

**DIA Support Project**



# **Brief Digital Accessibility Manual**



This manual is prepared on the basis of Web Accessibility educational series available on Dlia. Education web portal. The series was created within the framework of Digital, Inclusive, Accessible: Support for Digitalization of Public Services in Ukraine project implemented by the United Nations Development Programme (UNDP) in Ukraine in partnership with the Ministry of Digital Transformation of Ukraine with the support of Sweden.

The opinions, observations, conclusions or recommendations expressed in this document are those of the author and do not necessarily reflect the views of the UNDP, the Government of Sweden and/or the Ministry of Digital Transformation of Ukraine.

Author – **Dmytro Popov**, UNDP national consultant on web accessibility requirements, specialist in digital accessibility at Resource Center “Barrier-free Ukraine” NGO, director of Digital Accessibility Laboratory LLC, web accessibility specialist certified by the International Association of Accessibility Professionals (IAAP WAS).

# Contents

<b>List of terms and abbreviations.....</b>	<b>6</b>
<b>Introduction.....</b>	<b>7</b>
<b>Chapter 1. Digital accessibility: what and for whom is it?.....</b>	<b>8</b>
Computer or smartphone for mobility impaired users .....	9
Screen readers for computers .....	10
Screen magnifier as another assistive technology .....	10
Accessibility features for managing gadgets .....	11
How to make inaccessible content accessible .....	12
What is DSTU and how to apply it .....	13
Supplementary materials to Chapter 1.....	14
<b>Chapter 2. Alternative versions of non-textual content and multimedia .....</b>	<b>15</b>
Standard requirements for alternative text.....	16
Preparation of alternative text.....	17
Adding alternative text to images in text editors.....	18
Availability of images on Facebook and Twitter .....	18
Adding alternative text to images on the website.....	19
Subtitles and alternative versions for multimedia .....	21
Non-textual description .....	22
Transcript as an additional description .....	23
Supplementary materials to Chapter 2 .....	24
<b>Chapter 3. Colour, contrast and adaptability .....</b>	<b>24</b>
Colour that conveys meaning .....	25
Difference between hyperlink and text.....	26
Contrast of the text in the center of attention .....	27
Contrast for non-textual elements .....	28
Contrast detection tools.....	29
Adaptability on different screens .....	30
Supplementary materials to Chapter 3 .....	31
<b>Chapter 4. Ways to control the interface.....</b>	<b>32</b>
Using the mouse .....	32
Keyboard instead of mouse.....	33
What's the focus? .....	34
Full-scale keyboard operation .....	35
Keyboard and mobile applications.....	36
Supplementary materials to Chapter 4 .....	37

<b>Chapter 5. Structure and semantics.....</b>	<b>38</b>
It all starts with the name .....	38
Definition of language .....	39
Semantics and headers .....	40
Formulation of hyperlinks .....	41
Creation of tables .....	42
Supplementary materials to Chapter 5 .....	43
<b>Chapter 6. Fill out forms .....</b>	<b>44</b>
Visual accessibility .....	44
Work with forms.....	45
Label – a component of an available form.....	46
Instructions facilitate filling out .....	47
Checking the completed form .....	48
Supplementary materials to Chapter 6 .....	49
<b>Chapter 7. Comprehensibility of the text.....</b>	<b>50</b>
Simplifying reading .....	50
Facilitating the perception of the text .....	51
Recommendations from the International Plain Language Federation .....	52
Emotions in the text .....	54
Supplementary materials to Chapter 7 .....	55
<b>Chapter 8. Harmful and complex design elements .....</b>	<b>56</b>
Flashes and their consequences .....	56
Making flashes safe .....	57
Parallax effect .....	58
Background video, animation and sound .....	59
Complex interface components .....	60
Supplementary materials to Chapter 8 .....	61
<b>Chapter 9. Web accessibility testing .....</b>	<b>62</b>
Accessibility assessment methods.....	62
Automated vs manual testing .....	63
Availability of complex components .....	64
Assessment of accessibility of government websites .....	65
Components for manual evaluation.....	67
Supplementary materials to Chapter 9 .....	68
<b>Concluding remarks .....</b>	<b>69</b>

## List of terms and abbreviations

**Application, app** is a user program that allows solving specific application tasks of the user. A mobile application is a program that is designed to run on a mobile device such as a phone, tablet or watch.

**Assistive technologies** are special software used by people with visual, hearing, speech, cognitive, and motor impairments to work with digital technologies. One of the examples of such software is a screen reader.

**Carousel (in web design)** is an interface element that is a series of banners or announcements that are displayed in series and change at a certain frequency or at the request of the user. Unlike a slider, a carousel displays a series of images at once, rather than a single image.

**CMS (Content Management System)** — is a content management system. Software for organizing websites or other information resources on the Internet or individual computer networks.

**Digital accessibility** is the ability to use digital services for as many people as possible, including people with disabilities. Usually includes the availability of websites, mobile applications and other information and communication technologies.

**DSTU** is the state standard of Ukraine.

**DSTU EN 301549:2022** — the state standard of Ukraine “Information technologies. Requirements for the availability of ICT products and services”. The standard is based on the European standard EN 301 549, which defines the requirements for information and communication technologies that must be accessible to people with disabilities.

**ICT** — information and communication technologies.

**Mobility impaired users** are users of websites, mobile applications and other digital technologies who fall under the category of mobility impaired groups of the population. According to the official definition contained in the State Building Regulations of Ukraine “Buildings and Structures: Accessibility of Buildings and Structures for Mobility Impaired Groups” (2006), “mobility impaired groups of the population are people who experience difficulties in independent movement, receiving services, necessary information or when orienting in space.”

**NVDA** is a screen reader for the Windows operating system.

**Screen reader** is a program for reading from the screen. The software used by blind people to read text information using synthesized speech or braille and to interact with operating system interfaces and applications.

**Slider (in web design)** is an interface element that is a series of banners or announcements that are displayed one after the other and change at a certain frequency or at the request of the user. Unlike a carousel, a slider displays one banner or announcement at a time, rather than a series.

**Success criteria** are statements in WCAG that can be easily verified and are not related to a specific technology.

**TalkBack** is a screen reader that is part of the functionality of the Android operating system.

**VoiceOver** is a screen reader that is part of the functionality of Apple operating systems.

**WCAG (Web Content Accessibility Guidelines)** — Guidelines for the accessibility of web content developed by the World Wide Web Consortium (W3C).

**Web accessibility** — designing and developing websites, web tools, and technologies so that people with disabilities can use them. Web accessibility is a special case of the broader concept of “digital accessibility”, but sometimes the two terms are used interchangeably.

# Introduction

Recently, many government services have been digitized in Ukraine. Over the years, we have become accustomed to the fact that any information can be easily found in electronic form, and to order many services, only a smartphone is enough. On web portals or mobile applications, we register a child, pay taxes, participate in surveys, receive financial assistance, and register a business.

If earlier you had to stand in line to get a certificate or submit a declaration, now you can do it without leaving your home. In addition, some digital services can only be obtained in digital form.

Thus, digital services greatly simplify our lives – this is an undeniable fact. But the question arises: can all citizens use digital services? For example, blind or partially sighted people, hearing impaired or elderly people – can they even use mobile applications or websites on an equal basis with other users?

Usually yes, they can. But under a certain condition: if the application or website has accessible content (or they say barrier-free content). Therefore, if the owners of websites and mobile applications take care of their accessibility, people with disabilities and other users with impaired mobility will be able to use them.

In this manual, we will explain what digital accessibility is and how to make content accessible. After all, according to Ukrainian legislation, all websites of state authorities must ensure the accessibility of their websites and mobile applications for persons with disabilities.

Thanks to this guide, you will learn how people with disabilities use computers and smartphones and what assistive technologies are; how to make images and videos accessible to blind people and audio content to people with hearing impairment; why not everyone can use a mouse and how a mouse can be replaced by a keyboard; what is the difference between filling in forms for sighted and blind people; which design elements are harmful and which other accessibility problems most often occur when visiting Ukrainian websites.

Using the suggested recommendations, you will learn how to correctly select colours when creating a design, write clear text and properly structure it, create subtitles and alternative text for images.

This manual will be useful for designers, software developers, content managers, SMM specialists, videographers and anyone who creates websites, mobile applications or their content.

# Chapter 1

## Digital accessibility: what and for whom is it?



Accessibility is often said to be needed by people with disabilities. This is true, but not only by this category. Other groups of mobility impaired users may also benefit from accessibility.

---

Mobility impaired users are people with visual, hearing, motor and cognitive impairments, as well as older people and people with injuries or temporary health impairments. Those who are in adverse or extreme conditions, for example, reading a text under bright lighting or using a touch screen in the rain are also may be considered mobility impaired users. All these people need digital accessibility. However, the most vulnerable group is, of course, people with disabilities. They are the ones who face the most problems on the Internet.

## Computer or smartphone for mobility impaired users

Different groups of mobility impaired users can perceive content very differently. Someone definitely needs assistive technology for this. For example, blind users can use their gadgets almost equally with sighted users (with some exceptions, of course).

To access the content of the website, blind people use special software – screen reader programs, or more simply – screen readers.

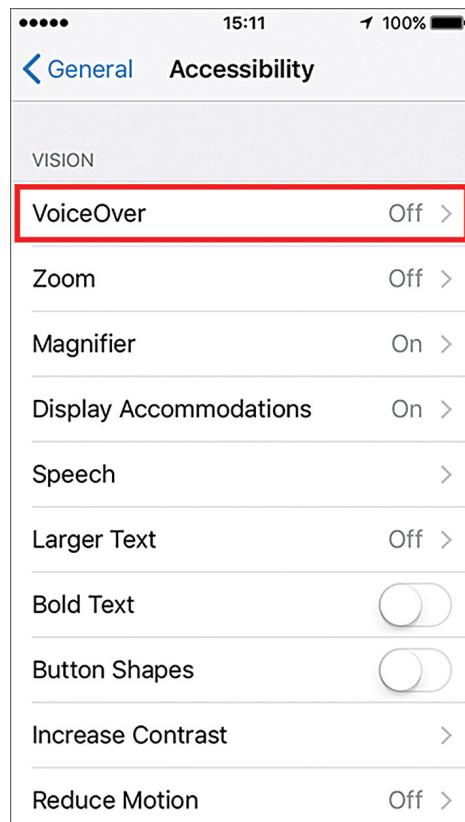
Let's see how it works. Let's imagine Mr. Petro, who is blind. He has an iPhone, and to use the screen reader, Mr. Petro needs to go to "Settings" and then open "Accessibility – **VoiceOver**". After that, he will hear a synthesized voice. Blind people use a quick command to launch **VoiceOver** – triple-pressing the Home button or the power button. But this requires that such an option be set in the settings. In recent iOS versions, it almost always works by default. If not, the blind person should have someone install the quick launch command.

It should be noted that when **VoiceOver** is launched, the control gestures change. So, if you turn on **VoiceOver** on your smartphone, you won't be able to turn it off with the usual touch gesture. If sighted people activate items with a single tap, when using **VoiceOver**, with the first tap, you hear the item's name, and the item needs to be tapped twice to be activated.

Then Mr. Petro returns to the main screen. In order to hear **VoiceOver** reading different icons, he needs to touch them. This method of navigation is called "screen exploration". However, if there are many elements on the screen, you can miss something. To avoid this, Mr. Petro can use another method of navigation – moving from one element to another by swiping the screen to the right (to the next element) or to the left (to the previous one).

By the same principle (to quickly touch an element twice for activation), you can type text on the on-screen keyboard: touch – hear a letter – touch twice to enter it. We should add that VoiceOver is not only available on **iPhone**, but on all Apple devices: **iPhone**, **iPad**, **iPod**, **Apple Watch**, **Apple TV**, as well as on **Mac** computers.

The **Android** platform also has its own screen reader – **TalkBack**. It works on a similar principle, only some gestures differ.



## Screen readers for computers

However, although mobile screen readers are now very actively developing, **Windows** remains the main platform for the blind. Screen readers for **Windows** exist since the 1990s. The standard Screen Narrator, paid **Jaws** and free **NVDA** area among the main screen readers. The latter is probably the most popular in Ukraine and in the world, as it is free. At the same time, it is not inferior to paid counterparts in terms of quality.

The working principle of **NVDA**, like other screen readers, is as follows: the user moves from element to element using the arrow keys, Tab and some combinations, and the screen reader announces the name of the element in focus.

To activate an element, for example, to open a program or folder, mobility impaired users use the **Enter** key. In general, blind people use the usual combinations to work in **Windows**: **Alt+F4** – close a window, **Alt+Tab** – switch between open windows, etc.

By the way, all blind people usually work only with a keyboard and do not use a mouse. This is due to the fact that in order to move the mouse cursor, you need to see the actual cursor.

The screen reader allows working not only with folders or documents, but also with web pages on the Internet. To do this, the user opens a regular browser, enters the address, opens the website, and the screen reader reads all these actions.

It should be noted that the screen reader voices not only the names of the elements, but also their semantic roles, state and other properties. The screen reader informs that the menu is rolled down, that certain text is the title, and that there are a certain number of items in the list. However, it is worth noting that sometimes the screen reader does not read the content completely or does not read it at all. The following Chapters of the brochure explain how to avoid this.

## Screen magnifier as another assistive technology

Screen reader is the most requested assistive technology for mobility impaired people, but there are others. This is, for example, a screen magnifier.

Let's imagine a mobility impaired person who has weak vision. Such individuals work with a smartphone and a computer in an almost normal way. Residual vision allows them to see app icons, photos, and more. At the same time, it is sometimes difficult to read the text, especially if it is small and has insufficient contrast.



Photo: support.microsoft.com

All popular platforms have special software – a magnifier. How does it work on [Android](#)? To turn on the magnifier, you need to open:

**Settings** → **Accessibility** → **Magnification**

Next, we will configure the keys for quick launch, for example “Hold the volume keys”.

To enlarge a part of the screen:

- press the volume up and down buttons to activate the magnification function;
- touch the screen anywhere except the keyboard and navigation bar;
- move two fingers to move around the screen;
- pinch two fingers to change the scaling level.

It works roughly the same way on [Apple](#) devices.

[Windows](#) also has a magnifier. However, it is usually not used often, as it is enough to enlarge the text in the browser.

After loading the page in the browser, you should press the **CTRL** and **plus (+)** keys, increasing the text to the desired size. Accordingly, press **CTRL** and **minus (-)** to decrease.

The main thing is that when magnifying, horizontal scrolling should not appear and part of the text should not disappear.

## Accessibility features for managing gadgets

There are many other ways to work with a smartphone and a computer: voice control, one-key (switch) control, etc. All of them are presented in the corresponding “Accessibility Features” Chapter on each modern platform.

Not all mobility impaired users use assistive technologies. However, regardless of how users access content, it is important for them that the content is presented in a form that is convenient for them, namely:

- videos with subtitles for the deaf or with comments for the blind;
- clear text without complex structures with good contrast and structure, which is easy to read simply from the screen or with the help of a screen reader;
- images, graphics and diagrams that are well described in the text, etc. Having a disability is not a prerequisite to having difficulty accessing content. **All people in certain situations can be limited in mobility and face barriers.**

For example, remember how it is difficult to read the text on the smartphone screen in the bright sun or to order a taxi with one hand when the suitcase is in the other. Have you ever had to re-read the notification several times to understand exactly what the application is asking you: should you agree or cancel the request?

Such problems often occur for the elderly. They believe (although sometimes it only seems to them) that they are unable to master work with a certain application. Thus, sometimes they do not understand what needs to be done to register, in particular, in which field, what and how to enter. Even if instructions are available, questions arise because of a significant number of obscure words and abbreviations that the elderly do not understand.

## How to make inaccessible content accessible

Digital content can be accessible to everyone. It is not at all difficult to do, it is only necessary to develop it considering the rules of digital accessibility.

Digital accessibility is an approach to making digital content accessible to all people regardless of their ability, age, and whether they have a disability or not.

---

**WCAG (Web Content Accessibility Guidelines) is the main document that defines accessibility principles, provides recommendations and contains accessibility criteria.**

The first version of this document was developed by the World Wide Web Consortium (W3C) back in 1999. Today, WCAG 2.1 is the latest version.

The Guidelines are based on four principles – perception, management, understanding and reliability, for the implementation of each of which recommendations are proposed.

Each recommendation contains success criteria against which compliance with the standard is checked. Some of these success criteria will be discussed in later Chapters of this brochure. The success criteria are divided into three levels from the lowest (**A**) to the highest (**AAA**, triple A). **AA** level (double **A**) is optimal.

**A**  
The lowest

The simplest level of compliance

**AA**  
Middle

Optimal level with readable text, average text contrast ratio and interface support

**AAA**  
The highest

The highest level of accessibility that requires sign language for audio content and a high contrast ratio for text

To be **WCAG** compliant at level AA, for example, a website must meet the requirements for all success criteria at levels A and AA.

Most international and local standards are based on **WCAG** recommendations. For example, the European standard EN 301 549 is based on **WCAG** version 2.1. This standard was also adopted in Ukraine as DSTU EN 301 549:2022 “Information technologies. Requirements for the accessibility of ICT products and services” in June 2022.

## What is DSTU and how to apply it

If **WCAG** is primarily recommendations and accessibility criteria for websites, then EN 301 549:2022 also covers mobile applications, electronic documents and any other information and communication technologies (ICT).

The requirements for websites are almost the same as in **WCAG**: a website must meet all the success criteria of Levels A and AA, and may also (optionally) meet the criteria of Level AAA. Requirements for websites are given in Chapter 9 of DSTU EN 301 549:2022.

Almost the same criteria apply to mobile applications and electronic documents, but with some clarifications and with some exceptions. These criteria are listed in Chapters 10 and 11 of the specified standard, respectively.

The standard also contains additional general criteria that can be applied to all ICT that meet certain conditions, such as implementing a biometric function or a voice communication function. These criteria are presented in Chapters 5, 6, 7, 12 and 13.

Some requirements of DSTU EN 301 549:2022 are discussed in more detail in the following Chapters of this manual.

## Supplementary materials to Chapter 1:

1. Getting started with accessibility options on iPhone.



URL: <https://support.apple.com/uk-ua/guide/iphone/iph3e2e4367/ios>

2. Availability of Apple products.



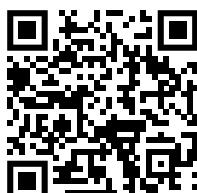
URL: <https://www.apple.com/accessibility/>

3. How to use the Android accessibility menu.



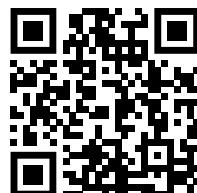
URL: <https://support.google.com/accessibility/android/answer/9078941?hl=uk>

4. Overview of Android accessibility features.



URL: <https://support.google.com/nexus/answer/6006564?hl=uk>

5. About NVDA.



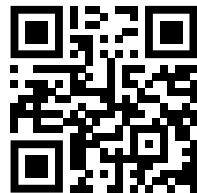
URL: <https://www.nvaccess.org/about-nvda/>

6. Web Content Accessibility Guidelines (WCAG).



URL: <https://www.w3.org/TR/WCAG21/>

7. Barrier-free handbook.



URL: <https://bf.in.ua/>

## Chapter 2

# Alternative versions of non-textual content and multimedia



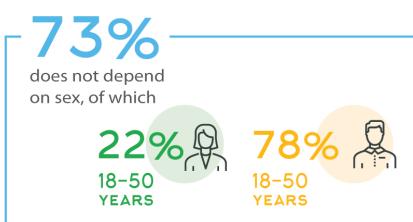
In this Chapter, we'll talk about non-textual content, multimedia, and alternative versions for these formats.

**Non-textual content** includes images, illustrations, graphics, etc. Such content generally helps to better understand the information, but can be a barrier for visually impaired users.

In Chapter 1, when we explored how screen readers work, we learned that these apps don't read all content. Namely, they can only read text and cannot reproduce image content.

This in no way means that images should be abandoned! Images complement, decorate and help to better understand the textual content. Sometimes they are the main content itself.

**WHO WILL MANAGE  
A UTILITY ENTERPRISE BETTER?**



Example of an infographic

## Standard requirements for alternative text

It is enough just to add alternative text to perceive the image, and blind users will be able to learn its content. So, when and how should you add text descriptions to images? Let's turn to the standards, namely the current **DSTU standard EN 301 549:2022**.

Subclause 9.1.1.1 covers the issue of non-textual content on websites, it states: "If the ICT is a web page, it must meet the success criterion 1.1.1 "Non-text content". The issue of non-textual content in electronic documents is covered by subclause 10.1.1.1 of the specified standard, and in mobile applications – 11.1.1.1.

In all three cases, the standard refers to success criterion 1.1.1 of **WCAG 2.1**. Let's consider this success criterion.

As mentioned in Chapter 1, **WCAG** consists of four principles – perception, management, understanding and reliability. Recommendations have been developed for each of the principles. The latter, in turn, contain success criteria – statements that can be used to check the success of compliance with the recommendations.

Success criterion number 1.1.1 means that you need to refer to the first principle – "Perception", find recommendation 1 in it, and the first (and only) success criterion in it.

Recommendation 1.1 Text Alternatives states:

*Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language.*

Success Criterion 1.1.1 Non-textual Content contains a clearer statement:

*All non-textual content presented to the user shall have an equivalent alternative text version, except for the situations listed below.*

Therefore, all informative images (that is, those that convey some information) must be provided with a textual description equivalent to that image. A blind user who reads this text description using a screen reader will receive almost as much information as a person who can see the image. In practice, this is not always possible, but we should strive for it.

## Preparation of alternative text

With well-written alternative text, blind users can understand the content of images.

Here are some tips for writing quality alternative text:

**1.** When writing alternative text, answer the following questions:

- Why should there be an image here?
- What information does it convey?

- If I couldn't see the picture, what words would I use to convey the same information or describe the function?
2. Always consider the context. For example, for the magnifying glass icon on the search button, it is better to give the description “**Search**” instead of “**Magnifying glass**”.
  3. Make sure all description text is useful. Words like “chart”, “images” or image file names are generally not very useful, so they should not be used in the text description.
  4. The length of the alternative text should not exceed 150 characters.

And do we need to add alternative text to all images? Let's return to the individual cases described in Success Criterion 1.1.1. Let's pay attention to the last case – “*Decoration, formatting, invisibility*”. If non-textual content is decorative, used only for visual formatting, or not presented to users, then it is implemented in such a way that it can be ignored by assistive technologies.

This means that if the image is not informative, but only decorative (e.g. a line to separate the text, a frame, an abstract background image, a ribbon under the title text, etc.), then such images do not need alternative text.

But it is important that assistive technology, i.e. a screen reader, can ignore such images and not try to read their title.



Examples of decorations

## Adding alternative text to images in text editors

Suppose you have written a good text. How to add it to the image? It depends on the application (network, etc.) where the image is placed.

In **Microsoft Office** documents, alternative text is called “placeholder text”.

Consider adding placeholder text in **Word** as part of the **Microsoft Office 365** package. Note that the procedure is similar in **Outlook**, **PowerPoint**, and **Excel**.

1. Right-click the object and select **Edit Placeholder Text**. Or you can select the object and open “**Format**” in its menu, then “**Text**”.
2. In the “**Placeholder Text**” area, enter a sentence or two in the text box to describe the object's content for visually impaired users.

In other versions of **MS Office**, the procedure is similar, but the location of the menu items may differ.

Then you should check the added text using a screen reader.

If the visuals are purely decorative, i.e. simply add attractiveness but are not informative, they can be marked as such to which it is not required to add alternative text. You can mark visual elements as decorative in [Excel](#), [PowerPoint](#), and [Word](#). To do this, set the “Mark as decorative” flag. After that, the text field will become inactive and the screen reader will ignore the decorative image, as required by the standards.

## Availability of images on Facebook and Twitter

The most popular means of distributing images today are social networks. Let's take a look at how to add text to an image in a [Facebook](#) post.

Facebook automatically adds alternative text to uploaded images, but these descriptions aren't always accurate. You can make your images more accessible to people with visual impairments by writing and adding accurate alternative text.

To add alternative text to an image you upload using the [Facebook](#) mobile app, open the app on your smartphone, create a new [Facebook](#) post, and then select a photo from your device to upload as usual. In the post preview, tap the three dots in the upper right corner of the image. A screen reader will read this button as “**More**”.

A menu will appear at the bottom of the screen. Here, tap on the **Edit Alt Text** option. You are now on the **Change Alt Text** screen.

Facebook will also display auto-generated text. In the text box below the image, enter text that accurately describes the image you are uploading.

After that, click **Done**. Alternative text is now added to the image or photo.

You can also add alternative text to your [Facebook](#) images from your computer. To do this, start creating a new image post as usual.

Then, in the post preview, click **Edit** in the top left corner of the image. The **Photo Details** window will appear. Here, click on **Alt Text** at the bottom of the menu on the left panel. The **Alt Text** menu will expand. Click the tooltip next to the **Custom Alt Text** option to select it, and then type the alternative text for the image in the text box. Now click the **Save** button at the bottom of the window.

Let's take a look at adding text to an image in a [Twitter](#) post.

Open the [Twitter](#) app on your [iPhone](#), [iPad](#), or [Android](#) device, create a new tweet, and add an image to it. When previewing an image, click the **Add Description to Photo** button in the lower right corner of the image. Then [Twitter](#) will ask you if you want to add a description to your image. Click **Yes** and enter the description you want to add.

Adding alternative text using the [Twitter](#) website is also easy, but the procedure is slightly different. Start tweeting as you normally would, then add an image to it. In the image preview, click the **Add Description** button below the thumbnail.

Click **Yes** to confirm that you want to add a description. In the Description field that appears, enter a description of the image and click **Save** in the upper right corner.

Note. The names of the buttons and their location may be different, as the application interface is sometimes updated.

## Adding alternative text to images on the website

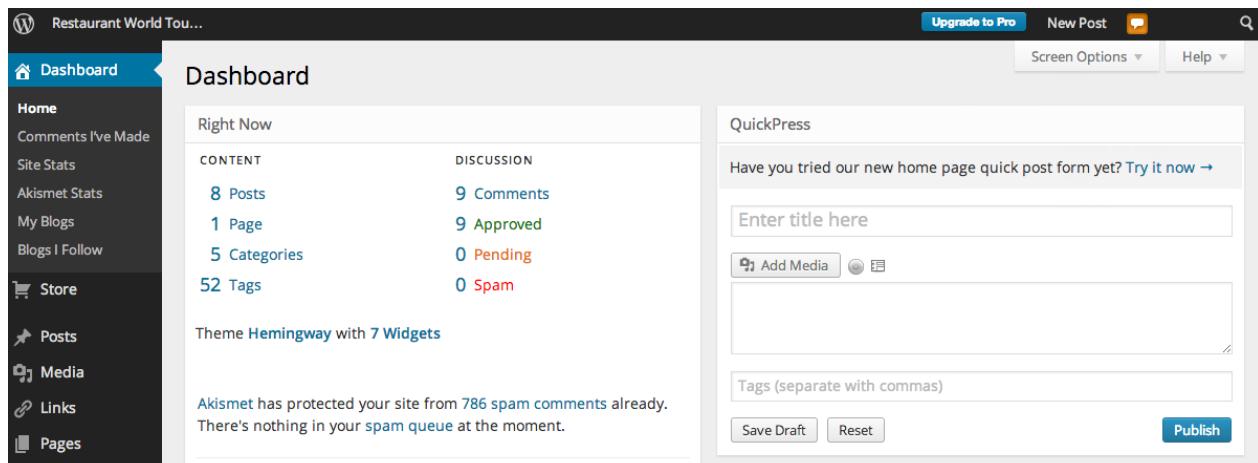
This manual does not cover issues related to website development, so we will only focus on ways to add alternative text using the admin panel.

Let's consider adding alternative text in the world's most popular content management system (CMS) – [Wordpress](#).

In [Wordpress](#), you can upload images from the media library or directly when creating/editing a post. First, consider the procedure for adding alternative text when loading an image through the media library:

- Go to **Media > Library**. Here you will see all the existing images uploaded to your website.
- Click the **Add New button** to upload a new image. After uploading, you will see the images in the list.
- Click on the image to open the editing screen. You can select any of the existing images to edit and add alternative text.
- Fill in the alternative text in the **Alt Text** field.
- Close the image and [Wordpress](#) will automatically save the alternative text for this image.

The Gutenberg editor (the default text editor in [Wordpress](#)) offers an additional way to insert alternative text. Once you've uploaded or pasted an image into your post, click on that image to see the **Block** settings on the right sidebar. Here you can add alternative text to your uploaded images in the [Settings](#) Chapter.



Screenshot of the Wordpress panel

The process of adding alternative text may vary by **CMS**. If you don't know how to add alternative text in your **CMS**, ask your website developers. So, we've looked at some ways to add alternative text to images. But not only images need an alternative version.

One common accessibility problem is missing text in **PDF** files because they are not accessible to screen reader users. This happens when we scan text and save it to **PDF**. In this way, we receive not a text file, but an image of the text, which, let's remind you, screen readers do not read.

To get an accessible **PDF** document, you need to recognize the image and then manually correct the text, since recognition programs often do not fully recognize the text. If possible, it is better not to use scanning, but to export the text from text documents such as Word documents. It is important to use the export function, not the print function.

## Subtitles and alternative versions for multimedia

Multimedia materials such as video and audio also require alternative versions, as they may not be accessible to everyone. For example, a person who is deaf is unlikely to be able to learn the video material of any educational course if the video does not have accompanying captions (also called "subtitles"). Subtitles are needed not only by the hearing impaired, but also by anyone who watches videos without sound or is in a noisy environment.

There are two types of captions you can use in your video: open and closed.

---

**Closed caption** is a time-stamped text file that is loaded into a media player for displaying in a video. The user can enable captions if he needs them. Open captions are always enabled because they are embedded in the video during creation.

The advantage of closed captions is that they can be turned on and off. But this possibility depends on the player that supports caption files. The advantage of open captions is that, regardless of the player involved, the person who needs captions will receive them. But captions can be distracting, so not being able to turn them off might annoy some.

Therefore, it is always recommended to use open captions when presenting video at a conference, so that the presenter does not need to worry about the type of player and the availability of caption file support. Closed captions are the best choice for online videos. You just need to make sure that the player on the website supports caption files.

### How to make good captions?

Here are some tips for creating captions:

- captions must be verbatim, unless simplified captions are intentionally created for an appropriate target audience, such as people with cognitive disabilities.

For any scripted material, captions must reflect the entire content of the script verbatim, including any placeholder sounds (such as “mm” and stuttering) that are intentionally included in the script. In live broadcasts or unscripted videos, stuttering may not be included if it may impair the perception of captions;

- important background sounds should be rendered in captions, preferably in [square brackets] or in (round brackets). However, it's important not to clutter the captions with too many descriptive words, as this can distract from viewing. Music must be identified by title and artist in captions, unless this would be inappropriate for the content;
- the text that is spoken behind the scene must be presented in subtitles;
- if the video does not show who is speaking, this person must be identified in the captions;
- if the speech cannot be heard or is difficult to understand clearly, this should be indicated in the captions.

A transcript, that is, a text transcription of the entire audio content of the video can be an addition to captions. Such text is usually placed under the video or on a separate page. It can be useful for people with hearing impairments, those with an imperfect knowledge of the language, and anyone else who is more comfortable reading text than watching videos. Recommendations for transcripts are almost the same as for captions.

However, the best way to present audio content to the hearing impaired is through sign language. When creating such a format, it should be taken into account that sign language differs in different countries, and there are not many sign language specialists in Ukraine. Thus, this format will require a lot of resources. However, if your content or event is specifically made for people with disabilities, be sure to use sign language.

## Non-textual description

The audio description (sometimes it is also called captioning) is an important addition to the video. This is a description of the events that occur in the video. Audio description allows people who cannot see the video to understand the content.

The film consists of many elements that are entirely visual, with no auditory component. For example, a person can make a certain facial expression, but not say a single word. The camera may show an amazing mountain landscape, but no one in the film says, “*This is a mountain landscape.*” There is a lot of visual information important to a video that can be difficult to understand for someone who can't see a screen.

**Audio description** is a solution that makes visual information in movies accessible to blind people. Audio description can be provided on the same audio track as other video audio, or as a separate track that the user turns on when needed.

Here are some recommendations for creating audio descriptions:

- descriptions should, as far as possible, coincide in time with the actions being described;
- descriptions should be written in clear, simple language. Technical terms should be avoided (except when the context requires it) and, if possible, with an explanation;
- descriptions must be formulated in the present tense;
- it is necessary to inform the name of those who appeared on the screen for the first time and will speak;
- descriptions should not give more information than a sighted viewer can get from the image;
- visual elements that are invented, remembered, or obviously part of a dream, should be distinguished from real events;
- descriptions must be neutral. Explanatory descriptions, evaluative judgments, and aesthetic judgments, for example, “*She is looking into the distance, thinking of home*”, should be avoided.

## Transcript as an additional description

In addition to the audio description, a **transcript** can be created. It is the only way with the help of which people with simultaneous vision and hearing impairment can access information. Therefore, this alternative version benefits a wider audience. It is better if you have both an audio description and a transcript.

Creating audio descriptions can take time and resources. Therefore, we recommend that when creating a video, a script should be developed so that all important visual information is also transmitted through the original audio accompaniment of the video. Then the audio description will not be needed.

Success Criterion 1.2.1 – 1.2.9 should be consulted to ensure that multimedia content conforms to **WCAG**.

## Supplementary materials to Chapter 2:

1. Alternative text for images.



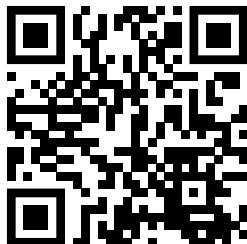
URL: <https://accessibility.psu.edu/images/alttext/>

2. WCAG: Understanding Success Criterion 1.1.1 Non-Textual Content.



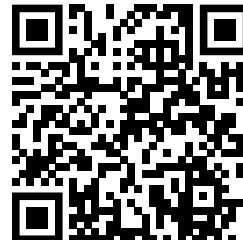
URL: <https://www.w3.org/TR/WCAG21/#non-text-content>

3. Guidelines and best practices for creating subtitles for educational video.



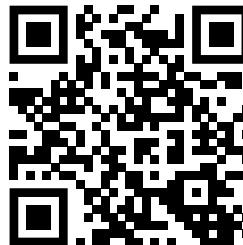
URL: <https://dcmp.org/learn/captioningkey>

4. WCAG: Understanding Success Criterion 1.2.2 Captions (Prerecorded).



URL: <https://www.w3.org/TR/WCAG21/#captions-prerecorded>

5. Course materials on audio description from ADLAB PRO.



URL: <https://www.adlabpro.eu/coursematerials/>

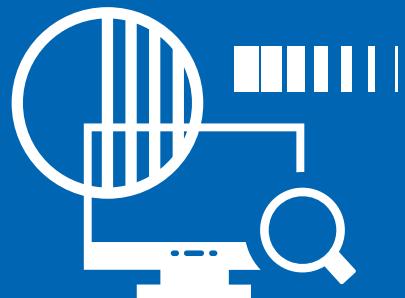
6. WCAG: Understanding Success Criterion 1.2.3 Audio Description or Media Alternative (Prerecorded).



URL: <https://www.w3.org/TR/WCAG21/#audio-description-or-media-alternative-prerecorded>

## Chapter 3

# Colour, contrast and adaptability



In the previous Chapters, the issue of content accessibility for those who cannot see it, but read it with the help of a screen reader, was considered. Now let's focus on visual accessibility for people who can see but may not perceive colours or do not have perfect visual acuity. This may also apply to people with good eyesight who read text in poor light or in bright sunlight.

There are almost three times more people with low vision than people who are completely blind. In general, about one in twelve people has some form of colour blindness, and it is much more common among men. Thus, in the USA there are approximately 8% among men and 0.4% among women.

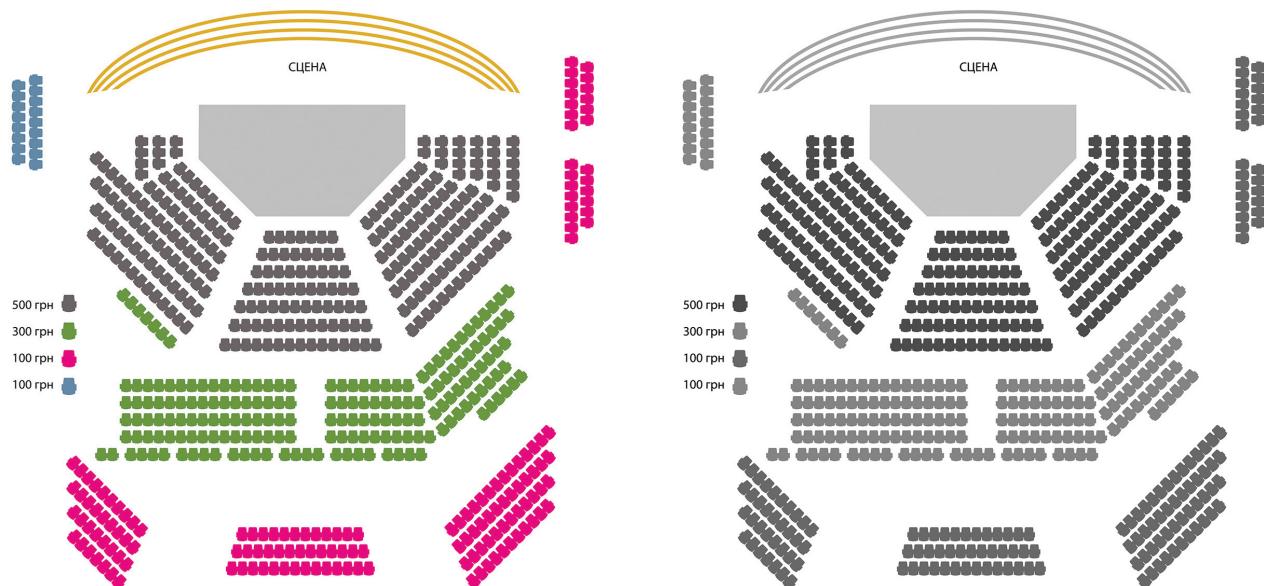
There are people who need to enlarge the text or have contrasting colours, or who suffer from colour blindness. There are also people with reading disabilities or certain types of cognitive disabilities who can have good vision. But their brains have trouble processing visual information, making it difficult to distinguish or understand text.

So, in this Chapter we will focus on the use of colours, contrast and adaptability issues.

## Colour that conveys meaning

In diagrams, tables and simply on pages, certain information is often conveyed through colour. This is very convenient, because you can quickly find the right place, for example, a place on the hall scheme: we look for green ones, that is, places of a certain value, we do not pay attention to the others.

However, if colour is used as the only means of conveying information, people with colour blindness, low vision or the blind will not be able to use it, because they will see the scheme in black and white.



Scheme of the concert hall, where tickets of different prices are marked with different colours (left) and diagram in black and white (right)

Or another example. Here is a table of training courses, colour-coded according to difficulty level, with no text markings in the table.

Accessibility courses	
Course name	Level*
<b>WAI ARIA application</b>	Red
<b>Mobile accessibility</b>	Red
<b>PDF accessibility</b>	Yellow
<b>Principles of accessibility</b>	Green

\* Red – advanced, Yellow – intermediate, Green – beginner

Example of a table without text labels

Thus, you need to add text marks. Inserting text into empty table cells makes it accessible to both colourblind and visually impaired users, as well as people who use screen readers. It also removes any possible ambiguity, which is good for all users.

Accessibility courses	
Course name	Level*
<b>WAI ARIA application</b>	<b>Advanced</b>
<b>Mobile accessibility</b>	<b>Advanced</b>
<b>PDF accessibility</b>	<b>Intermediate</b>
<b>Principles of accessibility</b>	<b>Beginner</b>

Example of a table with text labels

Any information conveyed in colour must be accompanied by alternative text version. The text should be the foundation on which the rest of the available content is built. Everything must exist in text format. However, this does not mean that colour marking should be abandoned. Simply, it should be used only as an additional tool for transmitting information.

## Difference between hyperlink and text

Links to web pages or sites are often provided along with the text. At the same time, it is worth remembering that sighted users should be able to visually distinguish links from text. Otherwise, they won't know which items to click and may miss an important link.

The best option is to use a different colour and underline (this is a typical presentation style). There is no need in changing the default style to conform to accessibility best practices.

Some designers don't like the look of underlined links, so they remove the underline from the style. At the same time, it is not recommended to do this in the main content of the document, because people with weak vision may not distinguish the colour of the text from the colour of the links before hovering a cursor over them or tabbing to them.

This recommendation applies only to links that are placed in the middle of the text, not to links in menus. Sighted users will understand that items are clickable menu items. At the same time, the menu itself should be designed so that it is obvious that it is a navigation menu.

Requirements for the use of colours to indicate elements are given in [WCAG Success Criterion 1.4.1](#).

## Contrast of the text in the centre of attention

The issues of colour and contrast are among the accessibility issues that are often overlooked. A person with low vision or colour blindness will not be able to distinguish text against a background without sufficient contrast.

The minimum contrast requirements are given in [WCAG](#) Success Criterion [1.4.3](#), and the expanded ones are in 1.4.6.

It should be noted that compliance with the advanced requirements is not mandatory, since success criterion 1.4.6 has a level AAA. According to DSTU EN 301 549:2022, the criteria of levels A and AA are mandatory. The minimum contrast ratio according to this DSTU is 4.5:1.

Providing sufficient colour contrast between text and background makes content easier to read for people with low vision. Such users may not use assistive technologies to enhance contrast if the contrast between text and background is 4.5:1. This applies to small text when a regular font size of less than 18 points or a bold font size of 14 points is used.

---

Bold text over 18 points or normal – over 14 points is called “[large text](#)”. As a rule, such text has wider strokes, which makes it easier to read at a lower contrast. Therefore, the [minimum contrast between large text and the background is 3:1](#).

Designers or content creators are advised to consider increasing the colour contrast beyond the minimum requirements – up to 7:1, as specified in the advanced requirements (Success Criterion 1.4.6). This ratio is considered the optimal contrast and is recommended, in particular, for viewing text on a mobile device. Due to the smaller viewing window and the different environments in which mobile devices are used, the improved contrast will make your content accessible to the widest possible audience.

---

Thus, the [minimum contrast ratio for small text is 4.5:1](#), for large – [3:1](#), but the optimal ratio will be 7:1 or more. The last value is recommended, not required.

## Contrast for non-textual elements

These rules also apply to people with colour vision impairments (i.e., colour blindness). Some violations of human perception of colours can manifest themselves in different degrees of their brightness for such a person. Therefore, following this rule also ensures that the colour does not affect the perception of the content on the screen by these users.

Note that the background can be not only a solid colour, but also a pattern, gradient, image, animation or video. For all these conditions, it is necessary to follow the rules regarding the minimum contrast.



Demonstration of the text on the background

Contrast rules exist not only for text, but also for non-textual content. Graphics that convey information should have a contrast ratio of at least 3:1 relative to adjacent colours.

This applies to icons that convey information, lines on line graphs, and sectors on pie charts. For example, users who cannot distinguish lines on a line graph or sectors on a pie chart will not be able to interpret the information.

Exceptions are cases when the colour change changes the essence of the information. Thus, changing the colours of flags, photos, heat maps distorts the information, so these images should not meet the contrast requirements.



Examples of images

## Contrast detection tools

Contrast ratio values can seem difficult to remember. Don't worry, though, because there's no need to memorize colour combinations or contrast ratios. There are many tools available to help you analyse the contrast ratio of your colours and determine if they meet DSTU requirements.

An example of such a tool is **Colour Contrast Analyzer** from [Deque University](#). Just enter the codes of two colours if you have them, or choose colours from the palette and get the result – whether the selected pair meets the **WCAG** contrast requirements (Success Criterion 1.4.3). Links to this and other tools are also available in the Supplementary materials for this Chapter.

If you need to determine the colour contrast in an image, document, or screenshot, but you don't know the colour codes, use the **Colour Contrast Analyzer** tool, a free program for [Windows](#) and [Mac](#). Using the eyedropper tool, select the colour of any element and the background colour and set the contrast. Also look for links to download this program in the materials for this Chapter.

To adjust the contrast of text in documents, use the tool for checking the availability of the corresponding program in the **Office package** – [Word](#), [PowerPoint](#), etc.

To receive notifications about accessibility issues in [Word](#), [Excel](#), and [PowerPoint](#), select the appropriate check box while working in these programs. Don't turn off the accessibility check until you finish the work. The **Accessibility** button will be added to the status bar and the ability to track accessibility issues in real-time. You can open accessibility checker at any time by clicking the button in the status bar.

Pay attention! Violation of contrast requirements is not a problem in [MS Office](#), so look for such violations among warnings.

In [PDF](#), this option is implemented using the **Accessibility Checker** tool in [Acrobat DC Pro](#). To do this, select the menu commands

**Tools** → **Accessibility**

This set of tools will be displayed in the secondary toolbar. Click on **Accessibility Checker**. The **Accessibility Checker Options** dialog box will be displayed. In the **Report Options** Chapter, select the desired output form. You can save the test results to an [HTML](#) file on your computer or attach them as a file in a document. Finally, you can review the report for contrast errors.

We will come back to the accessibility testing tools in the next Chapters.

Now let's move on to content adaptability and content scaling.

## Adaptability on different screens

Creating a beautiful responsive design that displays on all devices – mobile and desktop – is one of the most effective things you can do to help people with low vision. At the same time, when it comes to mobile devices, it can be said that all people have poor eyesight. Screens are so small that if you try to create a normal design for them, the text will become very difficult or almost impossible to read. If developers do not create a responsive design with enlarged text, users will have to zoom themselves. Most people think of it as a mobile design feature.

Note that responsive design works well in desktop browsers. In fact, this is where responsive design can make the most difference for users with low vision.

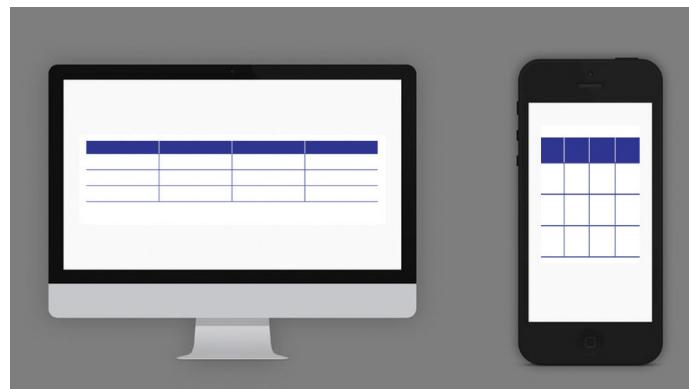
On desktop, such users can take full advantage of the large screen and zoom in as much as they need. To test whether the site responds to increased text on a desktop computer, reduce the size of the browser window or use the browser's zoom function.

In most browsers, scaling can be activated with **Ctrl + Plus (+)** (on **Windows**) and **Command + Plus (+)** (on Mac).

To reduce text, use **Ctrl + Minus (-)** (on **Windows**) and **Command + Minus (-)** (on Mac) respectively.

On a mobile device, it is enough to make sure that the site has been rearranged and the text can be enlarged by stretching with two fingers. Special attention should be paid to tables and forms. They should also rearrange properly, primarily without horizontal scrolling appearing.

According to **WCAG** requirements, a user should be able to enlarge text without assistive technology up to 200% without losing functionality. The requirements also state that the website must be rearranged without the need for horizontal scrolling. Requirements for text enlargement and rearrangement are given in Success Criteria 1.4.4 and 1.4.10.



Thus, now you can independently test the adaptability of the website of your organization or public authority. If something does not work – ask the developers to fix it during the next updating.

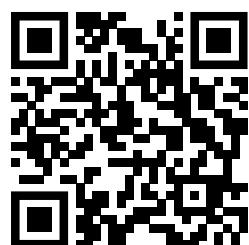
## Supplementary materials to Chapter 3:

1. Web accessibility criteria – colour contrast.



URL: <https://www.csun.edu/universal-design-center/web-accessibility-criteria-color-contrast>

2. WCAG: Understanding Success Criterion 1.4.1 Use of Colour.



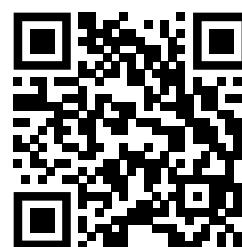
URL: <https://www.w3.org/TR/WCAG21/#use-of-color>

3. WCAG: Understanding Success Criterion 1.4.3 Contrast (Minimum).



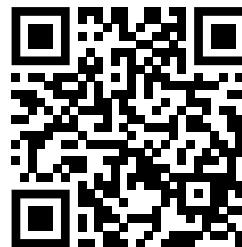
URL: <https://www.w3.org/TR/WCAG21/#contrast-minimum>

4. WCAG: Understanding Success Criterion 1.4.4 Resizing text.



URL: <https://www.w3.org/TR/WCAG21/#resize-text>

5. Deque Colour Contrast Analyzer.



URL: <https://dequeuniversity.com/color-contrast>

6. TPGi Colour Contrast Analyzer.



URL: <https://www.tpgi.com/color-contrast-checker/>

# Chapter 4

## Ways to control the interface



Let's move on to consider how to control the interface. Looking ahead, let's say that it's not just a mouse and a touchscreen.

### Using the mouse

Most developers take basic mouse accessibility into account without even thinking about it. High functionality is available for the mouse in almost all cases.

However, there are things that are often overlooked but are important to certain groups of users. Some users may use the mouse, but with less precision. For example, they cannot place the mouse pointer in the right place because of tremors or spasms. Or, because of poor vision, users cannot easily determine where exactly the pointer is.

For this audience, the **size of the active area** (that is, the area of the screen that needs to be clicked with the mouse) and the **availability of a visual hover indicator** are important. The size of the click area should be large enough to facilitate the use of the mouse without the risk of activating an adjacent link or button.

For people with low vision, increasing the size of the active element can help identify it more effectively. It should also be noted that people with various motor impairments (and even without them) can click the mouse involuntarily. Therefore, there are rules that allow users to prevent or undo any accidental actions. It is necessary that at least one of the following rules is followed:

1. No event on click: The **key pressed** event is not used to perform any action. That is, if we pressed the mouse key, nothing happens as long as we keep the key pressed.
2. Abort or Cancel: A function terminates during the **key released** event; there is also a mechanism for interrupting the function before completion or canceling the function after completion (i.e. if we accidentally pressed a key, we can move the mouse to the side so that the action does not take place).
3. Reverse action: A **key release** event cancels any result of the previous **key press** event.

These rules and exceptions to them are described in [WCAG 2.1](#), Success Criterion 2.5.2.

## Keyboard instead of mouse

However, not all people can use a mouse, even if the pointing indicator is improved. Blind people cannot see the mouse cursor, so they use the keyboard instead. Also, people with mobility impairments often prefer keyboard control.

That is why [WCAG 2.1](#) requires the availability of all functionality for keyboard control. These requirements are presented in Success Criterion 2.1.1.

Here's how it works: to focus on an element instead of pointing the mouse, use the arrow keys, to activate the element instead of clicking the mouse, use the **Enter key**. Support for control using the keyboard means that all actions on the website can be performed only in this way.

In [Windows](#), [Mac](#), [Linux](#) and other, in particular, in mobile operating systems, control of the interface using the keyboard is provided. Developers of sites and mobile applications should take this into account and support it.

## What's the focus?

Keyboard users navigate the web page using the **Tab** key. Therefore, it is very important that all elements on the page that can be focused with the mouse pointer can receive keyboard focus. If elements do not receive it, it is difficult or almost impossible for keyboard users to activate and interact with those elements.

The following elements should receive keyboard focus:

- link;
- buttons;
- switches;
- flags;
- text entry fields;
- drop-down selection list.

When the focus is moving, it is important to display its indicator so that we can see where the keyboard focus is now. It is also important that the indicator is visible and contrasting.

Browsers usually automatically indicate the element in focus by outlining it with a dashed line (as in [Edge](#) and [Firefox](#)) or a solid line (as in [Chrome](#) and [Safari](#)). The dotted line is somewhat difficult to see, especially for users with low vision. But the default browser focus indicator is considered sufficient for [WCAG](#) compliance, at least in terms of minimum compliance.

However, site developers often turn off this indicator because it may not match the style of the site. At the same time, they do not create their own style, because they do not consider it necessary. In this case, the focus indicator will not be displayed and you will not know where it is now.

Requirements for a visual keyboard focus indicator are given in Success Criterion 2.4.7.

The key difference between keyboard control and mouse control is that you can move the mouse over any element at any time. Whereas the keyboard focus moves sequentially from element to element.

During such navigation, the browser will start the movement of the focus at the top and then go through all the elements linearly until it reaches the bottom. The browser ignores all visual formatting (columns, floats, indents, positioning, etc.) and takes into account the basic order in which elements appear in the website's code.

When keyboard users navigate with it, the order of focus transitions should be clear so they don't get confused. Logical order of focus is also a [WCAG](#) requirement, Success Criterion 2.4.3.

## Full-scale keyboard operation

All actions must be available for full-scale control using the keyboard: activation of a link, button or other element with the Enter key, expansion of collapsed elements with the Enter key, marking of flags with the “space” key, etc.

During interaction, it is important that all actions can be performed using the keyboard. Otherwise, the user may fall into a trap. For example, a window that cannot be closed from the keyboard will open. The user can easily open it, but cannot close it, because the close button does not receive keyboard focus. That is, it is impossible to focus on it, and you can close this window only with the mouse.

Since keyboard users can only scroll linearly up and down the page, it is difficult for them to quickly get to the right place on the page. A typical situation: the user opens the page and starts moving with the Tab key. If the site has a menu with a large number of items and other elements in the header, then in order to get to the beginning of the main content, the user needs to go through all these elements one by one. Only then he will reach the main content of the page.

To make navigation easier for keyboard users, developers can provide a skip to main content tool. This is one or more hyperlinks in the header of the website. They are located first in the order of focus movement. The user can press **Enter** and quickly jump to the main content.

For example, on the website of the Cabinet of Ministers of Ukraine, a link is provided to go to the main content, and at the beginning of it – a link to go to different blocks of the page: news block, services block, etc. Pressing Enter on a link moves the focus to the corresponding block. This is most visible with a screen reader enabled.

To sum up, for full control with the help of the keyboard, the developer must provide:

- ability to focus on all interactive elements using the **Tab** key
- logical order of focus movement;
- focus indicator display;
- possibility of interaction with all interactive elements;
- absence of keyboard traps;
- providing tools to quickly jump to the desired piece of content.

You can check for yourself whether your website is keyboard accessible. Just repeat these steps by opening the website in any **Windows** browser and put the mouse down.

## Keyboard and mobile applications

You may be wondering: *What about mobile apps? Should they be keyboard controlled?* Answer: yes, they should. Although users do not often connect a physical keyboard to a smartphone, but keyboard support is also necessary. Recommendations for keyboard support are relevant not only when using a regular keyboard, but also for any other way of interaction.

For example, the user can control using the keys on the Braille display, a special switch – a device that has one or more buttons (such devices are used by people with motor impairments), or even by voice. And we are not talking about ordinary voice assistants, but about full control. All of these methods will be available if keyboard control is available.

However, it is not necessary to use an actual physical keyboard to test keyboard support on mobile devices. If you don't have it, you can test the availability of the mobile app with **VoiceOver** enabled on **iOS** or **TalkBack** on **Android**. These screen readers are discussed in Chapter 1 of this manual.

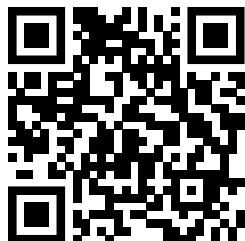
Thus, you need to check that all interactive elements (i.e. buttons, tabs, etc.) are getting **VoiceOver** focus and that double-tap activation is working. Swipe from left to right is used for navigation.

**VoiceOver** indicates the current focus with a frame around the element. This frame is not controlled by the developer, it is added by **VoiceOver**. The task of the developer is to make all elements available for focus. We stop the focus on a certain element and perform a double touch. The button worked – everything is fine. We check other interactive elements the same way.

If screen reader control is available, other control methods (keyboard, switch, Braille display, voice control) are likely to be available as well. If your app passes this test, it will be accessible to all of your users, regardless of whether or not they have a disability.

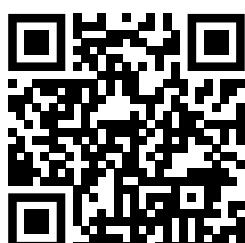
## Supplementary materials to Chapter 4:

1. WCAG: Understanding Success Criterion 2.1.1 Keyboard.



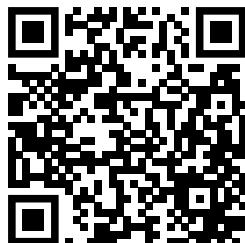
URL: <https://www.w3.org/TR/WCAG21/#keyboard>

2. WCAG: Understanding Success Criterion 2.4.3 Focus Order.



URL: <https://www.w3.org/TR/WCAG21/#focus-order>

3. WCAG: Understanding Success Criterion 2.5.2 Pointer Cancellation.



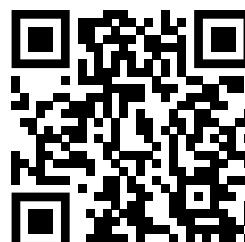
URL: <https://www.w3.org/TR/WCAG21/#pointer-cancellation>

4. WCAG: Understanding Success Criterion 2.4.7 Focus Visible.



URL: <https://www.w3.org/TR/WCAG21/#focus-visible>

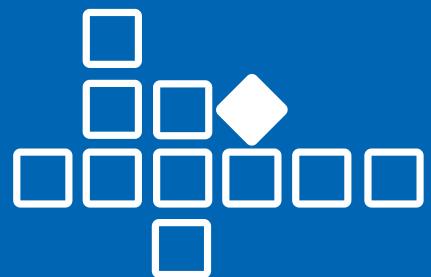
5. Creating links to skip content blocks.



URL: <https://webaim.org/techniques/skipnav/>

# Chapter 5

## Structure and semantics



Let's consider how to structure the content so that each user can quickly find the necessary information. So, let's examine the content structure and semantics.

### It all starts with the name

Imagine that a file named *document123.docx* is sent to you by mail or brought to you on a flash drive. To find out the content of the file, it is necessary to open it. You opened it and saw the heading: *Report on Monitoring Results*.

Now it is clear that we are talking about some kind of monitoring and the document contains a report on the results of its conducting. However, it is desirable that the title was even more informative, as it is currently not clear from it exactly what was examined. The title, which already allows understanding what the document is about, is, for example, *Web accessibility of the Websites of Government authorities. Report on the Results of Monitoring*.

An accurate and apt title will be convenient for everyone. This applies not only to the file, but also to the site Chapter, the name of the mobile application, the subject of the email, etc.

This is especially important for people with cognitive impairments and blind users. The latter, using screen readers, will first hear the name of the file or the title of the page they opened. Therefore, you should be very careful when choosing names.

Page title requirements are provided for in [WCAG](#) Success Criterion 2.4.2.

According to DSTU EN 301 549:2022, this requirement applies to web pages and electronic documents, but not to software such as mobile applications. That is why “empty” is indicated in criterion 11.2.4.2 of the DSTU standard.

We remind you that the numbering of the [WCAG](#) success criteria is preserved in the specified DSTU. Only the Chapter number is added before the success criteria numbers. That is, to find out how Success Criterion 2.4.2 is applied to websites, electronic documents and mobile applications, it is necessary to review subsections of DSTU 9.2.4.2, 10.2.4.2 and 11.2.4.2, respectively.

## Definition of language

Another thing that is very important for understanding the text in a document or on a website is the **programmatic definition of the language of the document or page**. This applies to users of screen readers, as well as people who translate text using automatic tools.

Most screen readers can read in several different languages. Users select a default language when installing and configuring a screen reader. Therefore, if no language is specified in the document, the screen reader will read the document in the default language.

Users who only speak one language will probably only visit websites in their native language, so there won't be any language problems. On the other hand, users who speak several languages, such as Ukrainians, are more likely to visit websites in several languages and want each language to be read with the correct pronunciation and accent.

Thus, when you create a document, make sure that the language is set correctly. In Word in particular, always make sure you have the correct keyboard layout enabled. If you enter, for example, Ukrainian text with a Russian keyboard layout, [Word](#) will mark this text as Russian. If [Word](#) marks correctly spelled words as misspelled, this probably indicates that the language of the text is not correctly defined.

For web pages, the language is set using the `lang` attribute. For the Ukrainian language, you should set `lang=<uk>`, for English – `lang=<en>`. Even if you don't know [HTML](#), you can easily find this attribute. Open the page's code view in your browser ([\*\*CTRL+U\*\*](#)) and look at the first few lines of code.

The correctness of the language attribute is also determined by automatic testing tools. However, the tools can only determine if the correct language attribute is being used (for example, “**uk**”, not “**ua**”). However, these tools cannot determine the actual language of the page and whether the language attribute corresponds to the actual language. More about testing is explained in the next Chapters of this manual.

Requirements for the language of the page or document are included in Success Criterion 3.1.1, and for the language of individual parts of the content – 3.1.2.

## Semantics and headers

When a screen reader reads a page, not only the text is read, but also the purpose of the elements. That is, the screen reader says “header”, “link”, “button”. In this way, the screen reader also announces the assignment of elements in the **Word** document. The situation is the same in mobile applications.

---

The screen reader knows the purpose of the element thanks to the **semantic structure of the content of the document, web page or application**. Semantic structure is the basis of accessible markup. It is thanks to semantic markup that browsers can analyse the purpose of elements and provide this information through accessibility **APIs** using assistive technologies such as screen readers.

To understand the importance of semantic markup, consider the following example. You have two ways to design a heading in **Word**:

- 1) to choose a large and bold font;
- 2) to choose a header style of a certain level. Visually, it may look almost the same, but the screen reader will only say “heading” in the second case, since only the second heading is the real one.

It works the same way when composing website articles, website pages, or mobile app screens. In order for a screen reader to be able to read the assignment of an element, these assignments (or roles) must be defined programmatically.

Headings are the basis of the textual content structure. They play an important role in the accessibility of any document, be it a **Word**, **PDF** or **HTML** document.

Headings are used for screen reader navigation and provide a clear document structure that is useful for all users as it makes content easier to view. And it is important that for **Word** documents, correctly designed headings make it very easy to automatically generate **Table of Contents** (TOC).

There is also an additional benefit: headings improve a website's **Search Engine Optimization** (SEO) by highlighting important parts of a web page and drawing attention to them. However, headings should not be abused as a way to add searchable keywords to your content. Headings that contain more search terms than meaningful structural information about the page will not be useful for screen reader users and will hinder their use in navigation.

Headings should convey a clear and precise structure of the content chapters of a document or web page. When creating documents, we recommend first preparing the structure, and then design the headings according to the structure. Heading levels should always be chosen based on structure rather than appearance, as screen readers and other assistive technologies focus on structure rather than look and feel. As an example, we will give the structure of article titles in Wikipedia.

The correct structure of headings on a web page is when there is only one first-level heading and that is the title of the article, chapters are indicated by second-level headings, subchapters by third-level headings, and so on. In Word, the rules are slightly different: here, each chapter of the document can have a first-level heading, and then the levels decrease according to the structure, as in web pages.

## Formulation of hyperlinks

Screen readers report the link and then read its text. You don't have to do anything special to make it work. However, there are some best practices to be aware of.

The link should have a text name associated with it so that screen readers can read something to users. This seems obvious, but sometimes images or icons are used instead of text. In order for screen readers to read such links, you need to add alternative text or a hidden label to the image.

If the link doesn't have an associated title, most screen readers will read the target **URL** to give users some clue as to what the link is for. Note that this cannot replace text, as **URL** is hard to hear. A link should always have a text label.

A link that opens in a new window or tab should have an appropriate indication.

- !** It is important that the text of the link is understandable even out of context. That is, if screen reader users look at a list of links on a page, they will not always understand the purpose of these links. To make the links as clear as possible, it is not recommended to use general titles such as "*Read more*", "*Learn more*", "*More*". Instead, you should use things like "*Read more: About the company*", "*Learn more about the company*", and "*Show more accessibility news*".

We will remind you that we provided some tips on visual marking of hyperlinks in Chapter 3 of this manual.

# Creation of tables

Screen reader users can navigate tables almost like **Excel** spreadsheets, moving from one cell to another in any direction. If the table's markup is correct, a screen reader will read aloud not only the data in the cell, but also the column or row heading, allowing blind users to understand where they are within the table. If tables are not properly marked up, screen reader users will likely have difficulty understanding the information presented.

Here are some rules for creating accessible tables:

## 1. Do not create pseudo-tables.

Tables have a drawback: they don't always display well on mobile devices. This drawback forced developers to look for alternative ways of presenting tabular data. What many of these methods have in common is that they use the `<div>` element and **CSS** to render the tables without a corresponding semantic structure.

The problem is that this impairs the ability of screen readers to convey semantic meaning and structure to blind users. It also impairs the ability to navigate table cells in different directions. Because the semantics and navigation are broken, users can only move through the pseudo-table sequentially and in the order in which the elements are located in the page code, which is a much less efficient way to explore tabular data.

## 2. Table column or row heading text should accurately describe the data in the corresponding column or row.

If the text in the title cell is vague (for example, "Column 2") or contains extraneous information (for example, links, buttons, or additional descriptions other than the column name), it can confuse screen reader users. It's best to keep your headline text to the point and simple.

## 3. Avoid nested tables.

Nested data tables are tables that are inserted inside other data tables. Nested tables break the accessibility of the data presentation. Screen reader users will not be able to understand the structure of such a table.

Thus, how do screen readers read nested tables property? A screen reader notifies users that a table is present and tells them how many columns and rows it has, for example: "A table with four columns and five rows." After hearing this, the user either decides whether to continue reading all the cells as usual, or can go cell by cell (this is called "**table navigation mode**").

In normal mode, the screen reader starts reading from the upper left cell of the first row, reads the cells of this row one by one, then moves to the next row, reads it from left to right, and so on in all rows. In the table navigation mode, the user can move in any direction using the appropriate keyboard combination in the screen reader. In **Windows** screen readers, this is usually **Control + Alt + Arrow keys**.

Thus, the content should be structured in such a way that its structure is as clear as possible for everyone. Semantic requirements are given in Success Criterion 1.3.1.

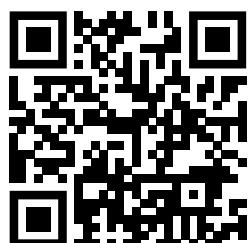
## Supplementary materials to Chapter 5:

- 1.** WCAG: Understanding Success Criterion 1.3.1 Info and Relationships.



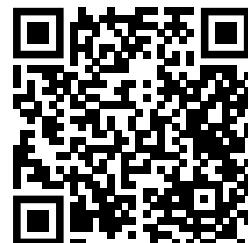
URL: <https://www.w3.org/TR/WCAG21/#info-and-relationships>

- 2.** WCAG: Understanding Success Criterion 2.4.2 Page Titled.



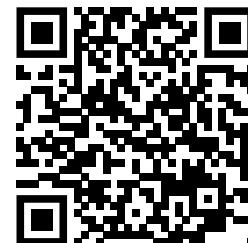
URL: <https://www.w3.org/TR/WCAG21/#page-titled>

- 3.** WCAG: Understanding Success Criterion 3.1.1 Language of Page.



URL: <https://www.w3.org/TR/WCAG21/#language-of-page>

- 4.** WCAG: Understanding Success Criterion 3.1.2 Language of Parts.



URL: <https://www.w3.org/TR/WCAG21/#language-of-parts>

# Chapter 6

## Fill out forms



Forms are the foundation of any digital service. If the form is unavailable, the service will also be unavailable. That is, one of the users will not be able to order the service. Thus, let's consider the issue of the accessibility of forms.

### Visual accessibility

In order for all users to know how to fill out the form and be able to do so, it must be visually and non-visually accessible (using a screen reader). This is mostly easy to do through:

- the definition of labels for form fields;
- providing instructions and hints where necessary;
- error prevention;
- checking that the form is filled in correctly.

To better understand the issue of form accessibility, you should understand how users fill them out. Accordingly, two main methods can be distinguished: visual and non-visual

.

**Visual** is the familiar way for most when the user sees the screen and uses the mouse, namely: the user sees a field, clicks the mouse, a cursor appears in the field and the user starts typing. Next, the user clicks on the next field and fills it in the same way. During visual access, it is important that the active area (the part of the interface that can be clicked with the mouse) is large enough and that labels, instructions, and error indicators are clearly visible.

For a screen reader to work, it's important that it reads all labels, instructions, and error messages, and that all form elements are accessible from the keyboard. Since users of screen readers usually do not use a mouse.

## Work with forms

Let's consider in more detail how to work with forms using the **NVDA** screen reader as an example. Most screen readers have two main modes for viewing pages: normal reading mode (or document mode, **Browse**) and form mode (or focus mode, **Focus**). Each of them has a specific purpose, meaning you can only be in one mode at a time. The purpose of the different modes is to allow users to perform a specific set of operations that correspond to the type of content or type of interaction.

Let's focus on how screen readers interact with forms and what are the differences between general reading mode and forms mode. In reading mode, users can read text, use the arrow keys to navigate through lines, words, and even symbols to understand the spelling of complex words. This is the standard mode of operation of most screen readers.

When users want to navigate semantic elements such as headings, landmarks, tables, graphs, lists, links, etc., screen reader keyboard shortcuts are used to move forward or backward between these elements. Thus, to switch between headings you need to press the letter "**H**".

Work in forms mode is significantly different. When in a text field on a form, users want to be able to enter text in that field. Therefore, screen readers must disable all keyboard shortcuts related to letters and numbers to allow users to do this.

For example, the "**H**" key usually takes you to the next heading. Then what should a user do when entering text in a form field if he needs to enter the letter "**H**" but doesn't want to go to the next heading? To do this, **NVDA** automatically switches from reading mode to forms mode, disabling normal keyboard shortcuts so that users can type the letters and numbers they want.

When you enter text into a text field in forms mode, screen readers speak the letter you type (e.g. “**H**”) and allow entering the character. Screen readers usually automatically switch to forms mode as soon as you move focus to a form field with the tab key. Forms mode is not activated when simply reading form content in normal reading mode.

Some screen readers allow users to manually switch between form mode and document mode. In **NVDA**, for example, this is implemented by pressing **Insert + Space** or **Caps Lock + Space**.

While in a form, most users use the **Tab** key to move between the elements they can focus on. Text such as paragraphs, headings, etc., does not get focus when using tabs. Because of this, users can completely skip such unfocused text (form instructions, formatting tips, and headings to group Chapters of the form). And while users can always switch to reading mode to hear these things, the likelihood of that happening is very low. Therefore, developers shouldn't expect users to ever hear any out-of-focus text in a form.

Regardless of how you fill out the form (visually or non-visually), it is also important that the text of labels and prompts is clear. That is, so that users, regardless of the level of education and intellectual abilities, understand exactly what information needs to be entered. Therefore, we will consider in detail all the components of the available form.

## Label – a component of an available form

All form fields and control elements must have a label – a text label that is understandable to all users, whether they see the page or not.

Let's consider the basic requirements for labels. They should be:

- meaningful, i.e. clearly explain the purpose of the field;
- programmatically associated with the corresponding control so that the screen reader reads only the label that applies to a certain field;
- always visible, that is, they should not disappear after entering data into the field;
- located in close proximity to the corresponding field or control.

**!** **Nota Bene!** If the labels are placed as placeholders (or placeholder text inside the field), then after filling these fields, such labels disappear. And therefore, you will not be able to control yourself.

Developers often use placeholders instead of tags to save space. Whereas real labels, i.e. created using the **label** tag, increase the active area.

You don't have to click on the field to activate it. Clicking on the label will also activate the field to start typing. This solution is the best in terms of accessibility. Such a decision is also good. The label is formatted so that it looks like placeholder text, but is a real label. This example demonstrates a solution where a programmatically linked label is taken to a field and made to look like placeholder text. When the input field gets focus, the label moves up using **CSS**. When the focus leaves the field, the label remains at the top of the field if there is text in the input field, and returns to display as placeholder text if not.

Therefore, the field label is always visually displayed on the screen – either as placeholder text or at the top of the text field. That is, the label never disappears. Even if a label is visually next to a field, but not associated with it programmatically, the browser will not identify it as a label for the corresponding field, so it will not pass this connection to the screen reader, which in turn will not read the text of the label. Therefore, you should ask the developer to programmatically associate fields with labels. If he/she does not know how to do this, then a link to detailed advice [link #5] is provided in supplementary materials for this Chapter.

## Instructions facilitate filling out

One of the easiest ways to **make forms more accessible** is to help users avoid errors before they happen. In order for users to be able to fill out forms without errors, it is important to provide not only clear labels, but also instructions and other useful information.

Here's what to consider when it comes to instructions for forms:

- whether labels and instructions are clear and informative;
- whether instructions and additional information are provided for assistive technologies (when appropriate);
- whether there are restrictions on entering data into the form fields;
- whether mandatory form fields are defined.

As mentioned above, placing plain text inside a form runs the risk of not being heard by screen reader users. If users navigate only between focusable elements, they will skip all non-focusable text, unless that text is programmatically associated with a control or field.

We will not go into the technical details here. If necessary, you can find a detailed description in the article, the link to which is placed in supplementary materials for this Chapter.

**Instructions should be specific and meaningful** so that users know what to do to fill out the form field correctly. If a form field requires a specific format or other specific instructions, website administrators who host the form should notify users in advance. This will help avoid errors before they occur. For example, you can specify the correct format of the phone number, dates of filling, etc.

All mandatory fields must be **marked accordingly**. Clearly defining of mandatory fields helps prevent errors and reduces the chance that users will not fill out all the required fields in the form.

The general requirements for defining mandatory fields are as follows:

- information must be accessible to a screen reader. For this, you should set the special attribute **aria-required="true"**;
- an additional visible indicator must be available for sighted users. Or you can use the **required** attribute. In this case, the browser will not allow sending the form with an empty field and will show a corresponding notification.

It's easy to make standard form elements accessible, that is, add appropriate labels, descriptions, instructions, etc. However, accessibility is becoming increasingly difficult as web design becomes more complex and innovative. And it deviates more and more from standard elements of forms and design.

The good news is that dynamic forms and custom form elements can also be made accessible if built correctly. There is a specific set of **WAI ARIA** roles and attributes. It provides a toolkit that allows making any complex interactive element accessible. References to **WAI ARIA** technical documentation can be found in supplementary materials.

## Checking the completed form

However, no matter how good the instructions are, there can still be errors in the forms. An important aspect of form accessibility is how to handle such errors.

Here are some points to consider when checking the form:

1. Notification to users of any input errors that prevent form submission, or notification if the form submission was successful. If an error occurs and the user doesn't receive any notifications from their screen reader, that's a bad experience. It is also a problem if a sighted user cannot visually see what errors have been made in the form.
2. Saving as much as possible of correctly entered information.
3. Displaying error notification in context for quick fixing.
4. Giving users enough information to correct the error.

One way to provide feedback to users after submitting a form is to display the result of the form submission at the top with a success or error message. Another way to help users correct form validation errors is to move focus to the first field that contains errors after submitting the form. With this technique, the user does not need to search for the first error, he can go through the form from the first error to correct other errors.

There is also a way to check using the browser. This built-in validation can be used to tell users whether or not they are filling out a form field correctly before submitting it. For example, if the field has the type "**email**", the browser will allow sending the form only when the **email** has the correct format.

Requirements for the accessibility of forms are given in the **WCAG** Success Criteria 3.3.1-3.3.4.

## Supplementary materials to Chapter 6:

1. WCAG. Understanding Success Criterion 3.3.1 Error Identification.



URL: <https://www.w3.org/TR/WCAG21/#error-identification>

2. WCAG. Understanding Success Criterion 3.3.2 Labels or Instructions.



URL: <https://www.w3.org/TR/WCAG21/#labels-or-instructions>

3. WCAG. Understanding Success Criterion 3.3.3 Error Suggestion.



URL: <https://www.w3.org/TR/WCAG21/#error-suggestion>

4. WCAG: Understanding Success Criterion 3.3.4 Error Prevention (Legal, Financial, Data).



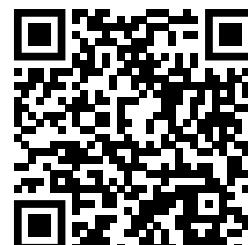
URL: <https://www.w3.org/TR/WCAG21/#error-prevention-legal-financial-data>

5. General accessibility of forms.



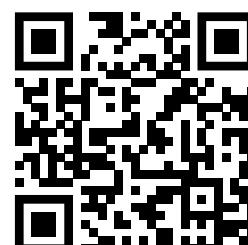
URL: <https://webaim.org/techniques/forms/#:~:text=Accessible%20forms%20are%20easy%20to,order%20are%20logical%20and%20intuitive>

6. Convenient and accessible form validation and error recovery.



URL: <https://webaim.org/techniques/formvalidation/>

7. Accessible Rich Internet Application (WAI-ARIA) 1.2.



URL: <https://www.w3.org/TR/wai-aria-1.2/>

# Chapter 7

## Comprehensibility of the text



In the previous Chapters, we have already noted that text is the basis of accessible content, as it is easy to adapt it to different needs or turn it into any other form of content presentation. For example, blind users can listen to the text using a screen reader or read Braille on the display, and users with low vision can enlarge the font. However, some users have difficulties with reading and understanding the text. In this Chapter, we will consider how to make the text accessible to a wide audience.

### Simplifying reading

Many people with cognitive disabilities have difficulty reading. Some people read more slowly than their peers, and some people can't read at all. These users can be helped by supplementing the text with illustrations, video or audio, as well as writing the text in plain language.

Some people with dyslexia have high levels of cognitive functioning at the conceptual level. But visual encoding of text can be difficult for them because they have difficulty writing or reading words.

Dyslexia makes it difficult to recognize written or printed words and associate them with the corresponding sounds. This is called **text decoding**. Decoding should be automatic so that people can read freely. The process of decoding the text in the “word by word” mode is accompanied by the expenditure of a large part of the mental energy needed by most people to understand what they read. Text that uses short common words and short sentences is easier to decode. It usually requires less reading skill than a text with long sentences and complex or unfamiliar words.

People with dyslexia usually describe their reading experience as letters moving, crossing each other and switching places.

The following are guidelines for presenting text to improve readability for users with reading difficulties:

1. Supplement the text with illustrations, video or audio.
2. Avoid the highest level of contrast between the text and the background.  
Don't use high-contrast text (e.g. black on white), but make sure the contrast remains within the range acceptable for people with low vision. We remind you that the minimum ratio is 4.5:1. It is better to use gray text on an off-white background.
3. Do not use text alignment by width.

When enlarging such text, large blank spaces may appear between words, which will prevent some users from sticking to the line.

The given recommendations relate to the presentation of the text itself. At the same time, accessibility is also affected by the complexity of the text.

## Facilitating the perception of the text

Success Criterion 3.1.5 in **WCAG 2.1** requires that the text either be comprehensible to people with lower secondary education, or have an alternative version that does not require reading skills higher than lower secondary education. The success criterion is **AAA** level, so according to the DSTU EN 301 549:2022 standard, it is not mandatory, but only recommended.

This success criterion helps people with reading disabilities and also allows authors to publish relatively complex content. The complexity of the text is described in terms of the level of education required to read the text. Levels of education are determined according to the International Classification of Education developed by UNESCO to enable international comparison of education systems.

Complex text can be suitable for most representatives of the target audience. But there are people with cognitive disabilities, including those with reading disabilities, even among highly educated people with special knowledge in a certain subject. The needs of such persons should also be taken into account. If the text cannot be made more readable, additional content is needed. This is usually a separate version of the material in plain language.

According to the [Open Society Mental Health Initiative](#), the **Easy-to-Read** concept cannot be universal: it is impossible to write a text that will suit the abilities of all people with literacy and comprehension problems. However, it is highly desirable to use as clear and simple language as possible.

## Recommendations from the International Plain Language Federation

**1. Audience and purpose.** Start by thinking about what goal you want to achieve with your writing. How do you imagine the situation when the goal is achieved? How can you measure it?

After that, you should think about the following:

- who are typical representatives of your audience (age, profession, education and skills);
- what information they may already know and what they may need;
- what they want to achieve by reading your text.

Assessing your audience will help you choose the best combination of language style and design.

**2. Structure.** To choose the right communication structure, ask yourself: what are the typical structures used for this type of communication; which structures will be familiar to readers; which sequence will be logical and convenient for readers.

For example, a procedure description is best organized chronologically, while a longer report may contain a series of topical chapters.

What you should pay attention to when improving the structure:

- use of a generalized presentation of key information (summary) before details;
- division of each part into approximately equal chapters;
- adding some transition text to link one chapter of text to the next.

**3. Design.** Currently, the visualization of the text is becoming more important than the structure and language. Therefore, three aspects should be considered here: layout, typography and information graphics.

To design your text to make it as accessible as possible, consider the following tips:

- a. Make sure the layout and margins contain plenty of white space, as this helps separate different parts of the text.
  - b. Use meaningful headings to help readers navigate.
  - c. Choose a font and its size that will be easy to perceive.
  - d. Make sure there is enough contrast between the text and the background, and that the spacing between lines and paragraphs is at least the size of the font.
  - e. Consider using visual aids such as tables, charts, photos, and bulleted lists to make the information more accessible and attractive.
- 4. Formulation.** Think carefully about the wording. Although some techniques differ from language to language, the main aspects to consider are as follows:
- tone: use personal pronouns and write in a formal but at the same time conversational style;
  - vocabulary: use the simplest word that conveys the desired meaning;
  - jargon: avoid jargon unless your reader is familiar with it. Also explain technical terms new to him;
  - syntax: formulate sentences with an average of 15-20 words, and separate sentences – no longer than 30-35 words.

Once you've laid out the information in plain language, the last step is to make sure your audience can understand and use the information correctly.

- 5. Evaluation.** Always evaluate whether your writing will be successful before publishing it. This may involve editing the text according to a standard or checklist, or having another person review it.

Wherever possible, test your text with your audience. This can be done through surveys, interviews or testing with individuals who represent your readers. User testing may not always be necessary or cost-effective, for example in the case of internal work communication via e-mail. But it is important for a document that has a wider audience, such as a user guide or a public report.

These guidelines will help you write accessible text, but even simple language will be inaccessible to those who cannot read. For people who cannot read text, it is preferable to provide information in other formats (images, audio or video) whenever possible.

## Emotions in the text

We will separately consider the issue of using emoji. This issue is not controlled by accessibility standards, and there is no consensus among users about the use of emoji in text.

Hello! 😊 ❤️ 😊. I'm going ✈️ for a trip this summer 🎪. If you want – join me 😍 🌟 ❤️



Example of a text with emoji

That is true, everyone loves emojis. These are small emotional images that can say a lot and are becoming increasingly difficult to avoid. Thanks to emojis, we have brought back the feelings lost in our online communications. They literally gave a face to our text. However, there is a question: are emojis interpreted and perceived the way we want them to be?

On the one hand, emojis make the text easier to understand because they play the role of illustration and make the text less uniform. In addition, they help to remember information better.

It is important that these symbols are recognized and named by screen readers. On the other hand, screen readers usually read emojis without warning that the text is specifically about emojis. Therefore, users of screen readers cannot always distinguish between emojis and written words until they read the text symbol by symbol. But users read the text by symbols only when they did not understand what was written, and in other cases – they do not read.

For example, the text of the beginning of the news headline “🤡 Vasyl Vasylenko, Deputy Head of the Department, stated...” the screen reader will read as follows: “Clown Vasyl Vasylenko, Deputy Head of the Department, stated...”

If there are many emojis in the text, it is very difficult for screen reader users to hear the text itself. The problem is aggravated by the fact that the authors do not always accurately understand the meaning of the emoji symbols, and the screen reader reads their true meaning, so in this case the incorrect meaning may be conveyed.

For example, authors often put an emoticon that screams with fear in front of important messages, and for users of screen readers it sounds like this:

“*Screaming with fear, we suggest a new offer for you.*” Authors also often use the Braille dot symbol as a bullet in lists, and screen reader users cannot understand what Braille is for. And this, of course, is not a complete list of symbols that authors often use in a non-literal sense.

For people with low vision, some emojis can look very similar. For example, pensive, relieved, and disappointed faces have only small differences that are often imperceptible when these emoji are displayed on a smaller screen or viewed by a visually impaired person.

However, all this does not mean that emojis should be abandoned, because they are convenient and useful. The main thing is to follow some simple recommendations:

1. Use emoji only where appropriate.
2. Don't use too many emojis in one sentence or line of text. One or two characters is enough.
3. Use emoji symbols only literally, in the meaning that the developers put into them.
4. Avoid ambiguous emojis. Remember that some symbols may have different meanings in different cultures.
5. Don't use emojis as a substitute for text. Text from which emojis have been removed must retain meaning.

Thus, write easy and clear texts! After all, now you know how to do it.

## Supplementary materials to Chapter 7:

1. WCAG: Understanding Success Criterion 3.1.5 Reading Level.



URL: <https://www.w3.org/TR/WCAG21/#reading-level>

3. Emoji and accessibility.



URL: <https://www.tiny.cloud/blog/emojis-and-accessibility/>

2. What is plain language.



URL: <https://plainlanguageagenetwork.org/plain-language/what-is-plain-language/>

## Chapter 8

# Harmful and complex design elements



There are design elements the presence of which in content makes it completely unavailable, even if the content meets other criteria. If a web page, electronic document, or mobile application contains flashes, background sound, or motion (such as animations, background video, parallax effects, or transitions), these can become accessibility barriers.

### Flashes and their consequences

Thus, in some people, flashes can trigger seizures. People with vestibular disorders will feel dizzy if the movement on the page seems too intense for them. People with attention deficit disorder can become overly distracted. In addition, when there is background video sound, it can be difficult for people using screen readers to hear their own screen reader.

Some people are sensitive to flashing lights, such as strobe lights, and may experience seizures known as photoepileptic seizures as a result. Seizures, while causing physical damage, are dangerous, making them a serious health problem that goes beyond basic accessibility.

In particular, creators of multimedia materials must be especially careful not to create visual effects that can cause convulsions in viewers.

Seizures can be caused by many things, including brain injury, dehydration, lack of sleep, and even flashing lights. This last category should worry content creators.

One of the best-documented cases of flashing lights causing seizures was a 1997 Pokemon cartoon that sent 685 children to the hospital. They had convulsions as a result of watching an intense scene with flashing lights. This scene caused people to have seizures right in the cinema.

## Making flashes safe

To prevent this from happening, **WCAG** requirements should be followed. These requirements are described in Success Criterion 2.3.1 (a mandatory AA level criterion according to DSTU) and 2.3.2 (optional as it is AAA level criterion). According to the mandatory requirements (Success Criterion 2.3.1), content should not flash more than three times per second unless the flashing content is sufficiently small. And the flashes are low contrast and do not exceed the general flash thresholds or the red flash thresholds.

Content that flashes or flickers more than three times per second may trigger a seizure in people with photosensitive epilepsy or other types of photosensitive seizures. If content that flashes more than three times per second is small enough or below certain contrast thresholds, the risk of provoking an attack is greatly reduced.

Therefore, content that flashes more than three times per second is allowed if one of the following conditions is met:

**1.** The flash field is small enough.

What does it mean? **WCAG** defines “sufficiently small” as “less than 25% of 10 degrees of the visual field (representing the central visual area of the eye)”.

Calculating this parameter for all possible screen sizes would be time-consuming, so W3C also formulated a rule of thumb to determine a safe threshold for users on a wide range of screen types.

The recommendation boils down to this: *“...any single flash event on the screen (with no other flashes) smaller than a continuous area of 21824 square pixels (of any shape) will satisfy the general and red flash thresholds.”* For example, a flashing area of 200 x 100 pixels would satisfy this condition.

**2.** The contrast of flashing content is low enough.

How to determine which contrast is low enough? As with the maximum size of flickering content, **W3C** provides a formula for calculating contrast in its definition of “Common Flash and Red Flash Thresholds”. References are provided in the supplementary materials.

In this case, [W3C](#) recommendation is to use a flashing content evaluation tool such as [The Trace Centre](#). The [Photosensitive Epilepsy Analysis Tool \(PEAT\)](#) is free software that can be downloaded and used to identify seizure risks in web content and software. A link to download this tool is also provided in the supplementary materials.

While small flash areas or very low-contrast flashes are considered acceptable by accessibility standards, they can still be annoying and distracting at times. This can be a problem for users with attention deficit disorder or cognitive impairment. Therefore, to keep consumers of your content safe, it is recommended that you consider removing all flashes.

## Parallax effect

Even content that does not contain flashes may cause some users to experience adverse health effects, including dizziness, nausea, or disorientation. It's about animation and movement.

Parallax effects refer to scripted visuals that respond to user actions. This is, for example, scrolling or moving the mouse, which causes some elements on the page to move differently than other elements. That is, the background image can move more slowly than the upper text. Or the background image can remain completely static and still, while the text scrolls over it.

The visual effect is striking mainly because it is somewhat unexpected and new. At the same time, this effect has some disadvantages. People who are prone to nausea, dizziness or vestibular disorders may experience these negative symptoms when viewing parallax effects. The more parallax effects there are on a page or app, and the more extreme they are, the greater the risk that people with vestibular disorders will experience problems.

A bad example: the parallax effect on the [iPhone](#)'s home screen. Some mobile devices with motion detection technology include parallax effects in the operating system. On the iPhone, the background image of the home screen moves up and down or side to side when a person tilts the device. Icons stay in place, although some icons have elements that move slightly with the background.

The visual effect is caused by three-dimensional graphics and active movement. It was all the rage at the time, but when the feature came out, some people started complaining. Apple has updated the operating system to allow people to turn off this feature ([Settings](#) → [Accessibility](#) → [Motion](#) → [Reduce Motion](#)).

To reduce the negative impact of this effect, the following recommendations should be followed:

- a) parallax effects should be minimized in terms of the total number of effects, the proportion of parallax within each individual effect and the size of the influence zone;
- b) all content and content features with parallax scrolling must be keyboard accessible. That is, if some part of the content is hidden, access to it should be implemented using not only the mouse, but also the keyboard;

- c) the contrast of the text against all parts of the moving background should be at least 4.5:1 for small text or 3:1 for large or bold text. These are standard contrast requirements, but should be kept in mind when creating parallax effects, as the background image may be non-uniform.

## Background video, animation and sound

Besides parallax, there are other effects that can create barriers for the user. For example, background video, animation or background sound. These items may be available if certain guidelines are followed.

1. Important content should not be conveyed through background video and animation unless users have full control over playback and full access to subtitles, transcripts and audio descriptions.

The general idea behind background videos is to provide a visually appealing and slightly unusual decoration to web content. The background video should be considered additional, not the main information. But if it has already happened that important information is transmitted only in the background video, then it should be fully accessible – with subtitles, transcription and audio description.

2. A method should be provided to pause, stop or hide any background video or animation that starts automatically and lasts 5 seconds or more.

It's best to prevent content from automatically playing on page load. Instead, you should allow users full control over when content or audio content starts and ends. As a bit of a compromise, [WCAG](#) at **AA** level allows for auto-playing videos up to 5 seconds long.

But if it takes longer to play the content, a mechanism must be provided to pause, stop or hide the content. Requirements for animations and automatic playback are given in [WCAG](#) Success Criterion 2.2.2.

3. The contrast of the text against the background of the video must meet the contrast requirements.

The contrast of text on a video or animation background, as well as on a static background, should be at least 4.5:1 for small text or 3:1 for large or bold text.

4. Movement in the background video and animation should be minimal.

Any movement on the web page attracts the user's visual attention. This can be good if you're trying to get the user to focus on something. However, by definition, a background video or animation should be background content, so it shouldn't be the focus. With this in mind, any movement in the video should be minimal and unnoticeable.

5. Background video and animation should not contain sound.

Although Success Criterion 1.4.2 allows automatic sound playback for up to three seconds, it is better not to include sound in background videos and animations. This forces users to listen to audio, which defeats the purpose of having a video or animation in the background.

Users of screen readers are faced with the fact that the video makes sounds during reading of the text, which can be disorienting and annoying. If the video is too loud, it can even muffle the sound of the screen reader. Screen reader users can't see where the video is located, so they have to look for the video player controls to mute the sound (if there are any). It may take a long time to find the controls. So, the user will be disappointed and have a bad experience with the web page. Fortunately, with time, such examples are more difficult to find.

## Complex interface components

There are things that do not cause health deterioration, but can create a serious barrier for users of screen readers or those who use a keyboard instead of a mouse.

**Complex interface components** are carousels, sliders, drop-down menus, pop-ups, tab panels, etc. They are present on almost any modern website, but they can lead to accessibility barriers.

For example, a slider may not be available for keyboard switching, form tooltips may not be announced by a screen reader, etc. In some cases, users can get trapped when the focus goes to a certain component, and it is impossible to get out of it.

A common problem is when, after typing in a search query field, tooltips appear below it. But when pressing the down arrow, the screen reader does not read these tooltips.

However, even complex components can be made accessible. This means that they should not be abandoned. These types of interactive elements require special attention during development and testing to ensure accessibility.

The developer has two ways:

1. Use only standard browser components – standard selection lists, buttons, radio buttons, etc. They are guaranteed to be available to all users, including screen reader and keyboard users.
2. Develop your own components if you need something more complex that goes beyond standard components. At the same time, attention should be paid to accessibility during development and testing with assistive technologies.

You must ensure that control of these components is accessible from the keyboard, and that the screen reader announces the names, roles, and states of the components. For example, a screen reader announces the name of a component, its role (“**tab**”), and state (“**highlighted**”). That is, everything that a sighted user sees can also be heard by a blind user.

To make complex components accessible to assistive technologies, there is a special markup – **WAI ARIA**. The studying of it requires basic knowledge of **HTML**. If you have them, you can learn about **WAI ARIA** at the relevant link provided in the supplementary material for Chapter 6 of this manual.

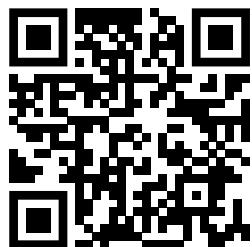
## Supplementary materials to Chapter 8:

1. Information on photosensitive seizure disorders.



URL: <https://trace.umd.edu/information-about-photosensitive-seizure-disorders/>

2. Photosensitive Epilepsy Assessment Tool (PEAT).



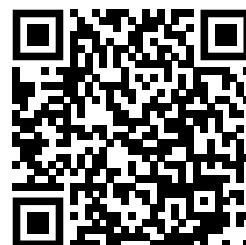
URL: <https://trace.umd.edu/peat/>

3. Banned Pokemon Episode That Gave Children Seizures.



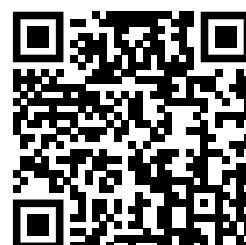
URL: <https://kotaku.com/the-banned-pokemon-episode-that-gave-children-seizures-5757570>

4. WCAG: Understanding Success Criterion 2.2.2 Pause, Stop, Hide.



URL: <https://www.w3.org/TR/WCAG21/#pause-stop-hide>

5. WCAG: Understanding Success Criterion 2.3.1 Three Flashes or Below Threshold.



URL: <https://www.w3.org/TR/WCAG21/#three-flashes-or-below-threshold>

# Chapter 9

# Web accessibility testing



In the previous chapters we discussed various components of accessibility and, to some extent, how to test these components. This chapter outlines a comprehensive approach to accessibility assessment.

## Accessibility assessment methods

Automated and manual testing should be highlighted among the main evaluation methods. Automated testing is checking the website for accessibility using special plugins, such as [Wave](#), [Axe](#), [Lighthouse](#), [Accessibility insight](#), etc. This also includes checking of electronic documents using built-in [Microsoft Office](#) and [Adobe Acrobat DC](#) tools.

The main advantage of the automated method is speed: you just press a button and get the result. And automated testing, unlike manual testing, is unbiased, as its results are not influenced by the human factor.

Automated accessibility tools can detect many content accessibility issues, such as images without alternative text, contrast, fields without labels, and more. Such tools are an important part of the overall accessibility assurance and quality control process. Even so, automated tools cannot detect all potential accessibility issues.

## Automated vs manual testing

There are certain types of problems that are difficult for automated tools to identify:

- **Quality and accuracy:** Inaccurate or low-quality alternative text is no better than no alternative text. Automated tools can only determine if an image has alternative text. However, they cannot tell you whether the alternative text is accurate or correct. The same applies to text labels for form elements, headings in tables, etc.
- **Keyboard focus order:** Automated tools cannot reliably determine where the keyboard focus should be at any given time. When the user clicks the button, the focus should go to a logical destination (such as a dialog box or error message), but the automated tool cannot determine the appropriate focus location.
- **Widgets** with complex components, such as dialog boxes, carousels, and accordion menus: The automated tool cannot detect all possible ways to create them incorrectly, especially when it comes to keyboard interaction scenarios. At best, an automated tool will be able to determine whether a component is created correctly by comparing names, roles, values, and keyboard focus order. However, even this is not always possible.

It's actually quite difficult to estimate the proportion of accessibility problems that automated tools can detect, primarily because each website is different from the rest. Usually this figure is 30-50%. The more complex the website is in terms of scripts and complex components, the lower the proportion of problems detected by automated methods will be.

We also mention false positives among the disadvantages of automated testing. That is, the plugin can find an error that does not exist or is not critical in this context. For the correct and appropriate application of the automated method of accessibility testing, its capabilities and limitations should be clearly understood.

For reliable testing, manual testing methods should also be used. Such methods involve checking accessibility by a person. This can be done both by an accessibility expert and by ordinary users who belong to groups with impaired mobility. It is best when both experts and ordinary users are involved in testing.

During such testing, all possible content access scenarios are reproduced: control using a keyboard without a mouse; touch screen access; access using a screen reader; increasing the font in the browser, reading and watching videos for people with low vision, etc.

The advantage of the manual method is that you can make sure that your content is actually, not formally, accessible. And the disadvantage is the need for greater material and human resources.

## Availability of complex components

Working with a screen reader can be unfamiliar, so you will have to spend some time learning it. The best way is to appeal to people who use a screen reader on a daily basis.

The only way to know if a complex component such as a dialog box, accordion menu, tab bar works with a screen reader is to test them using a screen reader. This is especially relevant when building components using the special **WAI ARIA** markup mentioned in previous chapters of this manual.

Blind people and people with mobility impairments prefer to use a keyboard instead of a mouse. Therefore, all functions and controls must be accessible from the keyboard.

Keyboard accessibility requires several things. Here are some of them:

- The website should display a visual indicator of keyboard focus. The user should be able to see where the focus is.
- Users should be able to navigate to all controls (links, buttons, form elements, etc.) using the **Tab** key.
- Users must be able to activate all relevant elements with the **Enter** key, and buttons with the **Space** key.
- Mouse pointing functions must be identical to corresponding keyboard functions. That is, if, for example, a menu expands when hovering the mouse cursor, the keyboard user should be able to expand this menu without using the mouse.
- When the popup dialog box appears, it should have keyboard focus. When the dialog closes, the focus should return to the original control or to another logical location.

These are the main things that should be tested manually. Keyboard and zoom testing can be done without special skills, while testing with a screen reader requires an understanding of how it works. Given the advantages and disadvantages of each testing method – automated and manual – to obtain the best result, these methods should be combined.

In general, specific evaluation methods depend on the objective and available resources. Taking them into account, you can conduct a simplified or in-depth assessment of accessibility. For example, to quickly find common errors on a large number of websites or pages, or to monitor changes, simplified evaluation through automated testing is more suitable. Whereas the most important content or complex components are better to be checked manually, i.e. perform an in-depth evaluation.

# Assessment of accessibility of government websites

In November 2021, at the request of the United Nations Development Program in Ukraine, a methodology for the basic accessibility monitoring of government websites was developed. This technique is designed to quickly check a large number of websites by specialists without experience in accessibility issues. Therefore, a simplified procedure, which consists of 70% automated tests and of 30% simple manual tests, was chosen for the monitoring.

Next, we provide a brief description of the testing procedure using this method.

The indicator of basic accessibility includes the following criteria:

**1. Sufficient text contrast.**

As we already know, this is important for people with visual impairments, cognitive impairments, or anyone reading text in bright sunlight, for example.

**2. Availability of alternative text for all images.**

Without it, blind people using a screen reader will not be able to find out what is shown in the picture.

**3. Availability of accessible text for all hyperlinks.**

If the link has no text, but only an icon or image with no alternative text, screen reader users will not be able to understand the purpose of the link.

**4. Availability of text labels for all buttons.**

Buttons (like links) should have text labels, not just icons.

**5. Availability of text labels for form fields.**

This component is important for all users, because otherwise it is impossible to find out the purpose of the field.

**6. Software definition of the main language of the page.**

This is important for the correct operation of screen readers and automatic translation in the browser.

**7. No errors in the page code.**

This ensures that assistive technologies correctly interpret the content of the page.

**8. Providing a mechanism to quickly go to the main content of the page.**

This is critical for keyboard users, particularly blind users.

**9. Keyboard focus marking.**

This is important for sighted keyboard users who need to see what they are focusing on.

**10. Support for keyboard control.**

This component is important for all keyboard users – sighted and blind, with impaired motor skills and those who simply prefer to navigate using the keyboard.

If the criterion is successful, it corresponds to 1 point.

To check criteria 1-7, automated web page testing using the Accessibility Insights extension for Google Chrome browser is used.

Let's evaluate the criteria from 1 to 7. To do this, we will scan a website using the specified extension.

The table below lists criteria 1-7 and the identifiers of potential errors that cause these criteria not to be successful.

Criterion	Error identifier
1. Sufficient text contrast	<b>colour-contrast</b>
2. Availability of alternative text for all images	<b>image-alt, role-img-alt</b>
3. Availability of accessible text for all hyperlinks	<b>link-name</b>
4. Availability of text labels for all buttons	<b>button-name, input-button-name</b>
5. Availability of text labels for form fields	<b>label, select-name</b>
6. Software definition of the main language of the page	<b>html-has-lang, html-lang-valid, valid-lang</b>
7. No errors in page coding	<b>duplicate-idm, duplicate-id-active, contain “aria-”</b>

Let's imagine that the obtained results indicate that there are “**image alt**” and “**link-name**” errors. Therefore, criteria #2 (availability of alternative text for all images) and #3 (availability of accessible text for all hyperlinks) are not successful, whereas the others are successful.

## Criteria for manual testing

Criteria 8 to 10 are checked manually, so it will take a little more time.

Criterion 8: providing a mechanism to quickly navigate to the main content of the page. This is the link that gets keyboard focus the first time you press the **Tab** key on a page. For example, on the website of the Cabinet of Ministers, this link is called “**Go to the main page**”. Press the **Enter** key on the link and go to the block with other links – “**Go to the news block**”, “**Go to the services block**”. If the link works, then the constituent has been reached.

Criterion 9: keyboard focus marking. The user should be able to see where the keyboard focus is. If this happens, then the component has been reached. To check, open the page and move with the **Tab** key. If you see a focus (a frame around or an underline under each interactive element), then the component has been reached. If the focus is not visible somewhere – the component has not been achieved.

Criterion 10: support for keyboard control. Open the page in the browser and launch the **Accessibility Insights for Web** extension. Click on “**Ad hoc tools**” and check the “**Tab stops**” box. This tool helps you identify the elements receiving focus by marking them with a circle.

Place the mouse cursor in the address bar or in the **URL** field of the browser, put the mouse down and check that the following two conditions are met:

1. You can go to each interactive element (links, buttons, fields, etc.) using the **Tab/Shift** + **Tab** keys. That is, when you press the Tab key, the focus moves from one interactive element to another without skipping a single interactive element.
2. Interactive elements can be interacted with using the keyboard. For example, you can use the Enter key to go to links or expand a submenu.

If both conditions can be confirmed, then the component has been achieved. If not, the component has not been achieved.

The indicator of basic accessibility is equal to the sum of the points of all its components. The minimum value of the indicator is 0 points, the maximum is 10 points. Based on the general indicator, we determine the level of basic accessibility of the website:

0-3 points	Low level
4-6 points	Medium level
7-8 points	Sufficient level
9-10 points	High level

Therefore, anyone can conduct monitoring, and it does not require special knowledge. However, please note once again that the simplified procedure cannot detect all accessibility problems.

## Supplementary materials to Chapter 9:

1. Availability check while working in Office programs.



URL: <https://support.microsoft.com/en-gb/office/check-accessibility-while-you-work-in-office-apps-ae9e8ea7-1f22-41af-ad04-cc2919daebae>

2. Creating and testing accessibility in PDF.



URL: <https://helpx.adobe.com/ua/acrobat/using/create-verify-pdf-accessibility.html>

3. Accessibility Insights for Web tool.



URL: <https://accessibilityinsights.io/docs/web/overview/>

4. Web accessibility of websites of government authorities. Report on monitoring results. UNDP in Ukraine, 2023.



URL: <https://www.undp.org/uk/ukraine/publications/accessibility-state-authorities-websites-report-study-results>

5. Infographic “Results of web accessibility assessment of 100 government authorities’ sites of Ukraine”. UNDP in Ukraine, 2023.



URL: <https://www.undp.org/uk/ukraine/publications/infographic-accessibility-100-government-authorities-websites-ukraine-results-monitoring-2022>

## Concluding remarks

Evidently, the topics in this manual do not cover all aspects of digital accessibility, but rather provide a basic understanding of this important and large segment of inclusivity. For further study of digital accessibility, we recommend referring to additional sources provided after each chapter.

One can find the most relevant thematic materials on the website of UNDP in Ukraine and on the organization's Facebook page. In addition, the UNDP YouTube channel periodically posts recordings of trainings on web accessibility, which can be viewed freely.

We wish you good luck and further interest in the development of digital accessibility in Ukraine.

