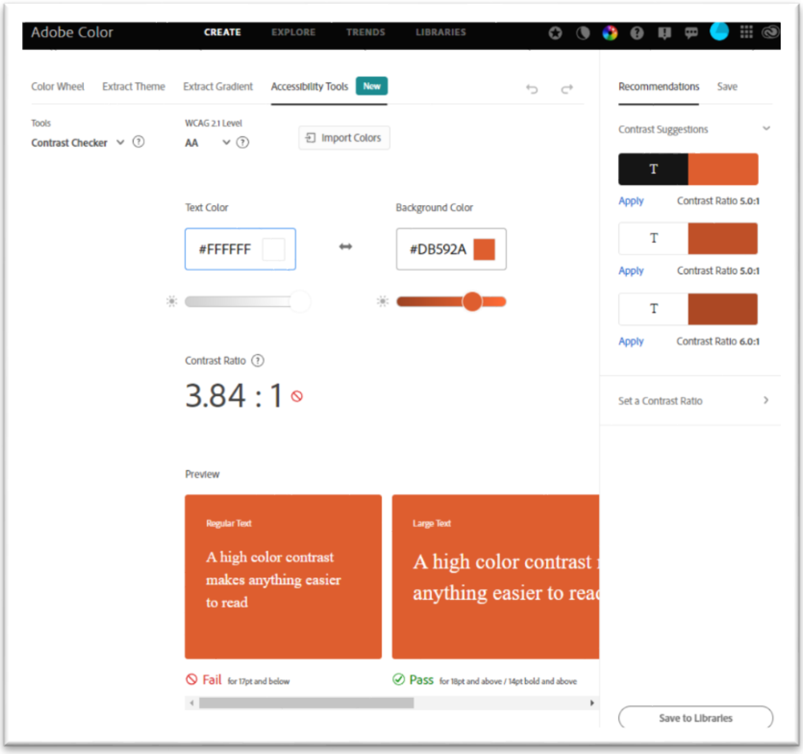
Apply Color Harmony Rule: Analogous

Name: My Color Theme

	A	B	C	D	E
	#397FED	#3BB6F7	#41D4E1	#3BF7D3	#39ED97
R	57	59	65	59	57
G	127	182	212	247	237
B	237	247	225	211	151

Color Mode: RGB

Illustration 2. Adobe Color tool: colour wheel.



Tools: Contrast Checker

WCAG 2.1 Level: AA

Text Color: #FFFFFF

Background Color: #DB592A

Contrast Ratio

3.84 : 1

A high color contrast makes anything easier to read

A high color contrast makes anything easier to read

Fail for 17pt and below

Pass for 18pt and above / 14pt bold and above

Contrast Ratio 5.0:1

Contrast Ratio 5.0:1

Contrast Ratio 6.0:1

Illustration 3 . Adobe Color screen: accessibility tool.

This tool can be accessed at https://color.adobe.com/en/.

2. Accessibility of content: WCAG 2.1 and its recommendations

As mentioned in the previous section, WCAG are the accessibility guidelines developed by the W3C consortium, an organization that also prepares Internet standards.

There is a widespread belief that this is a compilation of rules to follow in web design, yet these guidelines are really a series of recommendations to help developers to achieve the greatest web accessibility possible. They are based on the experience of WCAG experts, who have carried out in-depth studies on the different types of disabilities and related user needs, as well as the easiest ways to adapt websites and Apps to each type. If we follow these recommendations and adapt our work methods accordingly, web accessibility can be improved.

One of the main issues with these guidelines is that there are many and they are not always easy to understand. Some are straightforward and easy to implement, but others are quite complex and involve highly technical topics meant for developers.

This section will cover the WCAG recommendations, mainly focusing on how website content should be prepared to ensure its accessibility. This affects both websites and mobile apps, as well as the preparation of social media content.

The WCAG recommendations are divided into four fundamental principles, known as the pillars of accessibility. In basic terms these pillars state that a website or app should be:

1. **Perceivable**: What is displayed on a website or app must be perceivable by everyone. It is important to consider that perceivability does not only refer to visual elements: the content must also be perceived by people with other types of sensory disabilities, regardless of their nature.
2. **Operable**: All users should be able to interact with whatever is on the page. Remember that not everyone can use a mouse and keyboard, instead, users must be able to interact with other interfaces as well.
3. **Understandable**: all users must be able to understand how to use the web page and its content.
4. **Robust**: the web page must be built in such a way that it can be interpreted by all user agents, that is, it must be compatible, and its information must be available to accessibility services.

All of these principles have a bearing on a content editor's responsibilities, except perhaps the fourth which is more geared towards developers. The way these responsibilities are to be translated is indicated below.

2.1. Perceivable

Web page content must be perceivable, but how do you know if this is the case?

Image processing

This type of content is mostly used in virtual spaces (websites, apps, and social media), and therefore care should be taken when using it.

In order to process images properly, it is important to know how they are classified, and to ensure they are accompanied by a descriptive text and tags. This information allows screen readers to correctly read the function of images such as buttons, which are usually represented by icons.

What follows is a list of the principal types of images and how each should be handled.

Decorative images

First of all, let's delve into the most commonly used image type on the web: decorative images.

In most cases, images used on websites have a decorative purpose, that is, to make the page more attractive. They may also be used to draw attention to particular content.

In these cases, a null (empty) alt text should be provided. This is makes sense because if the image is only there to improve the visual appearance of the website, then for a blind user, this type of image does not add any useful information:

Illustration 4. Example of decorative image.

Still, people may find it difficult to decide when an image is decorative or not.

A good way to determine this is to delete the image and see if the same information is being conveyed after the image has been removed. If this is true, then image is decorative and does not require alternative text.

On the other hand, if you feel as though something is missing, then the image itself is part of the information that needs to be conveyed and you will need to find a suitable alternative text. Once you choose the alt text, you can replace the image with the text to determine whether the web page is information is complete without images.

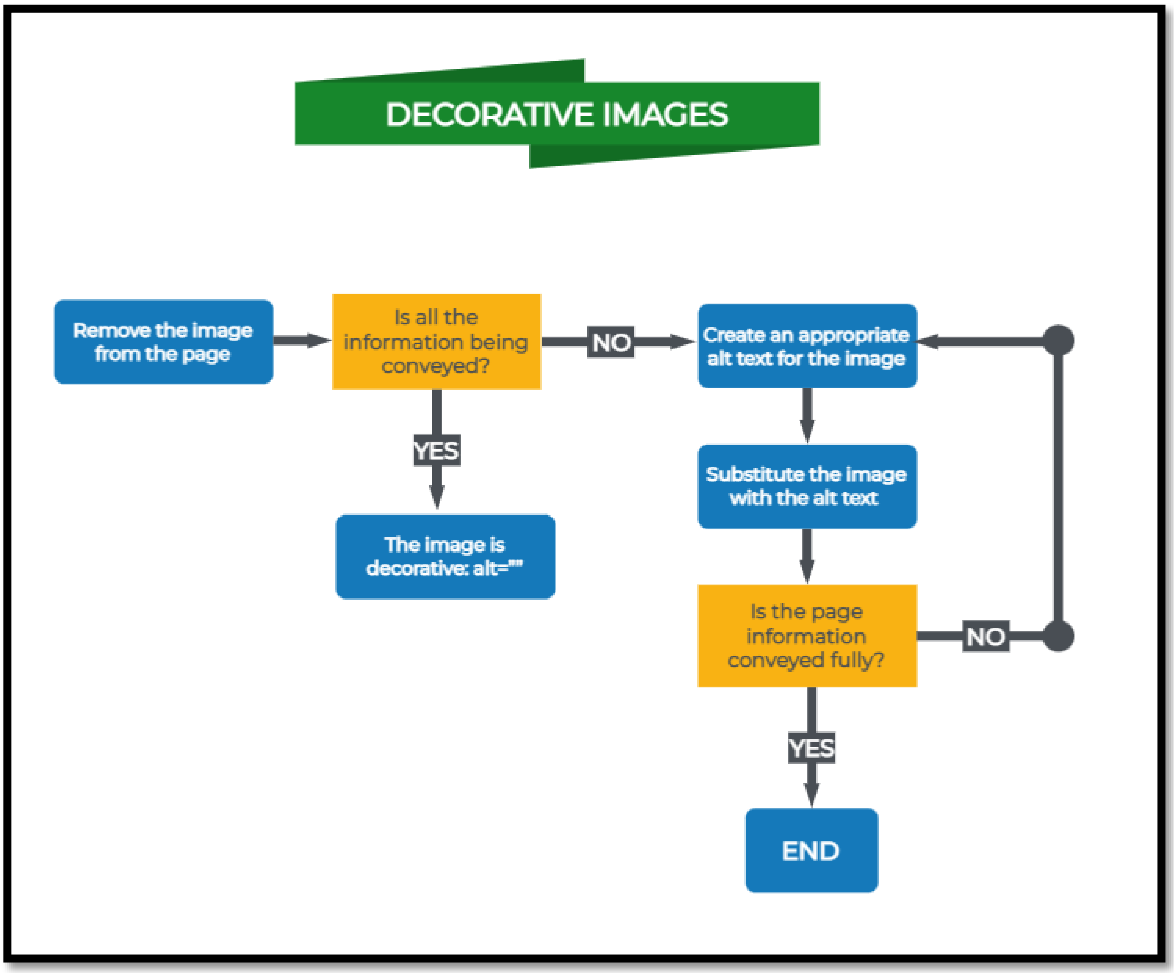
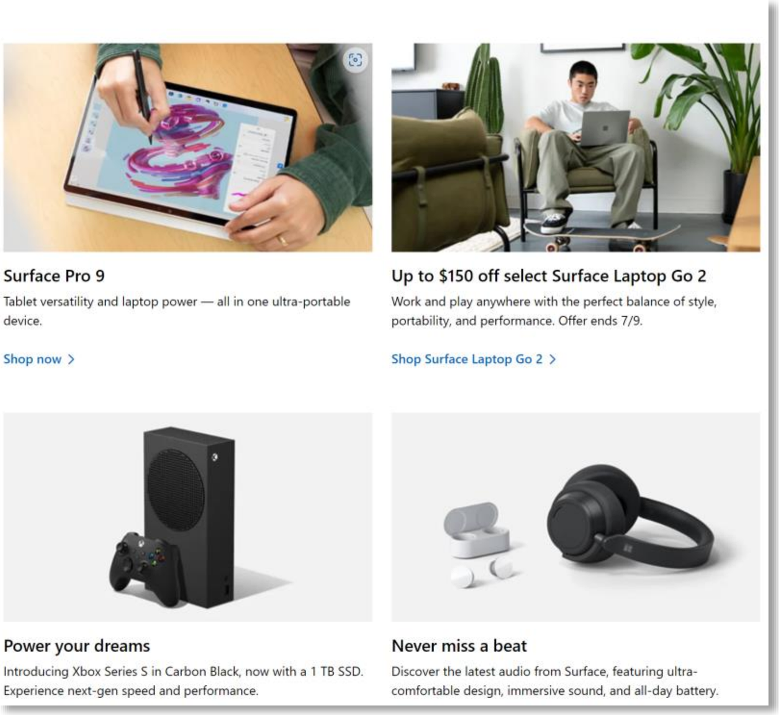

DECORATIVE IMAGES
Remove the image from the page
Is all the information being conveyed?
NO
Create an appropriate alt text for the image
YES
Substitute the image with the alt text
The image is decorative: alt=""
Is the page information conveyed fully?
NO
YES
END

Illustration 5.Flow diagram: identifying decorative images.

To assess this, that is, to be able to hide images or replace them with alt text, there are a number of tools you can use that offer this function. Of the tools indicated earlier, *Web Developer Tools*, offers features that are especially useful for this task.

We'll show you how to use it on a real website.

This is what one random web page looks like:



Surface Pro 9

Tablet versatility and laptop power — all in one ultra-portable device.

Shop now >

Up to $150 off select Surface Laptop Go 2

Work and play anywhere with the perfect balance of style, portability, and performance. Offer ends 7/9.

Shop Surface Laptop Go 2 >

Power your dreams

Introducing Xbox Series S in Carbon Black, now with a 1 TB SSD. Experience next-gen speed and performance.

Never miss a beat

Discover the latest audio from Surface, featuring ultra-comfortable design, immersive sound, and all-day battery.

Illustration 6. Screenshot of a web site Home.

As you can see, the page contains some images. *Web Developer Tools* allows you to hide the images in order to check if they contribute to web page information. To do this, click on the ⚙ icon to open the tool's menu:

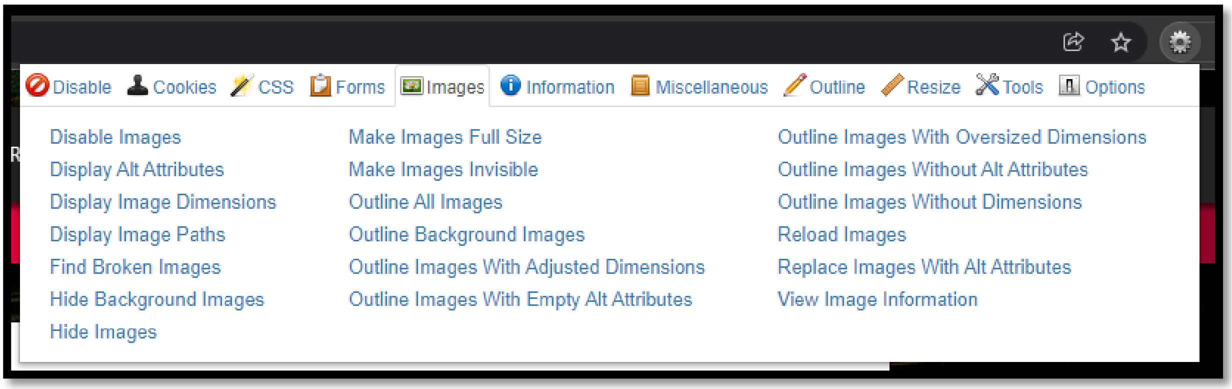
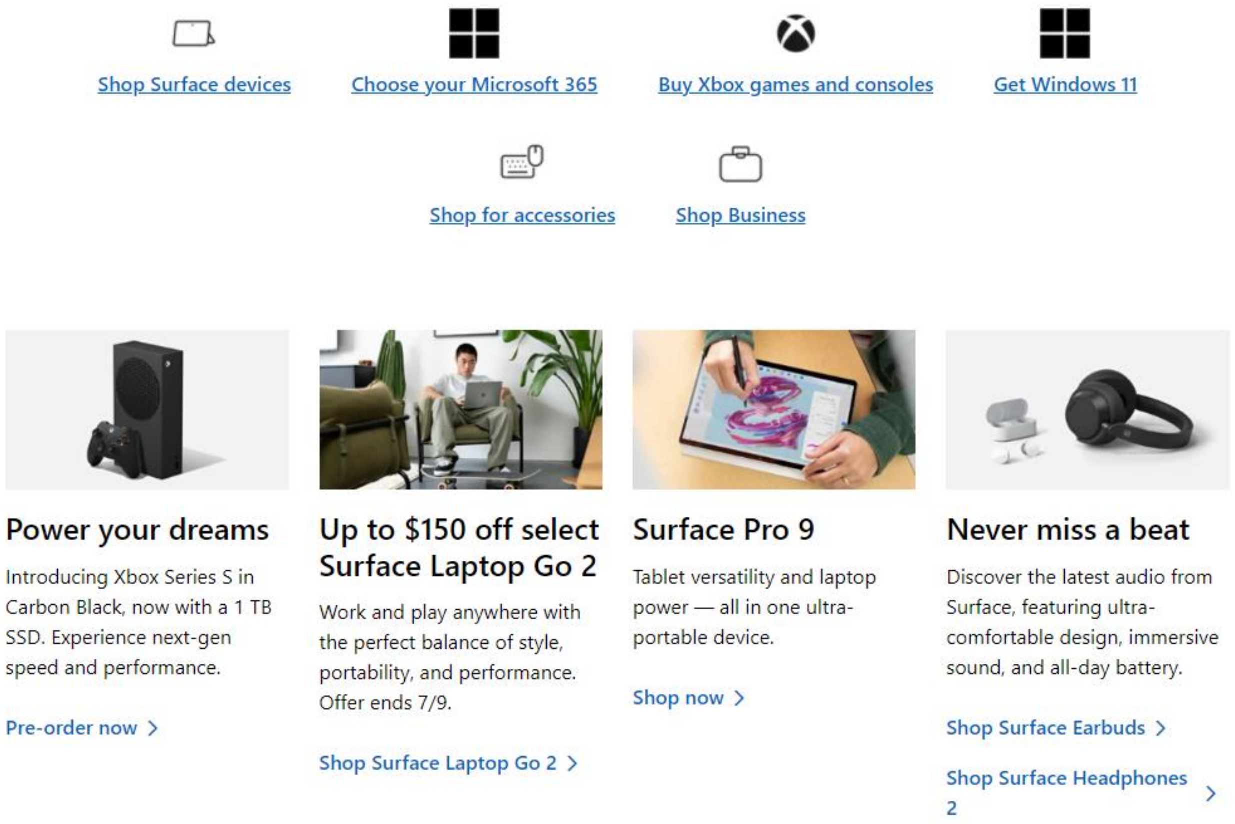


Illustration 7. Screenshot of Web Developer Tools options

Under the *Images* tab (as shown in the previous screenshot), select *Hide Images* or *Make Images Invisible* to hide images.

The screenshot below shows the web page after the *Make Images Invisible* option is selected:



Shop Surface devices

Choose your Microsoft 365

Buy Xbox games and consoles

Get Windows 11

Shop for accessories

Shop Business

Power your dreams

Introducing Xbox Series S in Carbon Black, now with a 1 TB SSD. Experience next-gen speed and performance.

Pre-order now >

Up to $150 off select Surface Laptop Go 2

Work and play anywhere with the perfect balance of style, portability, and performance. Offer ends 7/9.

Shop Surface Laptop Go 2 >

Surface Pro 9

Tablet versatility and laptop power — all in one ultra-portable device.

Shop now >

Never miss a beat

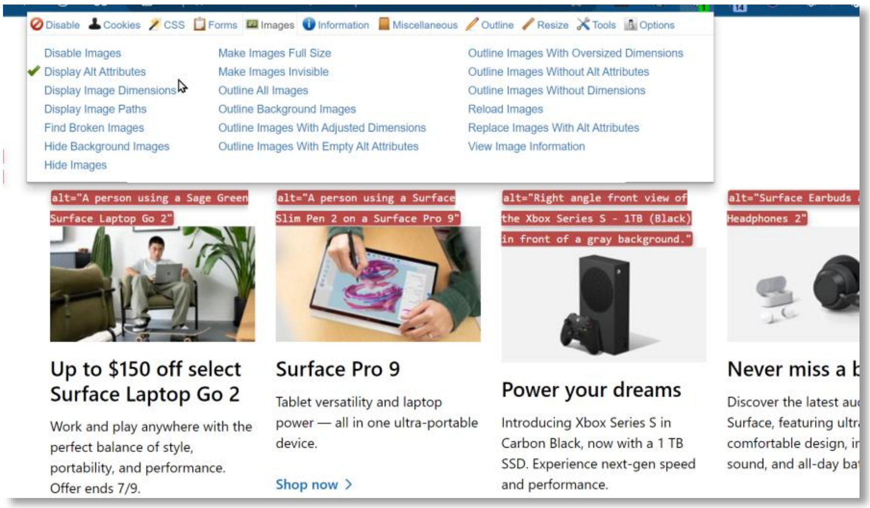
Discover the latest audio from Surface, featuring ultra-comfortable design, immersive sound, and all-day battery.

Shop Surface Earbuds >

Shop Surface Headphones 2 >

Illustration 8. Screenshot of Microsoft web page after images have been hidden.

The next option offered by this tool enables developers to display the image with its text alternative. To do this simply uncheck the option above and select *Display Alt Attributes*. This is what the web page will look like:



alt="A person using a Sage Green Surface Laptop Go 2"

Up to $150 off select Surface Laptop Go 2

Work and play anywhere with the perfect balance of style, portability, and performance. Offer ends 7/9.

alt="A person using a Surface Slim Pen 2 on a Surface Pro 9"

Surface Pro 9

Tablet versatility and laptop power — all in one ultra-portable device.

Shop now >

alt="Right angle front view of the Xbox Series S - 1TB (Black) in front of a gray background."

Power your dreams

Introducing Xbox Series S in Carbon Black, now with a 1 TB SSD. Experience next-gen speed and performance.

alt="Surface Earbuds a Headphones 2"

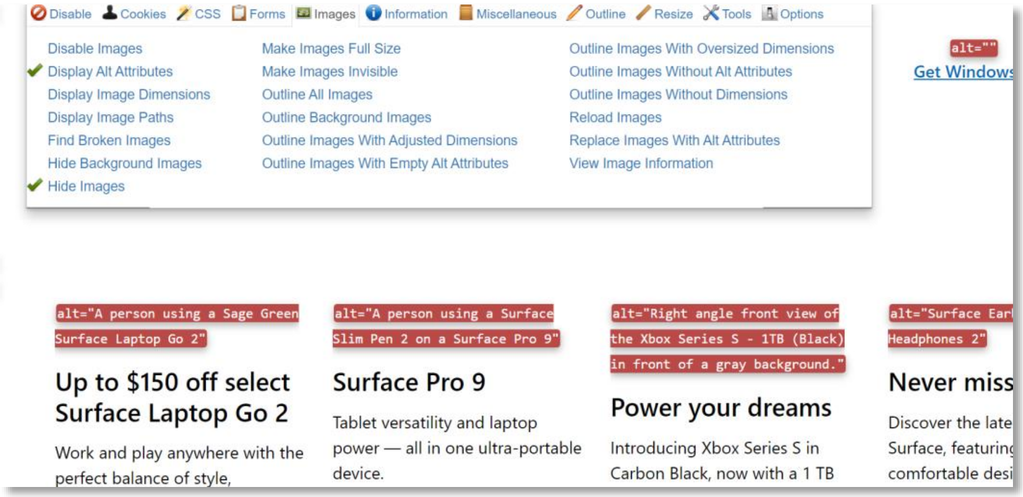
Never miss a b

Discover the latest au Surface, featuring ultr comfortable design, in sound, and all-day ba

Illustration 9. Screenshot of above website where the text alternatives for images are displayed.

As the above screenshot shows, all the images are accompanied by the same text alternatives, yet none of the alt text actually offers much information.

The last option replaces images with their text alternatives, which is what a screen reader will read. Here, you need to activate both the *Display Alt Attributes* and the *Make Images Invisible* options at the same time. The page will look like this:



alt=""

alt="A person using a Sage Green Surface Laptop Go 2"

Up to $150 off select Surface Laptop Go 2

Work and play anywhere with the perfect balance of style,

alt="A person using a Surface Slim Pen 2 on a Surface Pro 9"

Surface Pro 9

Tablet versatility and laptop power — all in one ultra-portable device.

alt="Right angle front view of the Xbox Series S - 1TB (Black) in front of a gray background."

Power your dreams

Introducing Xbox Series S in Carbon Black, now with a 1 TB

alt="Surface Earl Headphones 2"

Never miss

Discover the late Surface, featuring comfortable desi

Illustration 10. Screenshot of the web page where images are replaced by their text alternatives.

From the point of view of a blind person, the web page in this example conveys the same meaning if both the images and text alternatives are removed, since the information is the same in either case. We can then conclude that the images are decorative. A blind person wouldn’t be able to see the images and, if the text alternative doesn’t provide them with any information, adding more information to the page is unnecessary.

It is important to remember that decorative images do have text alternatives: all images that are part of the content of a page must carry an alternative text, but in the case of decorative images, this text is null or empty. That is, if the image does not add any information, the alternative text is empty, but still exists.

This is easy to do when designing the page directly: just leave the alt attribute of the image tag empty (alt=“”). However, in web content management systems this must be explicitly indicated. You will need to consult their help sections for instructions. For example, in WordPress, all images default to alt=“”, which leaves the image alt text blank.

Web Developer Tool is very useful in that it allows you to check whether images have correct alternative texts or if they actually require one. It is important to look at the web page from the user's point of view, to determine whether a text alternative should be included or if it's better to leave it empty.

Informative images (choosing text alternatives)

For images that do provide information and therefore require a text alternative, choosing an appropriate descriptive text is important and you need to be clear on what the purpose of the image is.

As a general rule, text alternatives should not include expressions such as "image of…" or "picture of", as screen readers already identify these elements as images and add that information automatically.

Informative images are broken down into the following types:

Images as labels

These images serve as labels for other elements. The most common example are icons. These images are associated with a clear concept and may be used to replace a title, for instance.

For example:

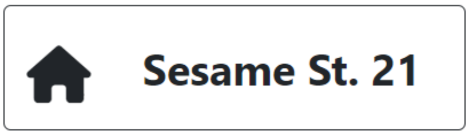


Illustration 11. Example of an icon.

What does the icon of a house alongside an address indicate? In this case, the image represents the concept of "address", and it serves as a substitution for a text telling us the same thing. The concept or text that the image replaces is the text alternative, which means that "address" could be used here:

alt="address"

Sesame St. 21

Illustration 12. Text alternative text for an icon.

When a user with a screen reader visits a page with the above icon, they will hear: "address image" followed by the address that follows it. This is in keeping with what needs to be expressed, so the text alternative chosen here is appropriate.

Functional images

Functional images are used as interactive user interface controls, such as buttons, links, etc.

Illustration 13. Example of a functional image.

Functional images must always be associated with a text, if not, then accessibility will suffer greatly. In this case, the function of the text will be twofold: it replaces the image and indicates the function of that particular interface element.

Image as a button or link: the image is the only visible element that fulfils these functions. Their text alternative indicates the function they perform. Here are a few examples:

alt="Search"

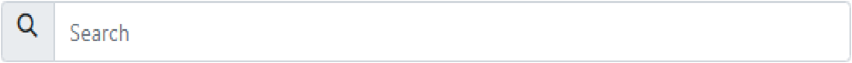


Illustration 14. Text alternative for a search box.

alt="Go to YouTube"

Illustration 15. Alternative text for a YouTube icon.

The first example is a button and its text alternative is the function performed by the button: to start a search. The second is a link, its alternative text indicates the destination of the link.

Button with an image and text: it is common to find buttons that include images with texts as in the following example:

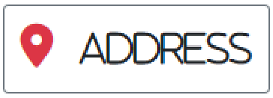


Illustration 16. Example of a button with an image and text.

In the above case, it may seem to be a functional image. However, upon closer inspection we find that it is actually a decorative image since the button still makes sense if the image is removed. That is why the text alternative should be empty:

alt=""

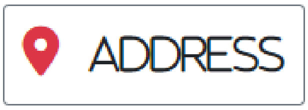


Illustration 17. Text alternative for a button with an image and text.

Infographics and complex images

In the previous examples we have seen images whose text alternatives are empty or very brief.

However, there are other images that convey a lot of information:

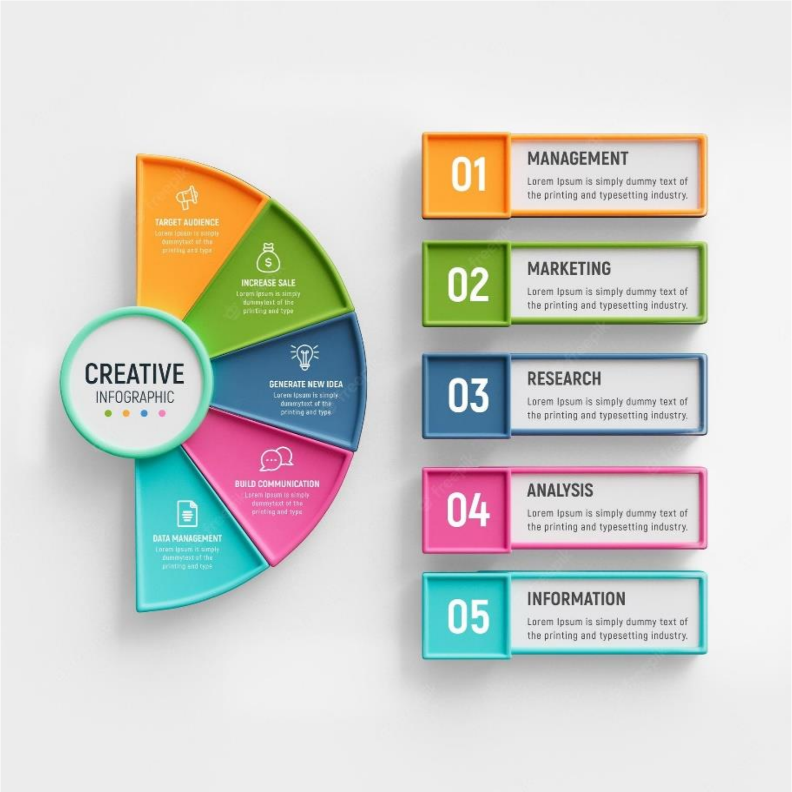

CREATIVE
INFOGRAPHIC
TARGET AUDIENCE
Lorem Ipsum is simply dummytext of the printing and type
INCREASE SALE
Lorem Ipsum is simply dummytext of the printing and type
GENERATE NEW IDEA
Lorem Ipsum is simply dummytext of the printing and type
BUILD COMMUNICATION
Lorem Ipsum is simply dummytext of the printing and type
DATA MANAGEMENT
Lorem Ipsum is simply dummytext of the printing and type
01
MANAGEMENT
Lorem Ipsum is simply dummy text of the printing and typesetting industry.
02
MARKETING
Lorem Ipsum is simply dummy text of the printing and typesetting industry.
03
RESEARCH
Lorem Ipsum is simply dummy text of the printing and typesetting industry.
04
ANALYSIS
Lorem Ipsum is simply dummy text of the printing and typesetting industry.
05
INFORMATION
Lorem Ipsum is simply dummy text of the printing and typesetting industry.

Illustration 18. Infographic.

Although there are no size limitations for the length of text alternatives, they should not be too long as they will be read aloud by assistive technologies as soon as they are detected, which could be bothersome for users.

For images that convey a lot of information, there are a number of options, depending on the image type.

Complex images

Text alternatives should not be used to provide a lot of information, which could overwhelm users.

Let's look at an example:

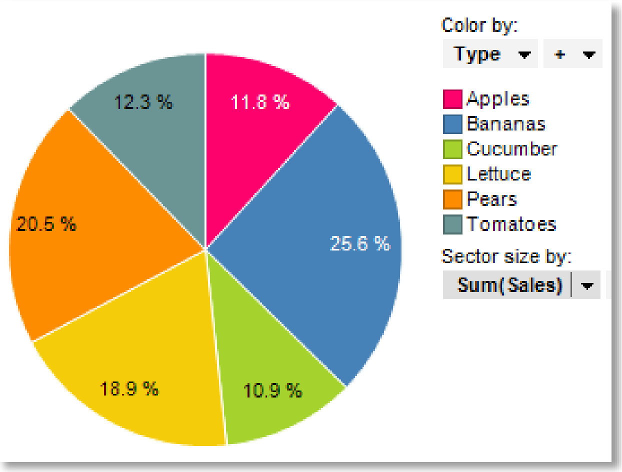


Illustration 19. Pie chart; example of a complex image.

This graph shows the distribution of visits to links found on a page, according to different areas of interest.

In these cases, we need to use two descriptions that complement each other: the text alternative that describes the purpose of the image; and the rest of the information in a longer description associated with the image. The user receives the basic information of the image and decides whether or not they are interested in knowing the rest.

In the previous example, this could be indicated as follows:

- **Text alternative:** percentage distribution of clicks by area of interest.
- **Long Description:** Pie chart showing the distribution of clicks based on the user's area of interest. The percentages shown are:
    - 22.1% Sports
    - 20.3% Culture
    - 19.9% Society
    - 17.9% Technology
    - 19.8% Other

There are several ways that longer descriptions can be added: One strategy is to include it right after the image and another is to add a link to the description located nearby. In HTML, the <figure> tag is a container for figures and images that can hold any information pertaining to the image.

Image maps

These types of images are even more complex, since they consist of an image made up of other functional images the user can interact with.

An example of this type of image is an interactive geographical map. Here, when users click on a certain part of the map, further information will appear, or simply a link leading to a page with more details.

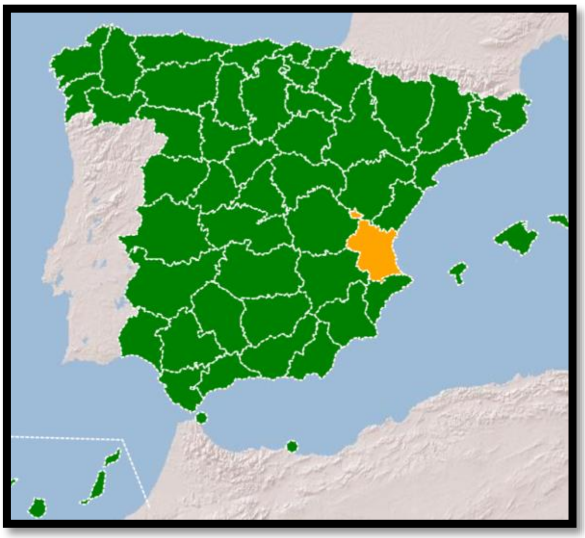


Illustration 20. Example of interactive map.

In this example, we see the map on the one hand, which is an image requiring its own text alternative; and on the other hand, a series of functional images that need to be handled as indicated in the section on functional images.

A possible alt text for the image could be: “Map of Spain with active areas. Click on the area of interest to obtain detailed information on this area.”

The map is made up of many images, and therefore each one must contain its own alternative text and longer description, if applicable.

Continuing with the above example, and if we imagine this map indicates the case rate of Covid-19 by province, the text alternative for the indicated area could be written as follows:

- Text alternative: Province of Valencia, Covid-19 case rate data by age.
- Long description: detailed percentage by age groups.

In addition to treating maps as images, remember that they also act as interface elements and when users click on them, something happens (further information is provided). This means that all users must be able to access all elements regardless of their navigation method.

Remember that users must be able to navigate around the image map using their keyboard.

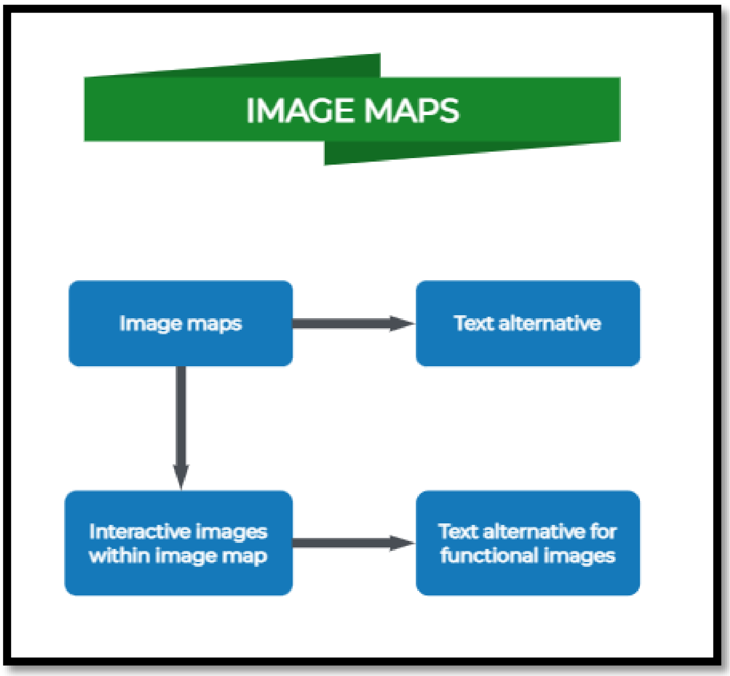


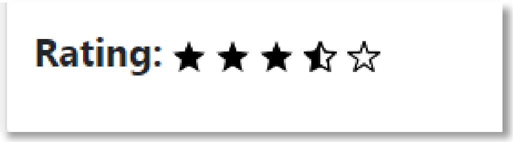
IMAGE MAPS
Image maps
Text alternative
Interactive images within image map
Text alternative for functional images

Illustration 21. Diagram showing how to handle image maps.

Image groups

In cases where there are several images that do not convey information individually, but make sense as a whole, you only have to include a text alternative for the entire group. The rest will contain empty text alternatives.

A classic example is a star rating: imagine that a review includes an image containing five images of stars that are either full or empty. In this case, a score of 3.5 could be represented as:



Rating: ★ ★ ★ ⯪ ☆

Illustration 22. Example of a group of images showing a rating.

Since you only need to include an alternative text for the first image indicating the score, you could write it as “rating of three point five” with the other four images containing the empty alternate text:

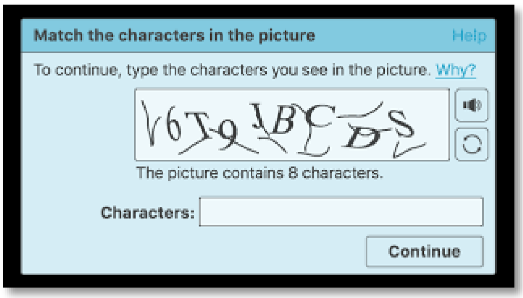
```
  Rating:
          < img src= "full-star.png" alt= "Rating of 3.5" /> < img src= "full-
star.png" alt= "" /> < img src= "full-star.png" alt= " " /> < img src= "half-
star.png" alt= "" /> < img src= "star.png" alt= "" />
```

CAPTCHAS

CAPTCHAS are elements used to determine whether a given web-based service is being used by a human, and not a robot. A Captcha may consist of a visual test wherein the user has to identify a series of letters, numbers, or images. In many cases, these tests pose a real difficulty for people with disabilities.

The earliest CAPTCHAS were tests consisting of identifying a series of distorted alphanumeric characters; not only were they difficult for blind, partially sighted, or dyslexic users, but also difficult to understand for most users in general.

To resolve this issue, some CAPTCHAs may also offer an alternative means consisting of distorted audio, but the difficulties persist:



Match the characters in the picture

To continue, type the characters you see in the picture. Why?

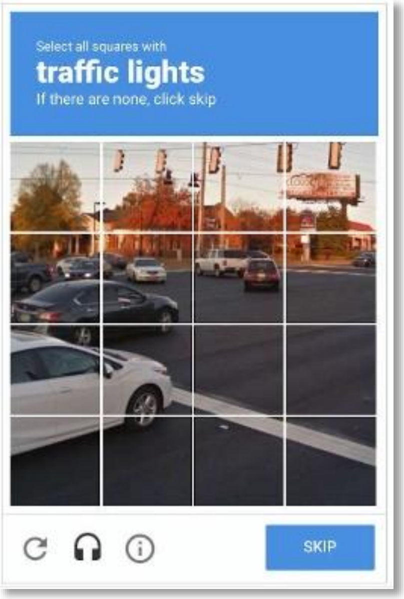
The picture contains 8 characters.

Characters:

Illustration 23. Captcha example using alphanumeric character distortion.

For a long time, many have questioned the effectiveness of this type of CAPTCHA, since it is possible for unauthorized access to occur with both of these methods. As a result, they are both ineffective and greatly restrict accessibility.

Other more recent methods use photograph-based puzzles in which you are asked to check, for example, all the photos containing traffic lights. These methods are not accessible for the visually impaired or even the elderly either:



Select all squares with
traffic lights
If there are none, click skip

Illustration 24. Example of CAPTCHAS based on a puzzle containing photographs.

The current trend is to avoid using this type of CAPTCHAS and to opt for others based on the way the user interacts with the page in order to determine if they are human or not. A very popular system is Google's reCAPTCHA, which only requires users to check a box:

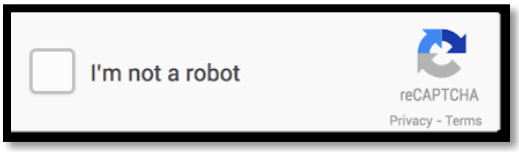


Illustration 25. Google reCAPTCHA.

Or the new reCAPTCHA v3, which does not require any action by the user and therefore does not affect accessibility, making it the perfect solution for the task at hand:

Illustration 26. Google's reCAPTCHA v3.

There are many more ways to verify that interaction with the page is taking place with a human and not a machine, but the ones mentioned here are the most commonly used.

In any case, if for some reason you would like or must use an older CAPTCHA version, it is important to know that to meet Level AA accessibility criteria, you need to ensure the following:

- The image-based CAPTCHA must have a text alternative that states two things: its purpose and that an alternative method exists that is not based on images.
- An alternative method indeed does exist, usually in the form of an audio file. Here, you need to ensure that the controls needed to play the audio are accessible as well. They need to be activated using the keyboard and have an accessible name.

Carousels

An image carousel is a set of photos with text (and optionally buttons) that are usually presented in full size. These are often found on home pages of a website, for example.

Although they are eye-catching and may include a lot of information presented in a dynamic and attractive way, special care must be taken when using them from the point of view of accessibility.

To study their impact on accessibility, let's take a closer look at some of their most relevant characteristics:

- While other types of content may be used, carousels usually contain a set of images which are often accompanied by text. Each one is called a slide.
- The carousel is usually in movement, displaying each of its slides in sequence.
- Slides often act as links to other pages.
- In many cases they include buttons to control or even stop slide movement.

First, it is important to bear in mind that carousels contain images and therefore each image must have a text alternative. Usually, the text alternatives that are given to images are the same as the message accompanying the slide, and as a result, any users exploring the slide with a screen reader, will get the same message twice. This is redundant and spoils the user's experience.

In these cases, it is better to consider the image as decorative and therefore its text alternative should be empty.

Second, carousels are constantly moving as their content changes from one image to the next; and it is precisely this characteristic which lessens their accessibility.

Criterion 2.2.2 of WCAG 2.1 basically states that any element that moves automatically must have a mechanism to enable users to stop, pause or hide such content. At the very least, then, users should at least be able to pause the carousel; however, the best solution would be to add control buttons to pause, fast-forward, and rewind content. Yet this is not always the case.

Accessibility must take into account all kinds of disabilities, and for some, the shortened times at which this information appears means that some users may need more time or have difficulties in processing the information while it is moving.

Another consideration when dealing with carousels is that the change from one slide to another means the content changes as well, and this needs to be announced by assistive technologies (e.g., screen readers). This is not a trivial matter as we need to clearly define when and how these announcements are made. Imagine that each time a slide advances, the screen reader announces this change, and since the carousel keeps moving, the user may be overwhelmed by unnecessary and repetitive messages. Therefore, new content should be announced when the user explicitly changes the slide, highlighting once again the advantage of adding carousel controls.

In practice, fully accessible carousels are very difficult to find. If you want to include a carousel on your website, remember to ensure the following:

1. They must have controls for stopping, starting, fast-forwarding, and rewinding, and these controls must be keyboard-accessible; ideally, they should not start automatically. This can be tested by using the keyboard to navigate around the website.
2. When the user explicitly changes slides, the screen reader is supposed to read the new content. The best way to check this is by navigating around the web page using a screen reader.
3. Remember that carousel content must also comply with other accessibility conditions (adequate contrast, correct text alternatives, links with a clear purpose, etc.).

Two of the tools described earlier are very useful for testing carousel accessibility: Web Developer Tools and Kontrast.

Multimedia

Multimedia content, and videos in particular, are another element commonly found on websites, mobile apps, and social media, since they communicate information in a manner that is attractive and easy to digest. A good video can convey information through images and sound, through what it says and what it doesn't say. Videos offer information using many channels. This is one of its greatest advantages. Yet at the same time this represents a disadvantage in terms of accessibility as it hinders access to this information for people with different types of disabilities.

That is why web accessibility initiatives such as the WCAG devote a significant portion of their guidelines to time-based media, including multimedia.

In general, multimedia can present difficulties for the following users:

- Blind people who cannot see the images.
- Deaf people who cannot hear the audio.
- People with mobility issues who find it difficult to operate playback controls.
- People with neurological disorders such as epilepsy, who are affected by certain images.

Without losing sight of the rest, it is typically the first two accessibility issues that are the most common.

To ensure that the majority of people can use and enjoy multimedia content (video, audio, or animated content), be it pre-recorded or live, the WCAG suggests the addition of the following:

- Captions
- Audio descriptions

- Transcripts

Captions

Captions provide audio information to people who are hard-of-hearing, although they are also very useful for other disability profiles, such as attention disorders, as they help users to follow what's being said.

Captions must be in sync with the audio.

The following guidelines indicate when captions should be included in order to fulfil WCAG 2.1 accessibility criteria:

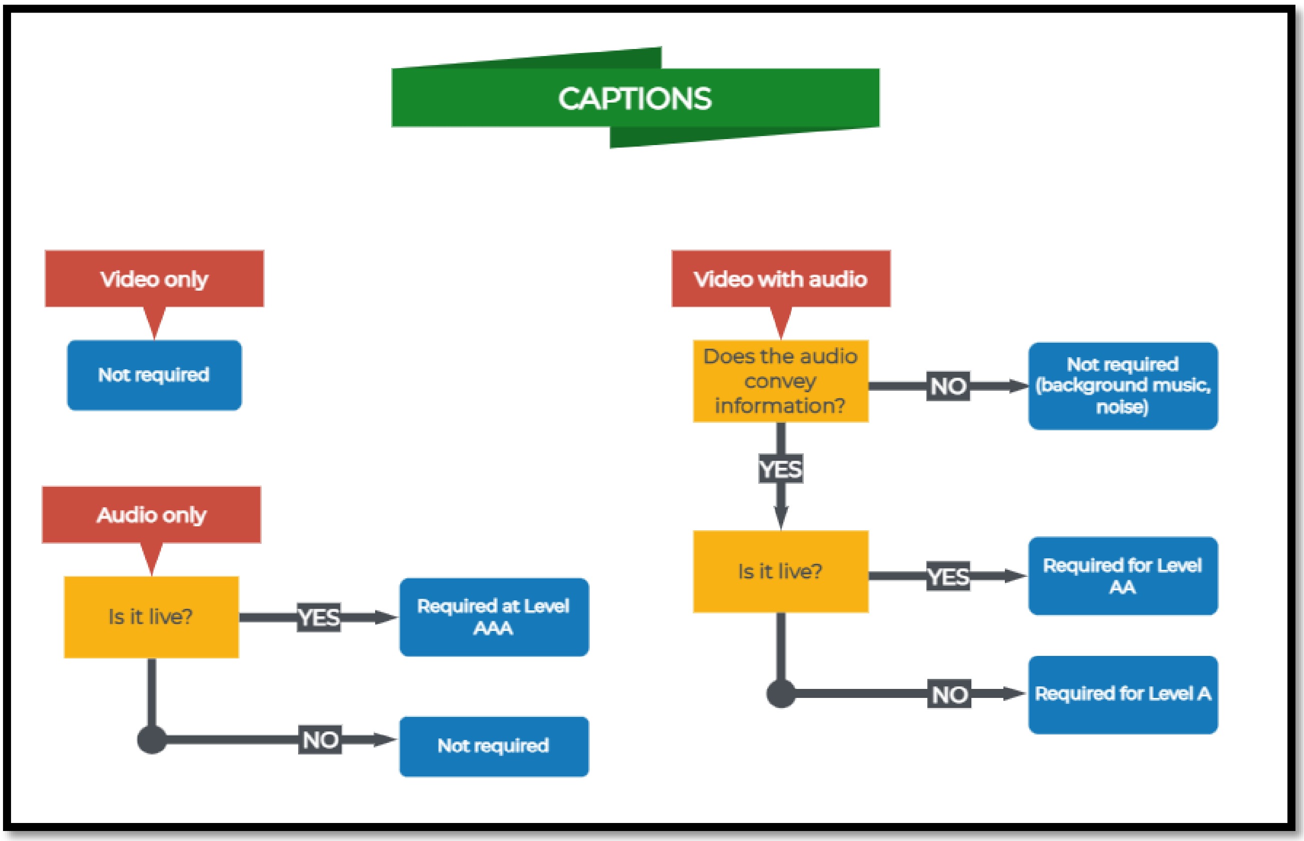


Illustration 27.Flow chart: when to use captions in multimedia content

If the content contains only video, captions are not required as there is no audio information to supplement.

For live audio, captions are required to achieve the maximum Level AAA but are not required for Level AA.

Finally, if both video and audio content are included, then captions are required, depending on the type of audio:

- If the audio does not provide information, as in the case of background music or random ambient sounds, captions are not required.
- If audio information is necessary, then captions are required (Level A), even for live broadcasts (Level AA only).

An important consideration is whether to use automatic captions such as those offered by YouTube, as they frequently contain transcription errors, and if they do, they are not acceptable. If they are used as support material, they must be manually corrected.

It is also important to bear in mind that captions are used not only to transcribe dialogues, but also have to indicate any sounds that influence the information, such as a laugh track in a sitcom.

Transcripts

A transcript is the written version of the audio portion of the multimedia content. Descriptive transcripts are also used. These include visual information and are similar to a script. Their characteristics help both users with impaired vision and those with hearing problems.

Unlike captions, transcripts do not have to be offered in sync with the audio/video but can be offered separately as long as the audio/video provides a link to the transcript. Obviously, the existence of a transcript significantly improves content usability for everyone and not just those with disabilities. That is why the use of platforms such as Etiqmedia, which provide transcripts and even live captioning, is widely recommended.

The following outline lists WCAG 2.1 criteria for including transcripts:

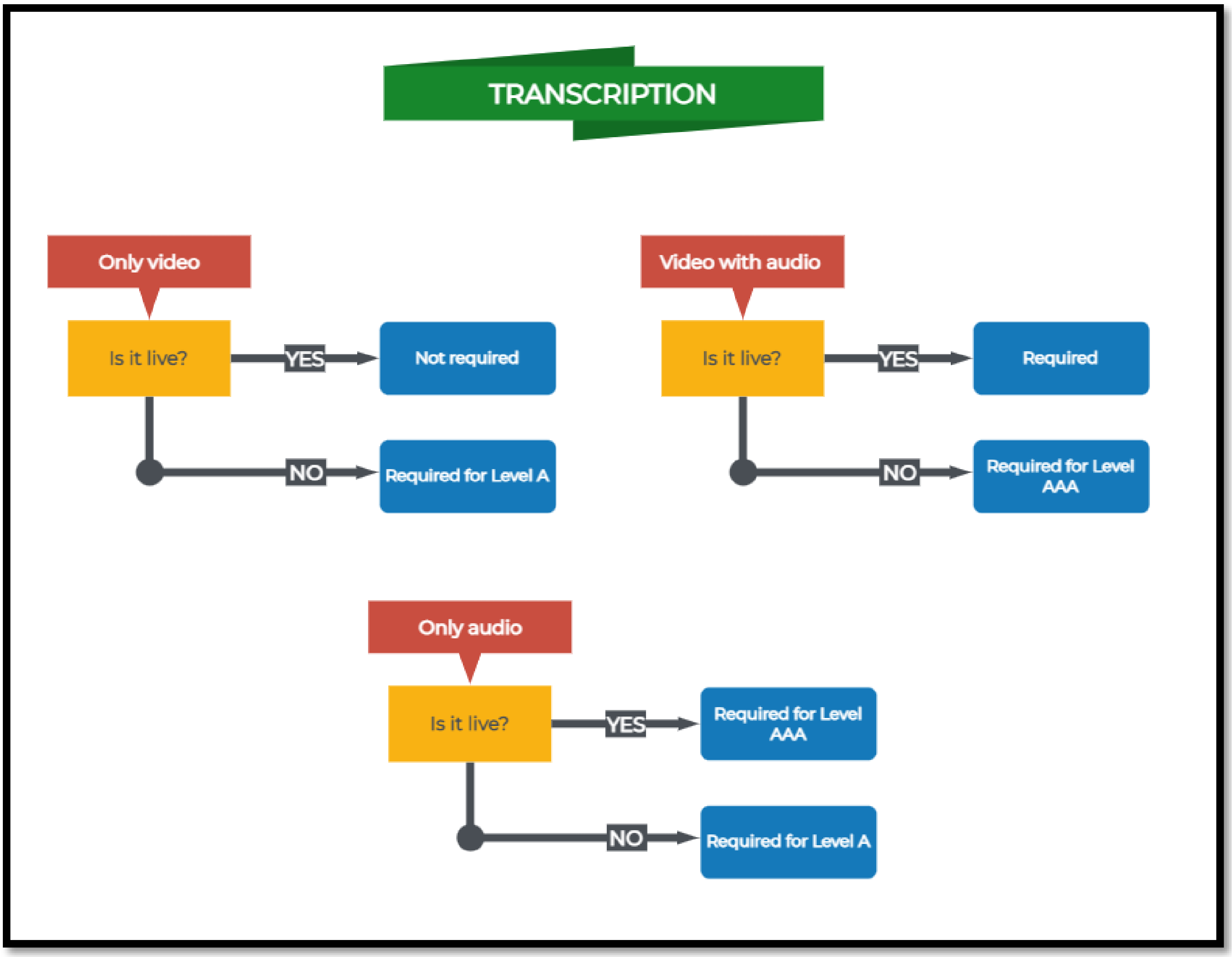


Illustration 28.Flowchart: when to use transcripts in multimedia content

- For video only, always use transcripts except when they are live.
- For audio only, always use transcripts, except when they are live; in this case, captions are only necessary for Level AAA.
- For video and audio, captions are not required if they are live and recorded, unless Level AAA is desired.

Audio description

This is a description provided in audio format on information pertaining to the multimedia content, including:

- Actions.
- Characters.
- Scenes.
- Body language
- Important text
- Other relevant visual content.

Like subtitles, audio description is provided in synch with the media work, which increases the difficulty in its implementation.

The situations in which audio description is needed are:

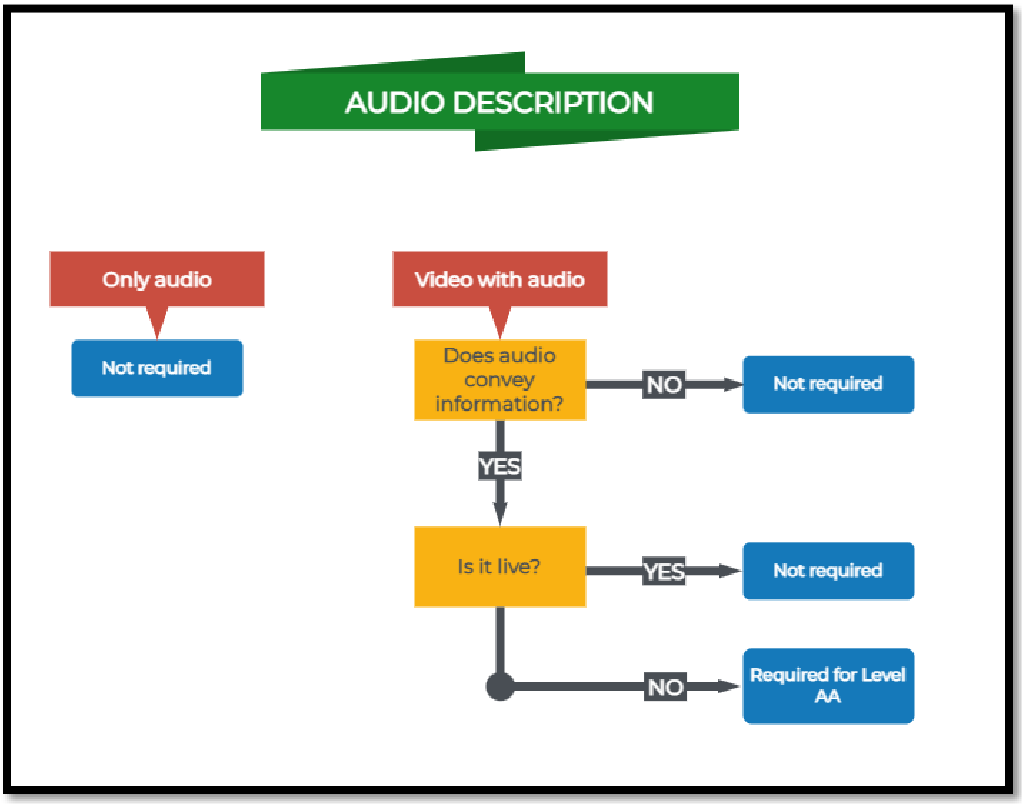


Illustration 29. Flow chart: when to use audio description in multimedia content.

- Description is not needed when the information in the video is not relevant to the message. For example, a video of a speaker at a conference giving a speech. Here, the only thing that is really important is the audio.
- It is not required for live broadcasts for the same reason transcripts aren't required: it is impossible to prepare them for live events where what is being said cannot be predicted.
- They are required for Level AA if the media work contains recorded video and the information is relevant. Using the same example as above, imagine a speaker who is gesticulating considerably in dramatic fashion. Here, the visual content is important to the message.

Therefore, for videos where attention has been given to both the visual and audio message, audio description is required at accessibility Level AA.

Implementing audio description is quite complex, both owing to its nature and technical aspects associated with this task. It entails the addition of a considerable amount of audio information, since it involves the narration of the aforementioned elements (characters, scene, atmosphere). This represents a lot of information which must also be synchronized with the video. This information is usually inserted during pauses in action; however, this may be insufficient given the short duration of pauses.

There are three situations where audio description can be used with video:

- The original recording already includes audio description: This solution usually works well for educational or training videos. In this scenario, all the necessary audio descriptions are included when the video is being produced. However, for other types of videos, this may not work, as the audio description interferes with the narration of the story.
- The necessary narration is added to the original story in between the gaps in the dialogue. As noted above, these pauses may need to be lengthened in order add this description, which in turn may disrupts the narrative pace of the video.
- A second audio track containing the audio description is added which can be activated or deactivated by the user. This does not affect the narrative pace as users are able to deactivate audio description if they do not need it.

None of these solutions are ideal, and to all of this we must also add technical limitations. While the inclusion of an additional track would be the recommended solution, many sites save their videos using the YouTube platform for various reasons, and YouTube currently only allows one audio track, ruling out the above option.

The only "simple" solution would be to offer two versions of the video: a "normal" version and an accessible one, including links to both. While this is not ideal, it is a good way to serve all users without spoiling the experience of any one type.

In any case, editing a video with audio description is a complex task that should be left to the hands of specialized professionals.

Colour

Colour theme and contrast

The WCAG 2.1 standards call for an exhaustive study of contrast, as this is what improves image perception and legibility of texts for people with low vision or colour blindness.

This is one of the most common issues in web accessibility and, given its importance, the WCAG is accordingly quite stringent on this aspect.

The contrast between the colour of the text and that of the background must at least be:

- A ratio of 4.5:1 for smaller fonts
- A ratio of 3:1 for larger fonts

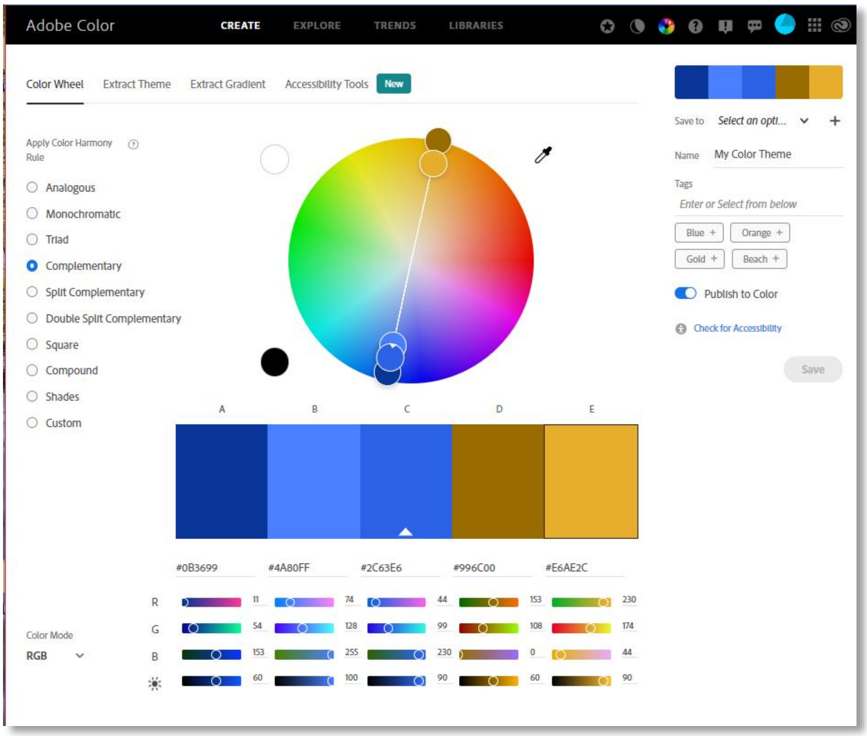
Large font sizes are those that are greater than 18 points or 14 if they are in bold.

The greatest colour contrast is that observed between black and white, whose ratio is 1:21.

That is why it is particularly important to choose a colour theme with the appropriate contrast ratios.

The choice of colour themes is usually handled by the graphic designer, although it is often conditioned by corporate colours.

To select the proper contrast, the use of Adobe Color is recommended. Described in the Tools section, this application allows you to select colour themes taking into account contrasts recommended by the WCAG:



Apply Color Harmony Rule: ● Complementary

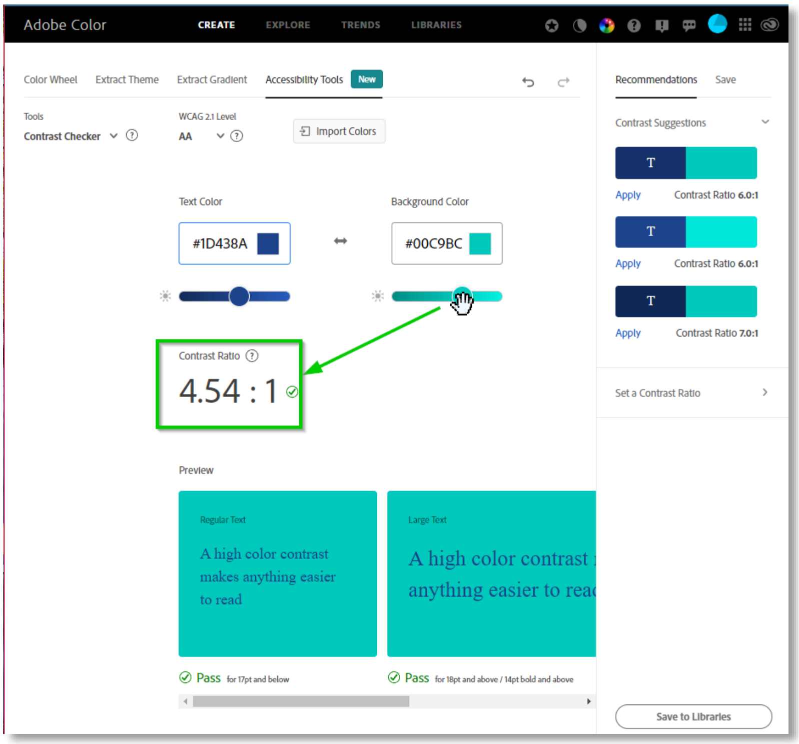
Name: My Color Theme

Tags: Enter or Select from below

	A	B	C	D	E
	#0B3699	#4A80FF	#2C63E6	#996C00	#E6AE2C
R	11	74	44	153	230
G	54	128	99	108	174
B	153	255	230	0	44
☀	60	100	90	60	90

Illustration 30. Complementary colours from Adobe Color.

Starting from an initial colour, Adobe selects a group of colours based on various criteria. Complementary colours are shown in the examples above, but what is important for us are the colours shown in the accessibility tab, as seen below:



Tools: Contrast Checker

WCAG 2.1 Level: AA

Text Color: #1D438A

Background Color: #00C9BC

Contrast Ratio 4.54 : 1 ✓

Contrast Ratio 6.0:1

Contrast Ratio 6.0:1

Contrast Ratio 7.0:1

A high color contrast makes anything easier to read

A high color contrast makes anything easier to read

✓ Pass for 17pt and below

✓ Pass for 18pt and above / 14pt bold and above

Illustration 31. Accessibility tab in Adobe Color.

In this section you can select your theme colours and see if they meet the minimum contrast requirements. If not, then you can vary your choice of colour to achieve a combination that does.

One last thing to say on this topic. The minimum contrast does not only apply to text elements; graphic elements used in the user interface must also have a minimum contrast of 3:1. Examples of graphic elements are buttons, icons, indicators of active elements, etc. In general, these include any graphic elements the user requires in order to operate or understand page operation.

Colour to convey information

One of the biggest errors that hinder website accessibility is the overuse of colour. It goes without saying that the strategic use of colour can make websites more attractive and can serve as an excellent resource owing to their subtle symbolic value. However, it is important to be very aware that colour may not be visible to a large segment of the population and as a result should not be used as the only means to convey information:

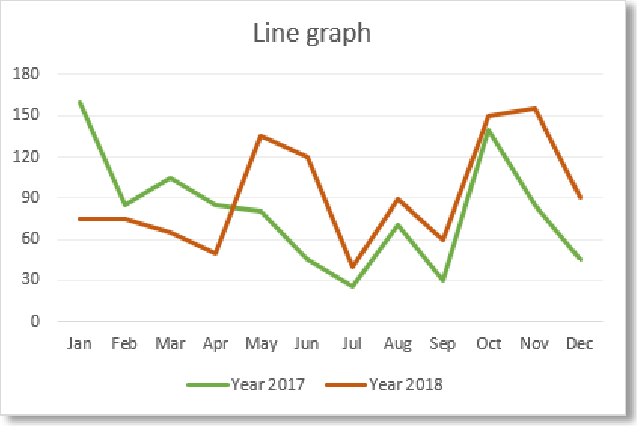


Illustration 32. Example 1: colour used exclusively to convey meaning

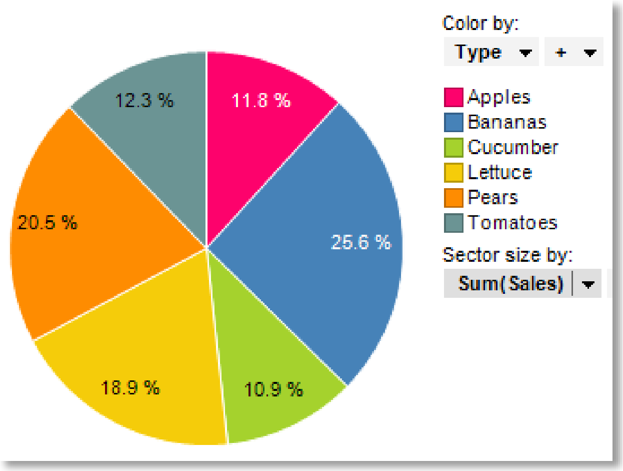
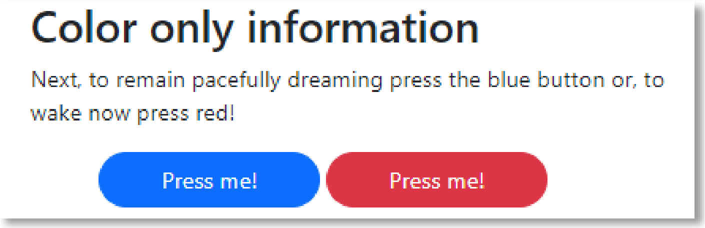


Illustration 33. Example 2: colour used exclusively to convey meaning.

In the example above, the data shown in the graphs can only be differentiated on the basis of colour.



Color only information

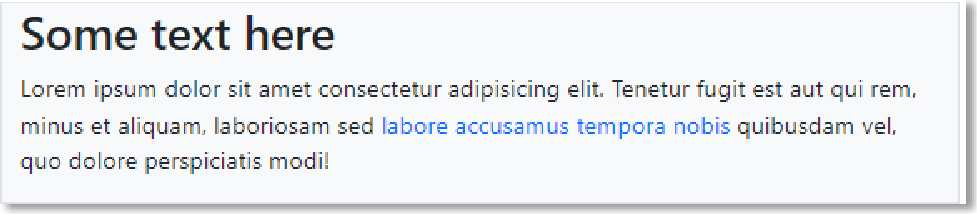
Next, to remain pacefully dreaming press the blue button or, to wake now press red!

Illustration 34. Examples of colour used exclusively in buttons.

In this example, colour is the only feature that differentiates the buttons.

Another example of using colour to convey information is the use of a different coloured font to indicate a link within a paragraph.

Let's look at the following example:



Some text here

Lorem ipsum dolor sit amet consectetur adipisicing elit. Tenetur fugit est aut qui rem, minus et aliquam, laboriosam sed labore accusamus tempora nobis quibusdam vel, quo dolore perspiciatis modi!

Illustration 35. Example of a link differentiated only by colour.

In the above image, the link https://dipcas.sedelectronica.es stands out only because a darker font colour has been used to distinguish it. It would be better, for example, to underline it.

The easiest way to avoid this problem is to use a second visual clue, such as different types of line or fill for the graphs:

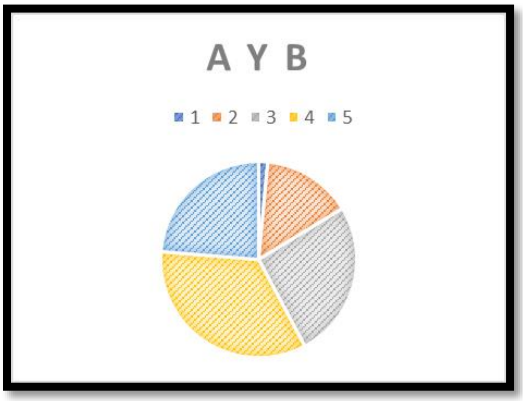


Illustration 36. Example 1: graph with a second visual clue, apart from colour.

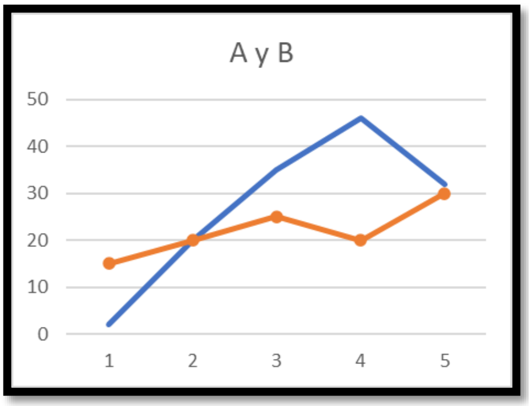
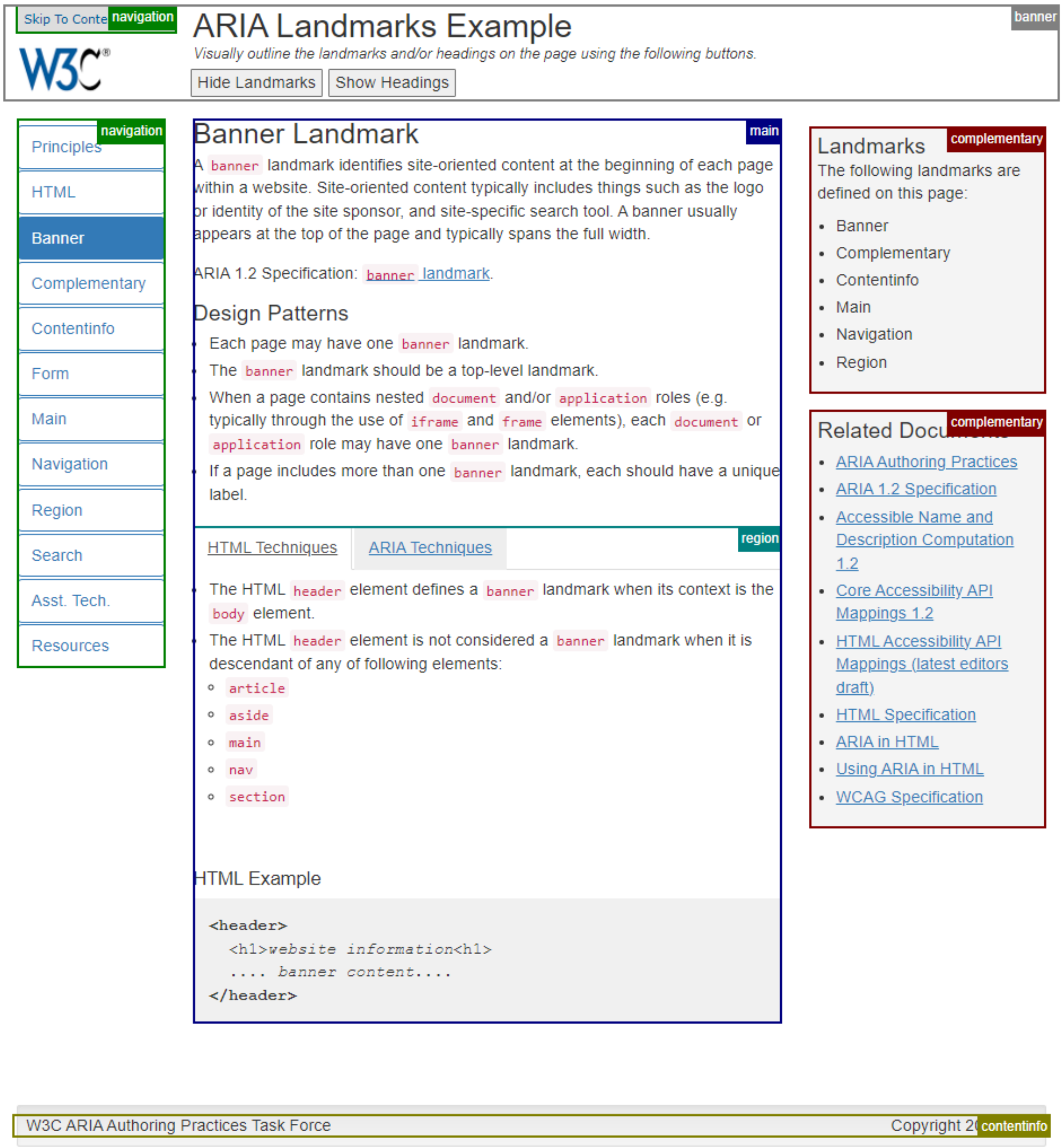


Illustration37. Example 2: graphs with a second visual clue, apart from colour.

In the case of links, it is standard to use a blue font (use of colour) and underlining (additional clue).

You can always add another element; a well-chosen clue will make the website even more accessible without losing the attractiveness that colour brings.

Structure, the foundations of accessibility



ARIA Landmarks Example

Visually outline the landmarks and/or headings on the page using the following buttons.

Banner Landmark

A `banner` landmark identifies site-oriented content at the beginning of each page within a website. Site-oriented content typically includes things such as the logo or identity of the site sponsor, and site-specific search tool. A banner usually appears at the top of the page and typically spans the full width.

ARIA 1.2 Specification: `banner` landmark.

Design Patterns

- Each page may have one `banner` landmark.
- The `banner` landmark should be a top-level landmark.
- When a page contains nested `document` and/or `application` roles (e.g. typically through the use of `iframe` and `frame` elements), each `document` or `application` role may have one `banner` landmark.
- If a page includes more than one `banner` landmark, each should have a unique label.

The HTML `header` element defines a `banner` landmark when its context is the `body` element.

The HTML `header` element is not considered a `banner` landmark when it is descendant of any of following elements:

- `article`
- `aside`
- `main`
- `nav`
- `section`

HTML Example

```
<header>
  <h1>website information<h1>
  .... banner content....
</header>
```

Landmarks

The following landmarks are defined on this page:

- Banner
- Complementary
- Contentinfo
- Main
- Navigation
- Region

Related Documents

- ARIA Authoring Practices
- ARIA 1.2 Specification
- Accessible Name and Description Computation 1.2
- Core Accessibility API Mappings 1.2
- HTML Accessibility API Mappings (latest editors draft)
- HTML Specification
- ARIA in HTML
- Using ARIA in HTML
- WCAG Specification

Illustration 38. Example of website structure.

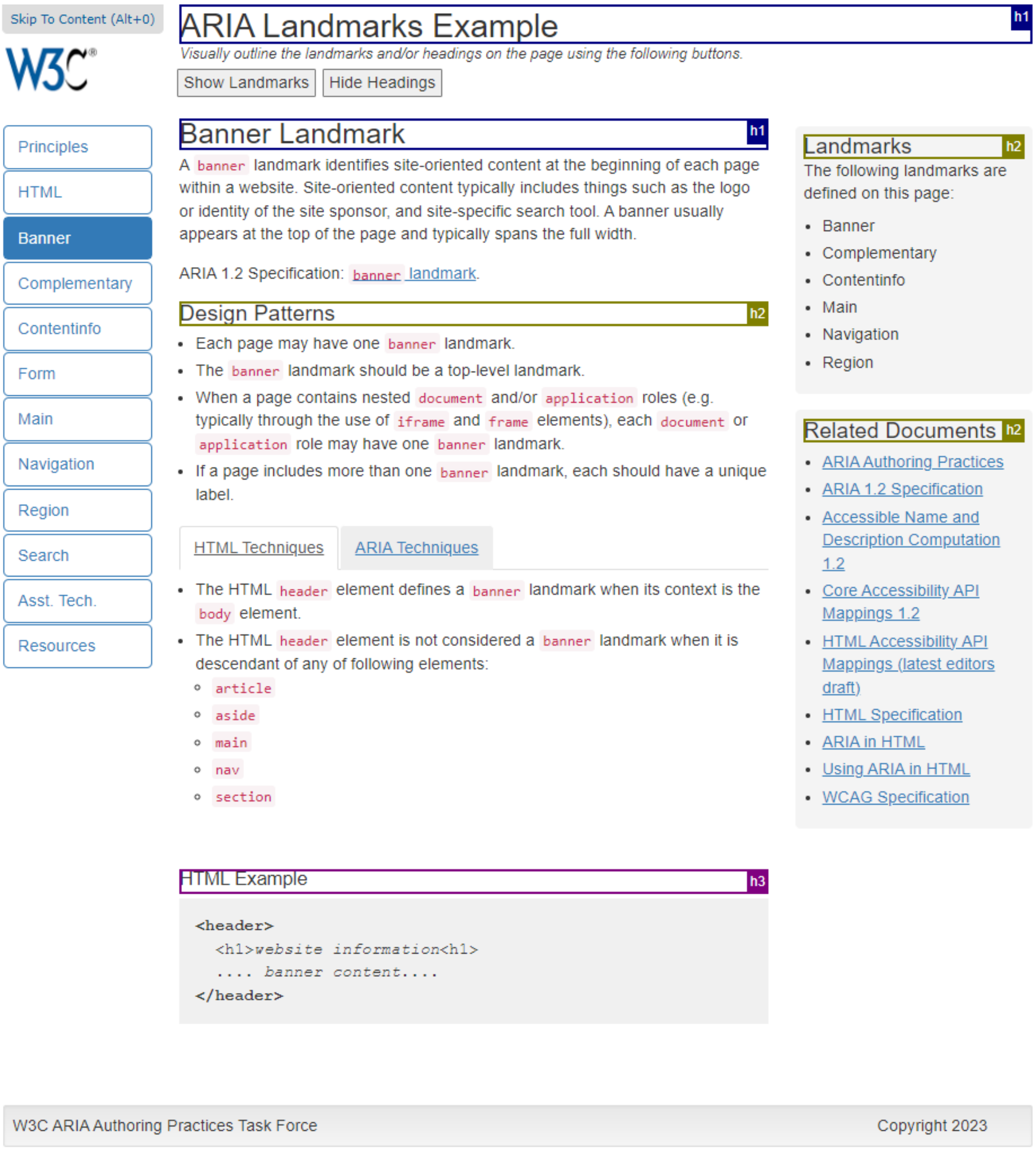
Web pages and mobile apps whose design is neat, clean, and consistent are more attractive and easier to use. On a well-constructed site, areas are distributed according to function. These areas may offer navigation menus, main website content, additional content, news, highlights, etc.

In addition to improving the appearance of the page, a well-structured page will facilitate the work for assistive technologies. A web page that has many unnecessary elements or is poorly organized can significantly hinder accessibility. For someone who is blind, the experience of navigating around a web page with 200 haphazardly placed links is not the same as navigating around one where the same links have been grouped into areas that are easy to distinguish owing to correct tagging. In the latter case, users only need to locate the area they are interested in, and search among the few links found there.

Since a well-organized structure is key to adequate web accessibility, it is a good idea to pause and think about how to best organize web page content. Each page must be properly divided into areas or regions, each with a label that best describes its content.

The content found within each area must be equally tidy: the various pieces of information must be easy to find and correctly labelled.

A good way to label and distinguish sections is by using appropriate headings, remembering that these should have a hierarchical structure. We are all familiar with text editors, which offer different headings, each numbered differently (Heading 1, Heading 2, etc.). A similar feature exists in web languages; following a hierarchy is important for accessibility because it helps to keep web page content organized.



ARIA Landmarks Example

Visually outline the landmarks and/or headings on the page using the following buttons.

Banner Landmark

A `banner` landmark identifies site-oriented content at the beginning of each page within a website. Site-oriented content typically includes things such as the logo or identity of the site sponsor, and site-specific search tool. A banner usually appears at the top of the page and typically spans the full width.

ARIA 1.2 Specification: `banner` landmark.

Design Patterns

- Each page may have one `banner` landmark.
- The `banner` landmark should be a top-level landmark.
- When a page contains nested `document` and/or `application` roles (e.g. typically through the use of `iframe` and `frame` elements), each `document` or `application` role may have one `banner` landmark.
- If a page includes more than one `banner` landmark, each should have a unique label.

HTML Techniques | ARIA Techniques

- The HTML `header` element defines a `banner` landmark when its context is the `body` element.
- The HTML `header` element is not considered a `banner` landmark when it is descendant of any of following elements:
  - `article`
  - `aside`
  - `main`
  - `nav`
  - `section`

HTML Example

```
<header>
  <h1>website information<h1>
  .... banner content....
</header>
```

Landmarks

The following landmarks are defined on this page:

- Banner
- Complementary
- Contentinfo
- Main
- Navigation
- Region

Related Documents

- ARIA Authoring Practices
- ARIA 1.2 Specification
- Accessible Name and Description Computation 1.2
- Core Accessibility API Mappings 1.2
- HTML Accessibility API Mappings (latest editors draft)
- HTML Specification
- ARIA in HTML
- Using ARIA in HTML
- WCAG Specification

W3C ARIA Authoring Practices Task Force Copyright 2023

Illustration 39. Structure example.

Typically, the Level 1 heading has a larger font than the Level 2 heading, and so on. Some people use this option to format the texts: if they want a larger text in bold, they give it a header style.

This was common practice before now, when the same language was used to insert content and to format it. However, this encourages the heading hierarchy to continue to be used based not on the importance of the heading but on font size. This is not an appropriate way to use headings and should therefore be avoided. It is best to always respect header hierarchy. Headings are one of the navigation methods used by screen readers and missing or disorganized levels can confuse users.

To conclude, here is a brief summary of key aspects to bear in mind in relation to web page structure:

- Be organized.
  - Group content according to function.
  - Label and use headings for key content.
  - Remember to ensure consistency in heading hierarchy.
  - Distribute content in keeping with window sizes.
- Less is more: leave sufficient space between elements.
- Clarity and simplicity.
  - Use simple icons.
  - Include instructions if you think they are necessary, it’s better for users to ignore them if they are unnecessary than to have users who don’t know what to do.
- Be clean.
  - Don’t use small font sizes.
  - Leave adequate spacing in the text and between paragraphs.
  - Automatic audio or video playback is very invasive, do not add this feature.
  - Beware of pop ups that hide content.
- Separate content from appearance
  - Structure the content using HTML tags to ensure good organization.
  - Use CSS to make your content attractive.
  - Don’t mix the two: don’t insert content using CSS and don’t decorate using HTML.

2.2. Operable

Web pages, are, almost by definition, interactive: of the content presented to the user there are controls and links that the user can select to personalize their browsing experience.

The technology underlying websites allows for many users’ interaction options. Most of us use a mouse or a trackpad and the keyboard is sometimes used when entering information, but not everyone is able to navigate this way.

The most obvious case is that of blind people who cannot use a mouse or touch screens, or at least not in the usual way. But there are also people with mobility issues who cannot even use keyboards and others with cognitive disabilities who require more time to understand and react to content.

The aim of the Operable principle is to ensure that everyone is able to interact with the Web, regardless of the methods they use. Normally all the recommendations in connection with this aspect are mainly aimed at web developers, but it is important for everyone involved in this task to be aware of operability needs and requirements.

Here is a brief summary of what is needed to ensure website operability.

Keyboard

All links, controls, and interactive elements on the page should be keyboard accessible, usually using a few keys such as the tab, arrow, enter, and space keys.

If desired, certain functions can be performed using keyboard shortcuts. It is important to always use keys or key combinations that do not interfere with data input.

Regarding keyboard navigation, always ensure that there are no keyboard traps, that is, areas that users can enter but cannot leave. This could occur with complex elements that open when they receive keyboard focus but cannot be closed. The user is stuck there, and they cannot continue. A common example is a chatbot, which can be accessed with a button, but then the Close button cannot be accessed because it never receives the keyboard focus.

Time

As stated above, website operability not only depends on the interface that is used. It is also necessary to take into account users who may need more time to assimilate content or to react to it. We have already seen an example of this in the use of carousels, which required a mechanism to stop carousel movement.

This is also true of other time limits. For instance, the use of an online form to access certain functions: if there is a time limit to respond, we may be leaving out those who are unable to respond as quickly because of their disability.

In this sense, the WCAG is clear: if a time limit is placed on content, you need to meet at least one of the following conditions:

- Users are able to turn off the time limit.
- Users can extend this limit at least ten times.
- Users are given a warning at least 20 seconds before the limit is over so that they can delay it by means of a simple action (for example, pressing the space key); users should be able to do this at least ten times.

The exception to this is when the time limit is essential and extending it would invalidate the activity. For example, a timed quiz.

Another option is to set a very long time-limit (more than 20 hours). This would remove any obstacles to accessibility.

Navigable

We already know that a website must be keyboard accessible and that we need to give users enough time to interact with the content. It is also essential to make it easy for users to find the content that interests them, facilitating navigability.

A distinction can be made between making it easier for users to find content and making it easier for them to distinguish content.

Navigation shortcuts and assistance

It is common to find websites with a lot of information and many pages, which multiplies the number of possible options for users. We are accustomed to identifying the parts that interest us at a glance and go directly to them, regardless of the number of sections and menus a website may contain.

This is easy for users who can use a mouse and touchscreen, but not for those who find it difficult to use such interfaces.

For a moment imagine that you can only use your keyboard to navigate around a page. In most cases, to access a given site, you need to use the tab key to jump from element to element to get to the item of interest, for example, an article found in the main content of the web page. Like most pages, the page you are visiting has a menu at the top of the page consisting of various options. By the way they are constructed, navigation through menus takes place option by option, in sequence. To get to the main content you need to run through the entire menu and perhaps other items until you find the item in question. Once there, you will still have to search for the article that interests you.

Facilitating and optimizing web page navigation is another essential objective that we need to achieve. A number of recommendations are provided in this regard.

The first is to provide links that skip over repetitive navigation elements such as the aforementioned menus, social media, or news blocks. At the very least, WCAG recommends that a link be included at beginning of a web page which leads to the page's main content so that users do not have to move from element to element.

The second recommendation that helps users to locate content is to group it into areas. This task is usually entrusted to a web developer. Using the appropriate labels, they are able to indicate where the main content, navigation area, and additional content, etc. are located. Unfortunately, there are some situations where this is not possible, for example if you are using a CMS (Content Management System) over which you have almost no control.

Finally, it is also useful to include links to all or the most important pages or sections in a central location on the website. This would help users find the content they are looking for. This is commonly done by way of a prominently displayed link on each page, which takes users to a map of the website. That way, users will be able to search for the topic of interest to them using the map, and clicking on the link of interest directly, instead of having to search through the menu options.

Improve content visibility

Although it is important for users to able to reach content, being able to identify it is equally important. Most users are able to visually identify content naturally, either through headings written in larger fonts that stand out from the text, frames placed around content, or through the use of colour and backgrounds.

Areas having the same function should look the same throughout the website: if a menu is included on one page, it should be immediately recognizable as such, and when another page is opened, the menu should look the same and appear in the same page position. The same should be true for headings and footers.

For users, seeing what they are doing is also important: being able to access all of the website's parts serves no purpose if there is no way of knowing where they are located at any given moment. The keyboard focus should always be visible.

Input methods

To operate a website, we also need to think about how interaction takes place using controls and input methods.

In accessibility, interaction must be predictable. Just entering a control shouldn't mean that the user is activating that action. For example, when entering a control that says, "Go to page...", this control should not take the user to that page if they have not confirmed such an action by pressing the Enter key.

Ambiguity should also be avoided: controls should always be well-labelled, their purpose clear, and should not force users to investigate whether the control does what they think it does.

2.3. Understandable

User interface information and operation must be understandable. This is only reasonable. In this section we will take a look at what makes interfaces understandable.

Page language

This is sometimes so obvious that it is taken for granted.

A main language must be defined for all pages (and can be extended to all documents).

Apart from indicating the website's main language, language changes must also be indicated. If, for example, a quote in French is included in an article found on the website, it is necessary to indicate the language change at that point.

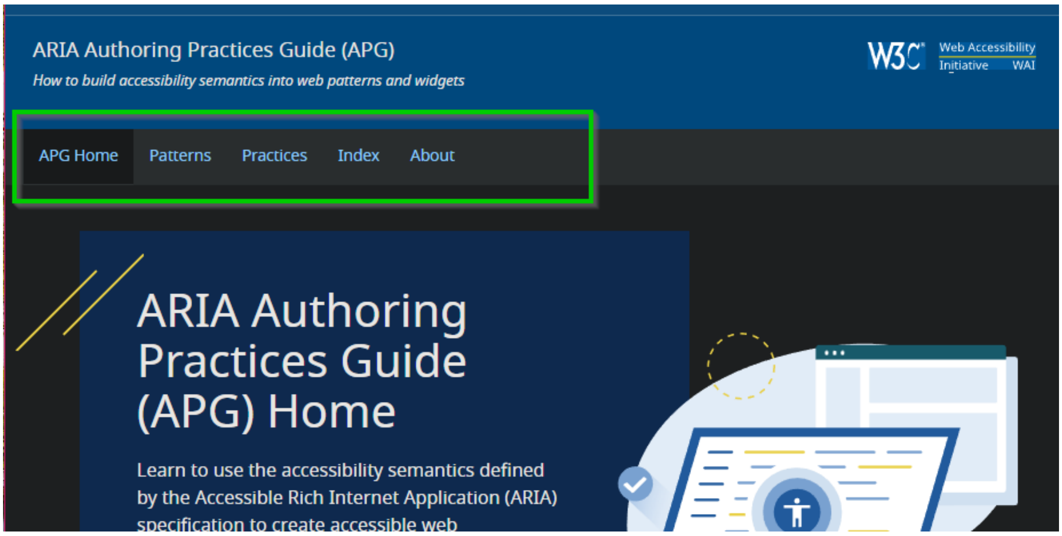
Indicating a change in language is mandatory as long as it is not an invented language, or the expression is a part of the vernacular of the language of the website. If, for example, the word *software* is used on a Spanish website, there is no need to indicate a change of language because it is a widely used word.

Appearance and placement of common elements

All web pages feature common elements that are repeated throughout the site, such as menus, buttons, the footer, the news section, etc.

These elements are of special interest to users and essential to the proper operation of the page. Being able to immediately identify these elements improves user experience, especially for disabled users who find it difficult to navigate website content.

In order for these components to be recognized quickly their appearance and location on the page must remain consistent throughout the website.



ARIA Authoring Practices Guide (APG)

How to build accessibility semantics into web patterns and widgets

ARIA Authoring Practices Guide (APG) Home

Learn to use the accessibility semantics defined by the Accessible Rich Internet Application (ARIA) specification to create accessible web

Illustration 40. Main menu of ARIA Authoring Practices.

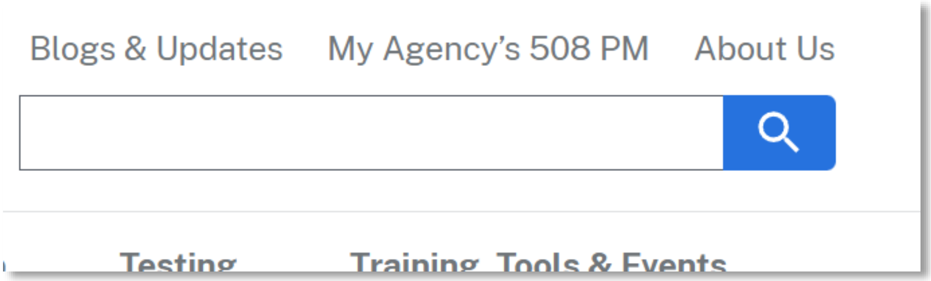


Illustration 41 Section 508 search.

Common website elements are shown in the two examples above: they feature elements that look the same and probably appear in the same position on all pages, making them easily recognizable.

Forms: assistance and error control

One of the most convenient functions of the Internet is that it allows users to carry out tasks remotely, and a fundamental tool in this regard are online forms. They allow us to carry out a multitude of tasks, ranging from filing income tax returns to requesting information from a vendor.

All of these procedures performed using online forms are tremendously useful, so it would be unfair to leave out an important part of the population. When it comes to government services, this is especially important, and therefore we need to ensure accessibility for online forms as well.

Again, it is the website developer who is in charge of how an online form is designed. First, let's take a brief look at the standards a form must follow for it to be accessible.

Labelling. It doesn't matter who's is looking at the form; if its fields aren't properly labelled, there's a good chance that a user won't know what to put in them. For blind individuals, if there are no labels explicitly associated with the field, they might as well be crossed out.

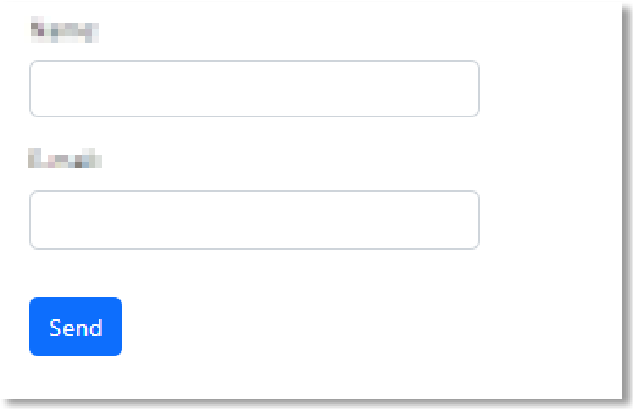
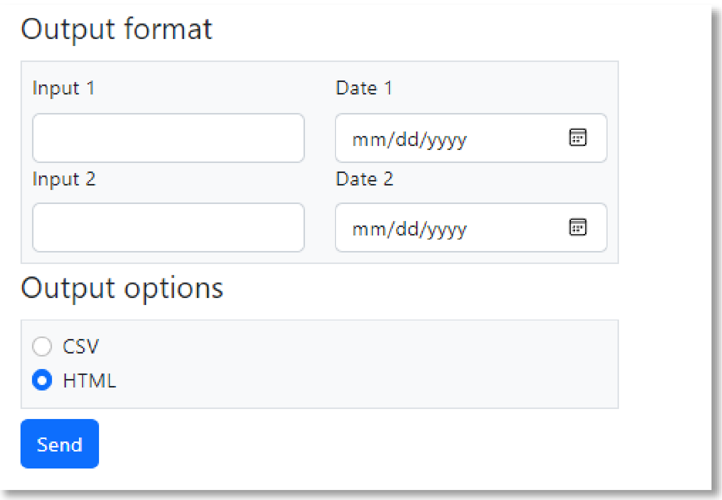


Illustration 42. Labelling in online forms.

Grouping together of related controls. Fields and controls that are interrelated must be grouped together. This makes the form more readable, as related fields are easier to locate, and the form is organized into smaller, more manageable elements.

Forms must be accompanied by instructions to help the user to complete it, for example:



Output format

Input 1

Date 1 mm/dd/yyyy

Input 2

Date 2 mm/dd/yyyy

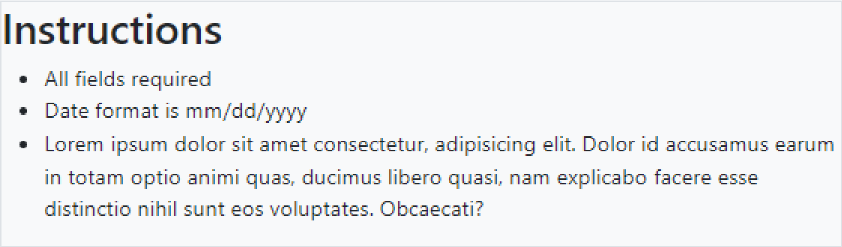
Output options

○ CSV

● HTML

Illustration 43. Instructions on forms.

General instructions. Somewhere on the form there should be instructions on how to fill it out and indications as to the form's required fields and details.

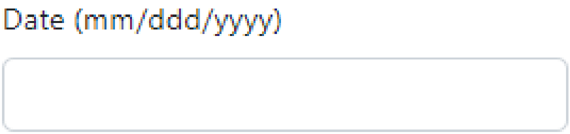


Instructions

- All fields required
- Date format is mm/dd/yyyy
- Lorem ipsum dolor sit amet consectetur, adipisicing elit. Dolor id accusamus earum in totam optio animi quas, ducimus libero quasi, nam explicabo facere esse distinctio nihil sunt eos voluptates. Obcaecati?

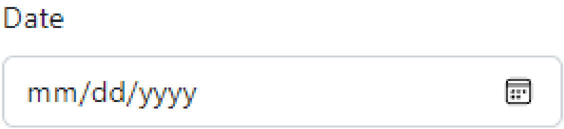
Illustration 44. Example of instructions for completing an online form.

Online Instructions. Both labels and placeholders can be used to add short instructions:



Date (mm/ddd/yyyy)

Illustration 45. Example of instructions within the field label.

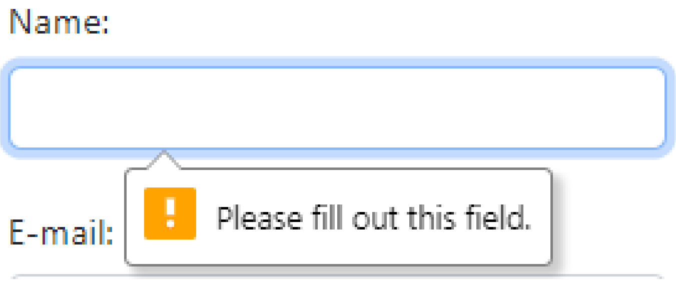


Date

mm/dd/yyyy

Illustration 46. Example of instructions as placeholder.

Validation and error detection. In addition to having properly labelled forms with clear instructions, we also need to support users through error detection and data input support. The online forms should be able to detect incorrect data input as much as possible. If errors are detected, it should suggest how to correct it:

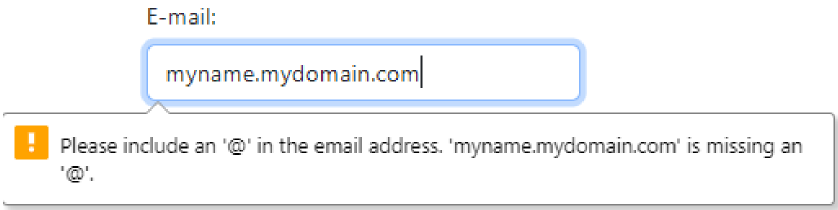


Name:

E-mail:

Please fill out this field.

Illustration 47. Example of an error: empty required field.



E-mail:

myname.mydomain.com

Please include an '@' in the email address. 'myname.mydomain.com' is missing an '@'.

Illustration 48. Example of error detection and suggestion for correct input.

Notifications. Once the form has been completed and submitted, the user must know if the submission has been successful or not: this includes both the result of form submission and any error messages such as those discussed above. Of course, all notifications must also be accessible, so just displaying a message or having the controls turn red is not valid. The previous screenshot shows the standard error messages shown by default. These messages are accessible and therefore screen reader will be able to detect such messages when they occur and notify the user of them. If a different type of notification is desired, it should behave as above.

Simplicity. One last recommendation in connection with online forms: they should be as simple as possible. For example, if a lot of data input is required, this should be broken down into multiple forms to make it easier for users.

Avoid lengthy lists of items to choose from. For example, the province of Castellón has 135 municipalities: a drop-down menu showing that many options are unwieldy and even more so if you have to listen to all of them until you reach the desired one. In these cases, a filter should be applied beforehand to reduce the number of options on the list. Remember that in terms of accessibility, actions that are easy to perform with a mouse and on-screen cursor may not be that easy if you have to use a keyboard and a screen reader.

2.4. Robust

The last accessibility principle is robustness, whose goal is to ensure website coding that is robust so that it can be interpreted by both current and future assistive technologies. This accessibility topic is the one that has the most to do with website development, so only the most important aspects and their effect on web accessibility will be mentioned here.

When a website is built, it is done using a markup language. The tags tell the browser how to interpret the data. Let’s take a look at a couple of simple examples; a paragraph which is marked with an opening paragraph tag and a closing paragraph tag will be interpreted as a paragraph:

```
<p> In a place of the spot whose name I don't want to remember...</p>
```

A Level 1 heading also has its own tag:

```
<h1>DON QUIXOTE OF LA MANCHA</h1>
```

This tag not only indicates that it is a heading, but also puts it at the top level of the heading hierarchy.

All the tags and components that are used to build a page are made with tags of this type. Their behavior, function, and characteristics are described in a standard regulated by the W3C.

If we observe the standard’s specifications, the web pages we build will comply with the Robust principle and assistive technologies will know how to interpret website elements. The W3C standard indicates the characteristics a component must have so that assistive technologies will be able to find them. For example, the function of links is to take users to another site, but they also contain text. This text is what screen readers look for to announce the link destination. If we fail to follow this standard and add a link without text, such as a link consisting of only an image, the screen reader does not know what to announce and accessibility breaks down.

The same goes for non-standard web components. They can still be used, but they must have the same characteristics as standard components. Unfortunately, non-standard components lacking accessibility support are all too common.

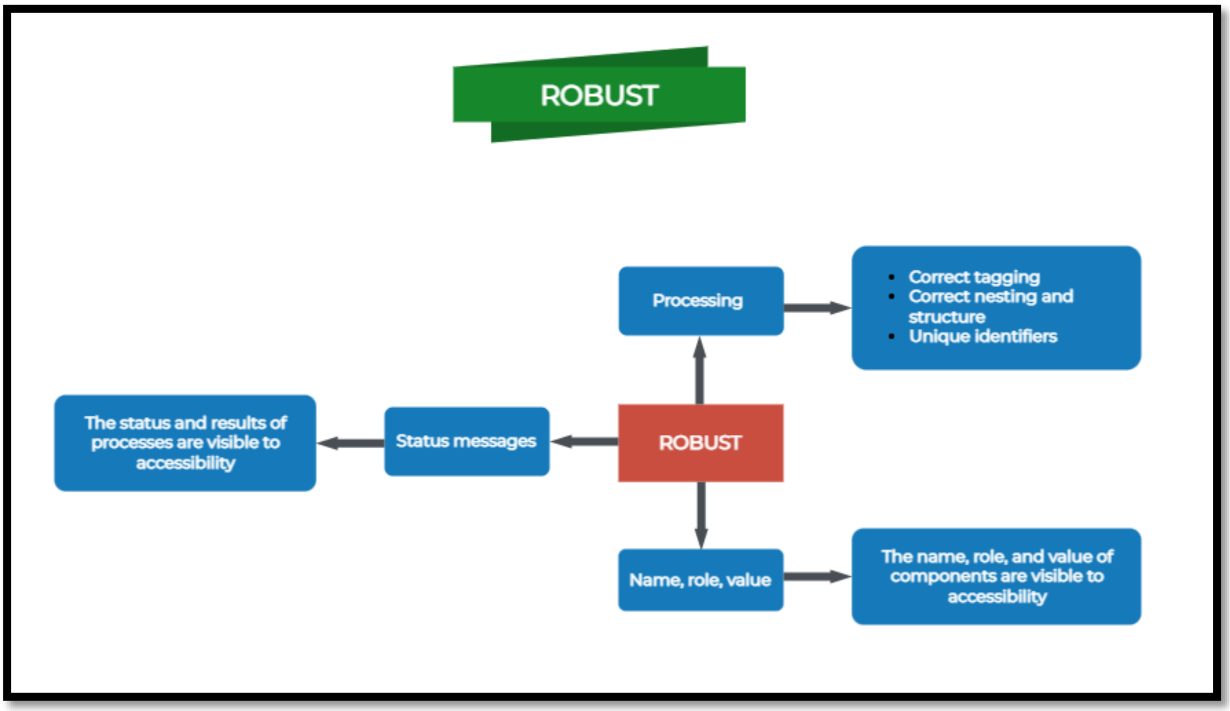

ROBUST
Processing
Correct tagging
Correct nesting and structure
Unique identifiers
The status and results of processes are visible to accessibility
Status messages
ROBUST
Name, role, value
The name, role, and value of components are visible to accessibility

Illustration 49. Diagram indicating main characteristics of the Robust principle.

3. Accessible documents

Many types of content may exist on a website (information, images, applications, etc.) and documents are just one type; these include downloadable documents and those that can be viewed on the page itself. In both cases, the users who access these documents are the same as those who visit the web; this means that they may also have the same accessibility needs. The documents should also be produced following accessibility criteria. These are essentially the same as those that apply to web pages.

As mentioned above, a document is subject to the same criteria as a web page, that is, the colours it uses must have sufficient contrast, the images must have a text alternative and the document must be structured so that it can be recognized by assistive technologies.

The vast majority of documents found on web pages are PDFs. But if you have ever handled a PDF editor like Acrobat (not Acrobat Reader, but Acrobat Pro) you will have realized that although it is described as a PDF editor, it is not designed to create documents. At most it allows you to modify or add text or images, but it is not designed for document creation, and much less for document layout.

Accessible documents can only be created using an editor specifically oriented to the creation of the document type in question. In most cases, the document need not be any more complex than a Word document or a similar type; for more complex documents requiring layouts that cannot be created in Word, the use of other more sophisticated editors may be necessary, for example, Power Point, Publisher for brochures, Corel Draw or Illustrator for posters, and Art Design or Indesign for more complex layouts. There is a wide array of options.

This guide will examine the most common case: the creation of an accessible PDF from a Word document, which is the *de facto* standard. Other editors, in the form of free or commercial software, all have similar functions, and therefore every option used below is also available with these.

It is important to consider the following aspects when creating an accessible document:

- Structure
- Text alternatives
- Use of colour

3.1. Structure

Although users are unaware of it, Word documents also use a tagging language that is somewhat similar to the one used in the creation of web pages. But unlike web pages, word documents do not need to be coded since the editor already does this for us. However, it is still necessary to use editing options correctly in order to create a proper document.

In a text document, there are headings, which are a part of the text that we want to highlight. The Word menu offers special options to do this: today, people should no longer be creating headings manually.

Although text editor options and tools facilitate document creation, they also have a second function that is not as obvious to us. When the heading option is used, not only does the appearance of the text change, but we are also giving it a tag. That is, we are classifying it as a heading and this information stays there, reflected in the markup language mentioned earlier in this guide, which assistive technologies are able to read and interpret. When the Word options are used properly, these tags are automatically included in the document, representing the basis of many accessibility features in the document.

Current text editors include many of these options that aid in text creation, and if they are used correctly, we will be able to create accessible documents. It's that easy: if the aim is to create an accessible document, forget about doing certain things manually. Do not create headings from scratch, use the recommended heading styles in the menu. If you don't like the style, you can modify it.

The same is true for tables - never use the tab key to create one (use the “Insert table” option instead) and lists should be created using the list option (no matter how simple your list is, never use hyphens to create one).

Remember that if you create an effect or text format without using an option from the menu, any assistive technology will not be able to distinguish this text from the rest because there is nothing to tell it otherwise.

3.2. Text tagging in Word

Let’s take a quick tour of the most important Word options, which in addition to text formatting, also adds the underlying tags.

Headings:

In Word they are called “Headings” or “Title”. Remember that they are hierarchical, and you should not alter this order or skip levels. When you include titles, you are also including tags that can be used to search for content in different parts of the document. Additionally, headings create a markup navigation menu that will then carry over to the PDF when you export it. This markup navigation is important because it is a requirement of the WCAG for documents containing more than 20 pages:

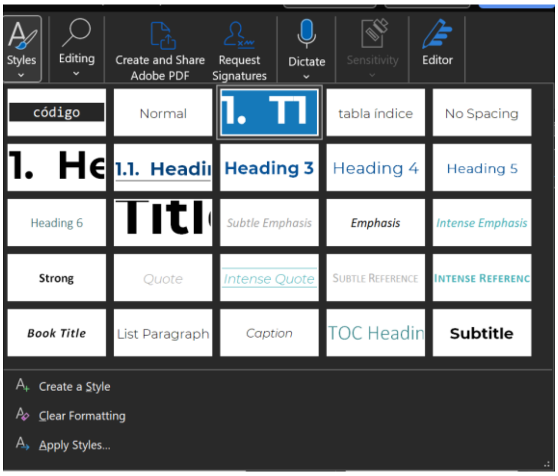


Illustration 50. Headings or titles in Word.

To see this navigation menu, simply open the View tab and select the “Navigation Pane” checkbox: a side pane will open allowing you to go to specific document content, based on the headings.

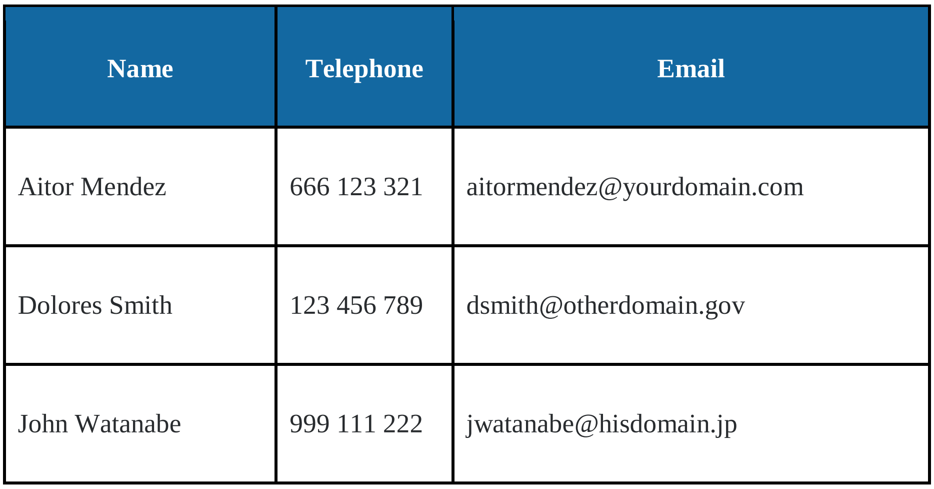
Lists

All text editors allow for the creation of at least two types of lists: bulleted and numbered. In the first list type, a bullet is used to indicate each item of the list (this is a bullet by default), but any symbol can be used. In the second type, instead of bullets, some other type of indicator is used (numbers, letters, etc.). As with headings, lists modify the appearance of the text, but when they are created, that part of the document is tagged as a list. This means that from the accessibility standpoint, important data is stored internally, such as the total number of elements and the position each one occupies.

Thus, a screen reader will be able to indicate that it is reading item six of a list of fifteen items. Thus, what we perceive visually can be conveyed through an alternative process.

Tables

If a table is going to be used in a document, designers prefer to use the table option, as it makes table creation much easier. With respect to accessibility, there are two types of tables in terms of function: those used for document layout and those used to display data.

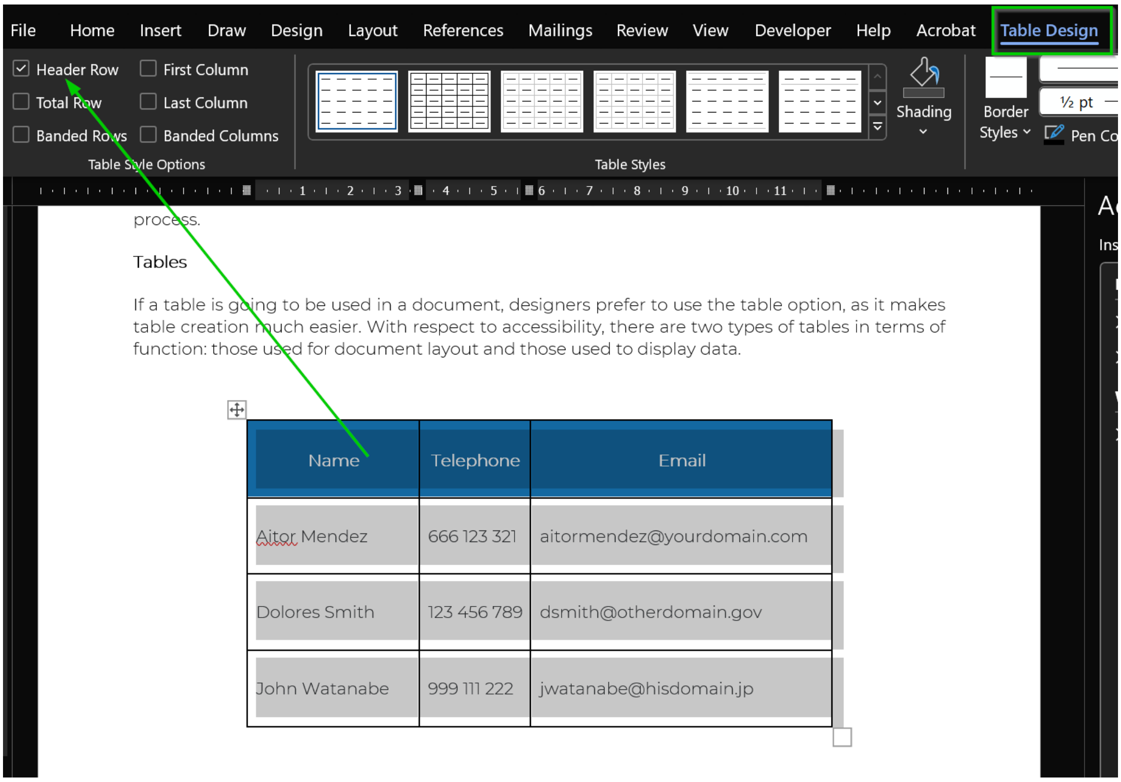


Name	Telephone	Email
Aitor Mendez	666 123 321	aitormendez@yourdomain.com
Dolores Smith	123 456 789	dsmith@otherdomain.gov
John Watanabe	999 111 222	jwatanabe@hisdomain.jp

To be accessible, the table must have headers, which serve as tags for the rest of the related data. In the example above, the first cell in the telephone number column is the header and is called Telephone; the rest of the data in that column is tagged as a telephone number. This structure will be detected by the assistive technology that will inform the user as to the type of data it contains when reading a cell. As in the case of lists, tables can also indicate the row and column being read and even the total number of rows and columns.

Making a table in Word is simple: Use the “Insert” menu to insert a table with the desired number of rows and columns. When adding data in the table, enter the label for the data in that column in the first cell of each column.

The only step that remains is to indicate that the table has headers. This is also very simple, but the commands are more difficult to find. To do this, go to the “Table Design” menu and check the “Header Row” check box. The latter is what converts the cells in the first row into header cells:



process.

Tables

If a table is going to be used in a document, designers prefer to use the table option, as it makes table creation much easier. With respect to accessibility, there are two types of tables in terms of function: those used for document layout and those used to display data.

Name	Telephone	Email
Aitor Mendez	666 123 321	aitormendez@yourdomain.com
Dolores Smith	123 456 789	dsmith@otherdomain.gov
John Watanabe	999 111 222	jwatanabe@hisdomain.jp

Illustration 51. Example of headers in a table.

One drawback is that it is usually only possible to insert column headers and not row headers. In order to do this, the table needs to be directly modified in the final PDF and this requires knowledge of Adobe's PDF tagging standard.

In Table properties (accessed via context menu -> right mouse button) you can also include something very interesting in term of accessibility, such as alternative text and a description.

To summarize the following points should be borne in mind when creating tables:

- They must have headers (if they contain data).
- Header data cells must be differentiated.
- Each row has to have the same number of columns.
- Each column has to have the same number of rows.
- A summary and a text alternative can also be added (optional).

3.3. Text alternatives

As in websites, any images added to a document must also have a text alternative. Unlike websites, when an image is used as decoration in a document, this must be stated explicitly.

Text editors offer the option to add this alternative text. In Word, this is done with the mouse by simply right clicking on the image. The context menu will open, and among the options you will find "Edit Alt Text". In the same box, if the image is decorative, it is possible to indicate this:

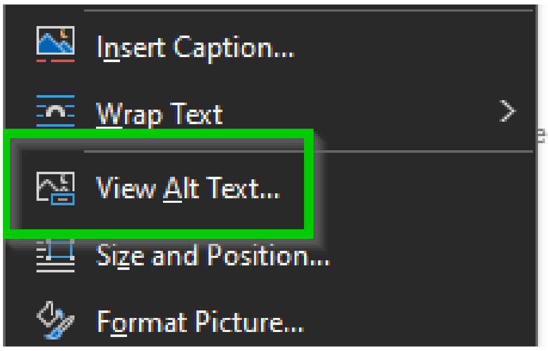
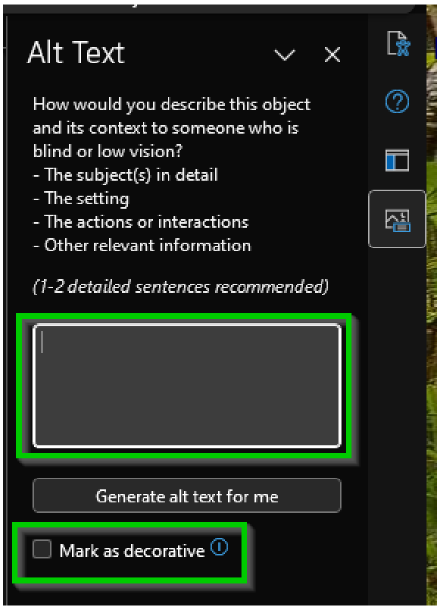


Illustration 52. Step 1: Inserting alternative text in a Word document.



How would you describe this object and its context to someone who is blind or low vision?
- The subject(s) in detail
- The setting
- The actions or interactions
- Other relevant information

(1-2 detailed sentences recommended)

Illustration 53. Step 2: Inserting alternative text in a Word document.

3.4. Contrast and use of colour

How we use colour and contrast and its impact on accessibility is universal, that is, it does not depend on the tool, the editor, or how it is used. The same aspects taken into consideration in the selection of colours for a web page should also be observed for documents. If our point of departure is a colour scheme that is known to be compatible with accessibility, it can also be used in our document; and the same goes for the use of colour to convey information: the same tools indicated in the section on web content can also be used in this case.

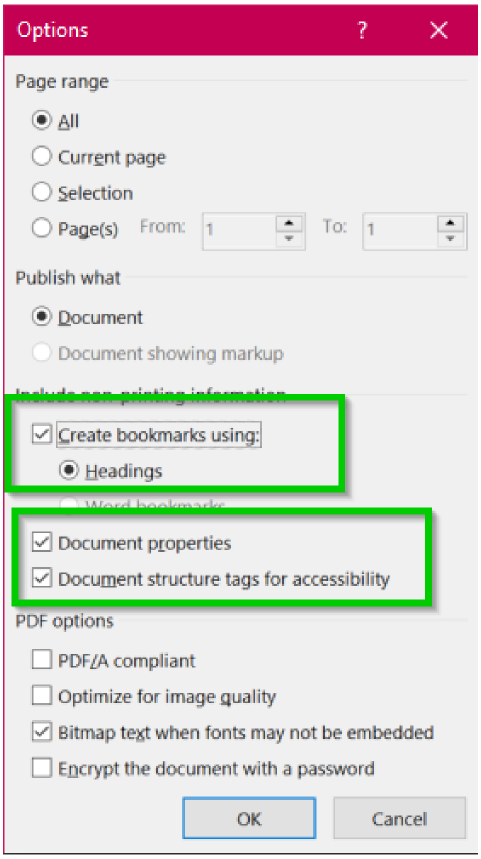
Here, Word is even more useful, as the most recent versions feature a Check accessibility option located under the “Review” tab, and it is quite effective. Among other things, it will indicate if a given text has enough contrast or if all the images have a text alternative. If you activate the accessibility, check and it indicates that everything is correct, there is a good chance that the document is accessible.

3.5. Converting to PDF

Once the entire document has been correctly formatted and tagged, the last step is to convert it into a PDF document. At this point people usually choose to print the document on a virtual PDF printer. This option is available, but it is not recommended to ensure an accessible PDF document.

Printing a document as a PDF result in a document that loses almost all tagging information and consequently, its accessibility features, which is why this option should never be used.

A better option is to export the document to PDF format. In Word this is done by simply selecting the “Save As” command and choosing PDF as the save format. Observe that when this is done, in addition to the file name and the location where the file is to be saved, an “Options” button also appears. Click on this button and make sure the relevant options are selected:



Page range

● All

○ Current page

○ Selection

○ Page(s) From: 1 To: 1

Publish what

● Document

○ Document showing markup

Include non-printing information

☑ Create bookmarks using:

● Headings

○ Word bookmarks

☑ Document properties

☑ Document structure tags for accessibility

PDF options

□ PDF/A compliant

□ Optimize for image quality

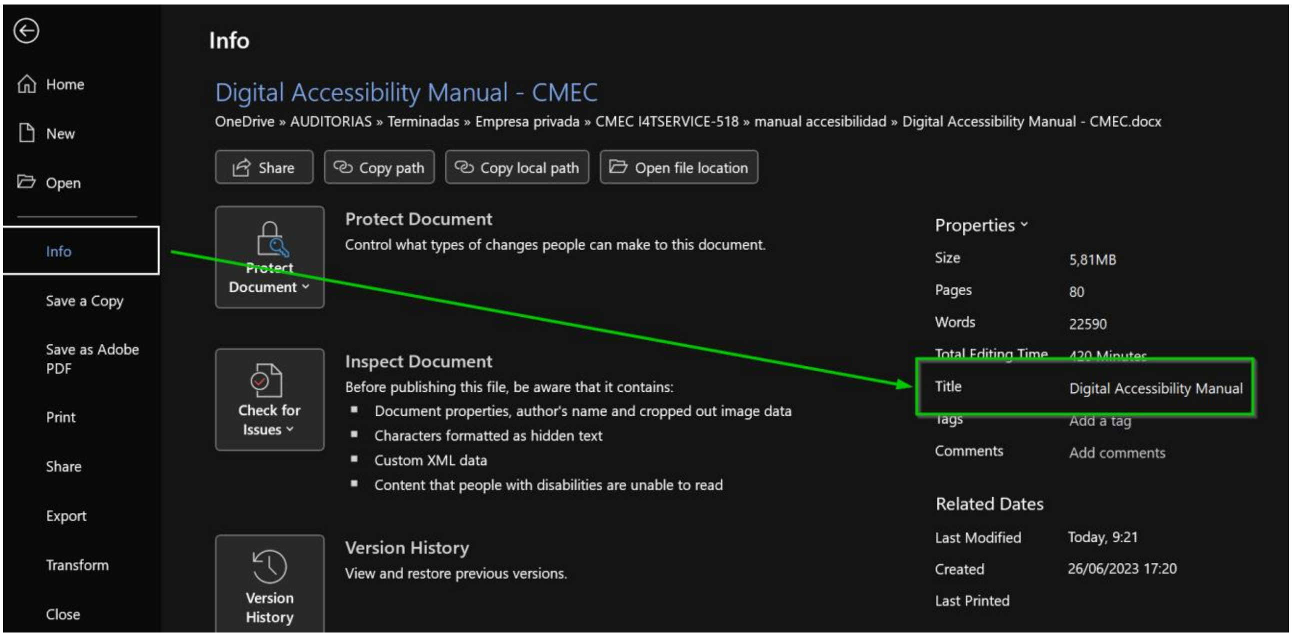
☑ Bitmap text when fonts may not be embedded

□ Encrypt the document with a password

Illustration 54. Saving a document as a PDF.

As shown in the above image, select “All” and “Create bookmarks using” so that the headings are transferred as bookmarks in the PDF. Check “Document properties” and “Document structure tags for accessibility”.

Among the criteria used to assess document accessibility, two of which are having a correct title and a defined main language. Both are document properties that require the “Document properties” box to be checked. Now, how do we indicate these properties in Word? Setting the language is easy, it is the one selected when the document is being written; if nothing is changed than the language is the one that was defined during installation. The title can be indicated by selecting File -> Info, where you will see something like the following:



Info

Digital Accessibility Manual - CMEC

OneDrive » AUDITORIAS » Terminadas » Empresa privada » CMEC I4TSERVICE-518 » manual accesibilidad » Digital Accessibility Manual - CMEC.docx

Protect Document

Control what types of changes people can make to this document.

Inspect Document

Before publishing this file, be aware that it contains:

- Document properties, author's name and cropped out image data
- Characters formatted as hidden text
- Custom XML data
- Content that people with disabilities are unable to read

Version History

View and restore previous versions.

Properties

Size	5,81MB
Pages	80
Words	22590
Total Editing Time	420 Minutes
Title	Digital Accessibility Manual
Tags	Add a tag
Comments	Add comments

Related Dates

Last Modified	Today, 9:21
Created	26/06/2023 17:20
Last Printed	

Illustration 55. Adding a title.

In the document properties column on the right, there is a field indicating the title of the document. As it is an editable field, a title can be entered here.

If you use Acrobat Pro, there is another way to convert documents into PDF files, using Acrobat's own PDF export extension. Although the result looks the same, it actually is not: the Acrobat extension is more precise, as we will see in subsequent examples.

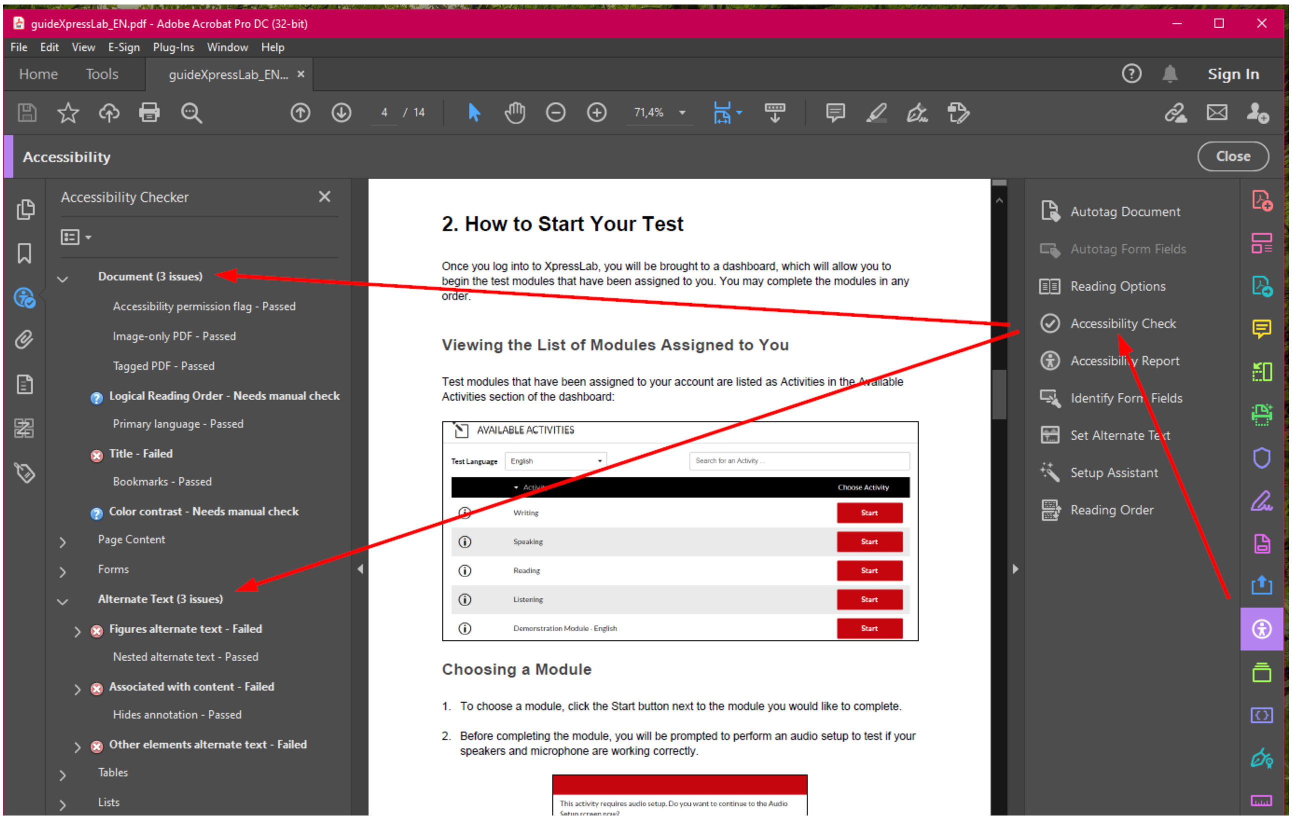
3.6. Final revision

If you have followed all of the above steps, your text will be fairly accessible, but a few details still require our attention.

As mentioned above, using the "Save as" option in Word is not the same as using the export extension provided by Acrobat, the latter offering certain functions not available with the "Save As" command. For example, in Word, decorative images are marked as decorative, but in PDF they are given the name of Artifact. If the "Save As" option is used, the image is not marked as Artifact, while in the Acrobat extension it is.

Another example is the use of tables for layout, where it isn't necessary to indicate that they have headers. That is why Acrobat gives an accessibility error if the PDF document is created using the "Save As"; however, with the Acrobat extension this does not happen.

What this means that if the document needs to be 100% accessible, it is advisable to purchase an Acrobat license, as this software suite offers full accessibility analysis to help correct most of the accessibility issues it detects, as part of its document processing functions:



Document (3 issues)

Accessibility permission flag - Passed

Image-only PDF - Passed

Tagged PDF - Passed

Logical Reading Order - Needs manual check

Primary language - Passed

Title - Failed

Bookmarks - Passed

Color contrast - Needs manual check

Page Content

Forms

Alternate Text (3 issues)

Figures alternate text - Failed

Nested alternate text - Passed

Associated with content - Failed

Hides annotation - Passed

Other elements alternate text - Failed

Tables

Lists

2. How to Start Your Test

Once you log into to XpressLab, you will be brought to a dashboard, which will allow you to begin the test modules that have been assigned to you. You may complete the modules in any order.

Viewing the List of Modules Assigned to You

Test modules that have been assigned to your account are listed as Activities in the Available Activities section of the dashboard:

AVAILABLE ACTIVITIES

Test Language: English

Search for an Activity ...

Activity	Choose Activity
Writing	Start
Speaking	Start
Reading	Start
Listening	Start
Demonstration Module - English	Start

Choosing a Module

1. To choose a module, click the Start button next to the module you would like to complete.
2. Before completing the module, you will be prompted to perform an audio setup to test if your speakers and microphone are working correctly.

This activity requires audio setup. Do you want to continue to the Audio Setup screen now?

Illustration 56. Adobe Acrobat Pro Screenshot.

4. Content Management Systems (CMS)

A CMS is software that allows you to create a work environment that helps users enter and administer website content. Broadly speaking, this is made possible via website templates which already contain pre-designed components for websites. All you need to do is adapt them to your needs and tastes and insert your content. While Wordpress, Joomla, and Drupal are perhaps the most popular, there are many others to choose from.

In theory, both Joomla and Wordpress contain accessibility clauses and statements stating that their content meets WCAG 2.1 Level AA. This means, that both CMSs should, a priori, be reliable with respect to the code they use.

This is not the case for Drupal, which does not meet WCAG 2.1 Level AA, conforming only to the previous version of the WCAG. This makes Drupal an outdated CMS in terms of accessibility.

However, because CMS are designed to be customized by users starting from a basic template, they offer the opportunity to apply themes and plugins. The various CMS only guarantee the accessibility of their tools before modification and are not responsible for any changes to accessibility that result from customization and even less for any content that is added by users.

One of the issues that arise in connection with a CMS is the great number of containers it uses, which makes accessibility quite difficult.

If users opt to use a CMS, then careful consideration should be given to which Wordpress theme to use, and how to use it. It is important to determine whether it offers accessibility at the desired level. This also applies to plugins.

This is not an easy task, so it may be necessary to have an accessibility expert review the website, as they would be able to verify if all the WCAG2.1 criteria are met.

A common issue with Wordpress web pages is that, natively, this CMS is not multilingual, and therefore, if you want your website to be in several languages, a plugin providing this support will be necessary. If we are unaware of this drawback, then the website will fail to meet Criterion 3.1.1 (Language of Page) and possibly 3.1.2 (Language of Parts).

5. Social Media

In the digital world, more than browsing websites, people communicate via social media. Any business or organization interested in reaching all its users will need to pay special attention to these channels.

Website owners usually include a frame containing their social media activity, with a Twitter feed being the most common. It is important that this channel is also accessible to users. The accessibility guidelines described throughout this manual are the same ones that apply to social media. In this case we need to ensure text alternatives for images and media, appropriate colour and contrast, clean design, and structure, if this is permitted by the platform. Also remember that frames containing newer Tweets are often too heavy to permit navigation using the keyboard: it is better to give users the option of skipping them.

The issue with accessibility and social media is that we have no choice but to play by the rules of the social media platform. Facebook, Twitter, Instagram, etc. may follow accessibility guidelines, only in part, or not at all; there is nothing we can do to change them. Fortunately, there is growing awareness on this issue and many platforms include assistance and mechanisms to make them more accessible. Let’s take a look at the most important ones.

When you link content from social media platforms, use a URL shortener.

Create hashtags by capitalizing the initial letter of each word, so that screen readers can identify and read each word correctly. For instance: #DigitalAccessibilityManual.

When writing out quantities, it is advisable to write out the number in numerals and in word form alongside. For instance: 20 (twenty).

5.1. Twitter

According to Twitter, Tweets with images get 150% more retweets than ones without. Twitter’s Help Center provides instructions on how to add alt text to images. Simply search for “How to post accessible images" in the Help Center search field and you will be given instructions for web, iOS, and Android.

You can learn more about Twitter accessibility at https://help.twitter.com/es/resources/accessibility

5.2. Facebook

Facebook is the most popular social network in the world and it they know that many of its users require a more accessible platform. This is why its Help Center offers instructions for using screen readers and keyboard shortcuts. It also allows users to add alternative text to images and subtitles to videos by way of a SubRip (.srt) file that can be added during the posting process. All of these functions are described in the Help Center.

Facebook’s Help Center has more information about accessibility options they offer: https://es-es.facebook.com/help/273947702950567

5.3. Instagram

Instagram also ensures good user experience with respect to accessibility. For images, not only does it support alternative text, it also allows you to add a long description. Alternative text, long description, and automatic captioning are also available for videos, and although the latter isn’t the best option, it may still be helpful to many users.

5.4. LinkedIn

LinkedIn allows users to tag and add alternative text to the photos they upload and also allows video captions by way of SubRip (srt) files. Furthermore, the LinkedIn website and the editor it uses for posting articles are also accessible, making user experience of this platform very positive in this regard.

5.5. YouTube

This platform has already been discussed in the multimedia section. First, the YouTube widget offers sufficient accessibility features – all its controls are keyboard accessible and have sufficient contrast. However, audio descriptions and alternative media pose problems as they are very difficult to implement. Second, YouTube’s automatic captioning is a risky option and should only be used if its accuracy can be checked first. If it’s not accurate, then captions can be added in several languages using YouTube Studio.

6. Apps

Mobile applications must be also accesible; this means that we must ensure accessibility for these tools just as we do for websites.

However, the trickiest part of accessibility lies with software developers. As content editors, the only aspects we can control are the same as those associated with website development. That is, we need to pay special attention to content, making sure it has accessible features like those described earlier in the section on web accessibility:

- Images and non-text content with their corresponding text alternatives.
- Multimedia with captions, alternative media, and their corresponding audio descriptions.
- A suitable colour theme that is compatible with accessibility, headers, banners, texts, and a link style not solely based on a different colour.
- Remember that documents must also be accessible.

For features exclusive to mobile apps, take the following considerations into account:

- They must work in two layout directions (landscape and portrait).
- Applications must be properly tagged and follow accessibility guidelines provided by the corresponding platform. All applications offer information to developers about resources and how they can be used to make the application accessible.
- If audio or video files are to be played, accessibility must be guaranteed for both the media works (captions, alternative media, and audio description), and the controls enabling content playback.
- They should not modify the accessibility features already offered by the platform.

7. Accessible technologies

Remember that your mobile apps and websites must conform with Level AA accessibility criteria. The first step is to determine the current compliance status of your mobile apps and websites by performing an in-depth manual audit in order to establish an accessibility plan that includes a schedule for correcting any errors. It is also important at this point to provide adequate training for personnel involved in content and web development.

Here is a list of solutions and technologies that have been developed to assist with these accessibility tasks.

Name	Description	Link
inSuit	This platform automatically improves web accessibility requirements and provides assistive technologies for people with disabilities.	https://www.insuit.net/es/
Sayobo	Chatbot specialized in Accessible Tourism for everyone.	https://www.sayobo.io/
Siteimprove	Platform that crawls websites and detects errors in accessibility and quality assurance issues with respect to web content, proposing solutions that can be implemented on your website.	https://siteimprove.com/es-es/accessibility/
Etiqmedia	This platform provides live and deferred automatic captioning/transcription of audio-visual content.	http://etiqmedia.com/

8. Appendix: Accessible Descriptions of Images and Infographics

This section contains accessible versions of infographics and complex images that appear in this manual. A link to each is provided in the image title.

8.1. Diagram: Decorative images

To determine whether an image is decorative follow these steps:

1. Remove the image from the page.
    - If the page retains all information, the image is decorative and its text alternative should be empty.
2. If the page no longer conveys the same information, find an appropriate alt text for the image.
    - If the page information is complete, this text is the text alternative.
3. If the page information doesn’t seem complete, repeat step 2 using another text.

8.2. Diagram: Captions

- Video only: not required.
- Audio only:
    - If live: Captions are required for Level AAA.
    - If it is not live, captions are not required.
- Video with audio:
    - The audio does NOT convey information (background music, noise): Does not require captions.
    - The audio DOES convey information:
        - If it is live, captions are required for Level AA.
        - If it is recorded, captions are required for Level A.

8.3. Diagram: transcription

- Video only:
    - If live, no transcription is required.
    - Recorded: Transcription required for Level A.
- Audio only:
    - Live: yes for Level AAA.
    - Recorded: yes for Level A.
- Video with audio:
    - If live: Transcription not required.
    - Recorded: yes for Level AAA.

8.4. Diagram: Audio description

- Audio only: not required.
- Video with audio.
    - Audio does not convey information (music, noise): Not required.
    - Audio conveys information:
        - If live, description not required.
        - If recorded, description is required for Level AA.

8.5. Diagram: principle of robustness

- Principle of robustness
    - Processing:
        - Correct tagging.
        - Correct nesting and structure.
        - Unique identifiers.
    - Name, role, value:
        - The name, role, and value of components are known to assistive technologies.
    - Status messages:
        - The status and results of processes are visible to accessibility.