

MANAGEMENT
ON THE CREATION OF SITES
AND CONTENT FOR PEOPLE
WITH DISABILITIES

SO-EDINENIE FOUNDATION

ART. LEBEDEV

Designed by Art. Lebedev Studio

Contents

Problems in everyday life

Problems in everyday life 4

The perception of information 6

Regulations 8

Techniques and examples of implementation

What do you need? 11

Text 12

Page 14

Layout 17

Input fields 25

Tables 33

Images 35

Thanks 40

About the foundation

Charitable Foundation for
the Support of the Deafblind
«So-edinenie» 42

Achievement 43

Problems in everyday life

4

Problems in everyday life

Sites that come across to ordinary people as clear and structured resources are often a challenge for people with disabilities.

All websites from Russian government agencies should be adapted for people with disabilities. However, this requirement is not always met.

Customization tools are available on some web pages, but more often than not, this is not enough.

According to the UN, over 285 million people in the world have vision problems, of which approximately 40 million are completely blind. Every day they face difficulties that an ordinary person does not think about: it is difficult for them not only to take the subway or travel to another city, but also to check the news feed on social networks.

According to the WHO, there are over 1 billion people with disabilities in the world today.

This is approximately 15% of the total population of the planet. And this number is growing every year.

www.who.int/news-room/fact-sheets/detail/disability-and-health

218 000

people in Russia will be unable
to use your website without
it being adapted

103 000

people will never see
your site

The two main tools which visually-impaired readers use to access information are the screen reader and the Braille line.

Both provide the ability to read information from the screen. However, this is not always possible.

For example, when sighted users see a blooming garden in one photo, and a factory floor in another, the blind only hear the word 'image', delivered in a mechanical voice.

This happens if the programmer has not written additional text information in the site code about the image that the reader can read.

People with deafness, hand tremors and other diseases face problems that, together with impaired or absent vision, practically deprive them of access to information.



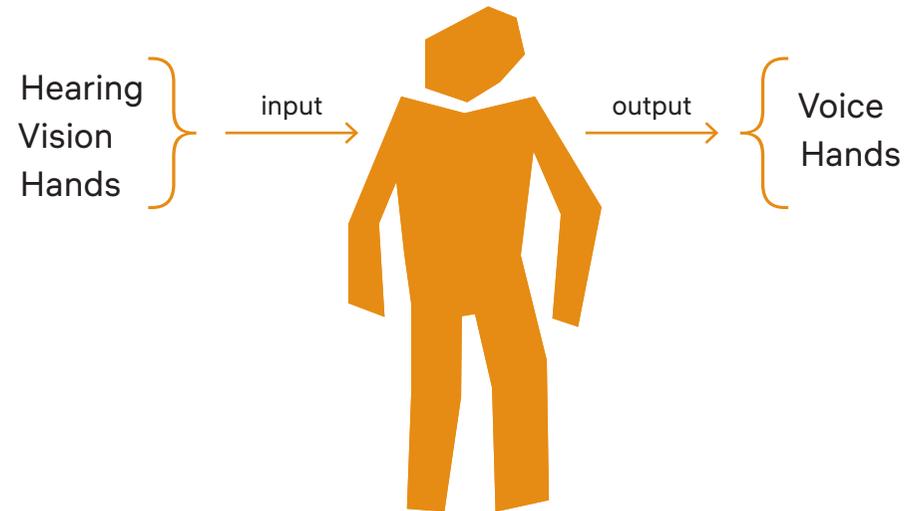
Tactile braille display

6

The perception of information

We can all be thought of as mechanisms that perceive (*input*) and reproduce (*output*) information.

The senses and their tools of transmission are hearing, sight, touch and voice.



At the same time, there are different degrees of disturbance in the senses.

Hearing—loss rate, hearing loss.

Vision—the level of loss, decreased visual acuity.

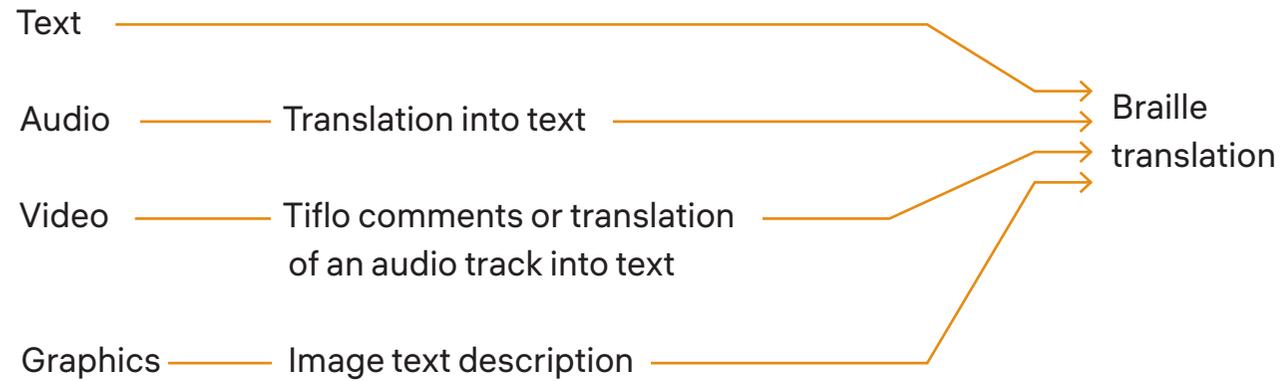
Voice—volume, quality.

Hands—sensitivity, trembling, presence.

The information we percieve can be divided into 4 main types.

Text	Audio	Graphics	Video
Sight, hands	Hearing	Vision	Hearing, sight

A person who has completely lost their hearing and vision will only be able to perceive text information which is transformed into Braille.



Regulations

Web accessibility in Russia is regulated by three legislative acts and one standard:

- Federal Law No. 419-FZ;
- Federal Law No. 8-FZ;
- Order of the Ministry of Communications and Mass Media of Russia No. 483;
- The national standard of the Russian Federation GOST R 52872–2012 “Internet resources. Accessibility requirements for visually impaired people.”

Federal Law No. 419-FZ

This law prohibits discrimination against people on the basis of disability. It is based on the UN Convention on the Protection of the Rights of Persons with Disabilities and on Federal Law No. 181-FZ ‘On Social Protection of Persons with Disabilities in Russian federation.’ The law defines discrimination as:

“Any difference, exclusion or limitation due to disability, the purpose or result of which is to diminish or deny the recognition, exercise or activities, on an equal basis with others, of all human and civil rights and freedoms as guaranteed in the Russian Federation in political, economic, social, cultural, civil or any another areas.”

The main requirements that regulatory documents impose on web resources are listed below.

Federal laws and GOST

All government websites should have a separate version for the blind and visually-impaired. The requirements are based on the provisions of WCAG 2.1.

**Order of the Ministry of Digital Development,
Communications and Mass Media**

Images should be described in text, unless they are decorative. On specialized sites for visually-impaired people, information should be posted on how to use public services.

Techniques and examples of implementation

11

What do you need?

When adapting sites for people with disabilities and those with lost senses, it is important to address reading functionality first. To do this, one of the following techniques must be used (in decreasing order of importance):

- layout (structural code);
- scale-up;
- recognition by screen readers;
- color correction;
- text description;
- audio description;
- subtitles and audio commentary.

Text

1. Decorative text

Embellish decorative text with font symbols, rather than display as a picture. Decipher abbreviations and explain complex terms. This will help users perceive the text as a whole.

Example

Eat some more of these soft buns

```
<span class="decor">Eat some more of these soft buns.</span>
.decor {
  font-family: 'Bad Script', cursive;
  font-size: 2rem;
}
```

2. Abbreviations

Decipher abbreviations. Add abbreviations to the `<abbr>` tag

Example

“FAP works from 8 to 20 hours”

```
<abbr title="Feldsher-obstetric station">FAP</abbr>works from 8 to 20 hours
```

3. Definitions

Use the `<dfn>` tag to highlight complex words in the text, use the `rel` attribute to style links, and use the `<dl>` tag to style the list of definitions.

Example

Tiflo commenting

a concise description of an object, space or action that is incomprehensible to a blind person without special verbal explanations

```
<dl><dt>Typhlo commentary</dt><dd>laconic description of an object, space or action that is incomprehensible to a blind person without special verbal explanations</dd></dl>
```

4. Numbers

In writing, integers in digital form can be broken up into groups (digits). Technical rules “tell” us to split numbers, starting with 5-digit ones. In Russia, this traditionally means that groups can only be separated from each other with spaces. However, in other systems it is permissible to separate the digits with periods or commas. This means it is easier for the screen reader to read the number written in one piece. So, visually for all users, *twenty-five million* will look like 25 000 000. But for the screen reader, you need to hide the tag with this number using the `aria-hidden="true"` attribute and write the number itself together: `aria-label="25000000"`. Only in this case can the screen reader interpret it as ‘twenty-five million’, and not as “Twenty-five, zero, zero, zero, zero, zero, zero.”

Page

5. Page title

Add the `<title>` meta tag to the code of each page. The attribute of this tag should describe the title and purpose of the page. When navigating the site without reloading the page, the content of the `<title>` tag should also be changed.

Example



[Foundation](#) ▾ [Activities](#) ▾ [Partners](#)

```
<title>Connection. Deafblind Support Foundation</title>
```

6. Content language

Specify the main language of the page in the `lang` attribute of the `<html>` tag so that the speech synthesizer can immediately determine in which language the site is developed. Examples of `lang` attribute values `lang: ru`—Russian language, `en`—English language, `de`—German language, etc. If the page contains text in another language (other than the main one), then it must be enclosed in the `` tag by adding a language attribute with a value that matches the inserted language.

Example

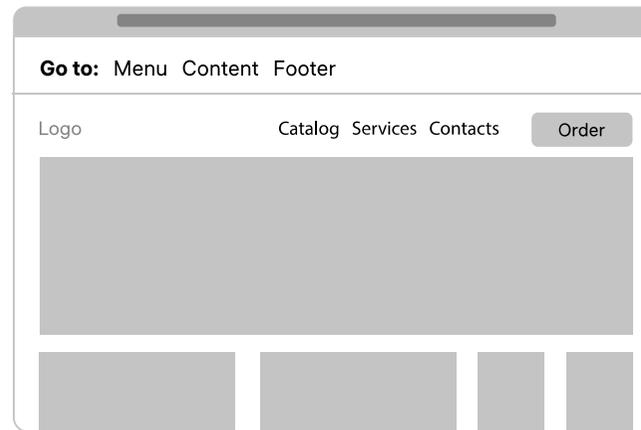
To find out how to get to the city center,
say: How can I get to the city center?

To find out how to get to the city center, say:
`` How can I get to the city center? ``

7. Repeating blocks

Add a link at the beginning of the site to jump to the main content so that the user can skip repeated blocks (header, navigation), if they wish.

Other options are to add links to all blocks of the site at the top of the page, or place a link to the end of each block at the beginning of each of them.



8. Search content

Give the user the ability to search for content in several ways: add a list of related pages, a table of contents, a sitemap, a search tool, a list of all pages on a site, etc.

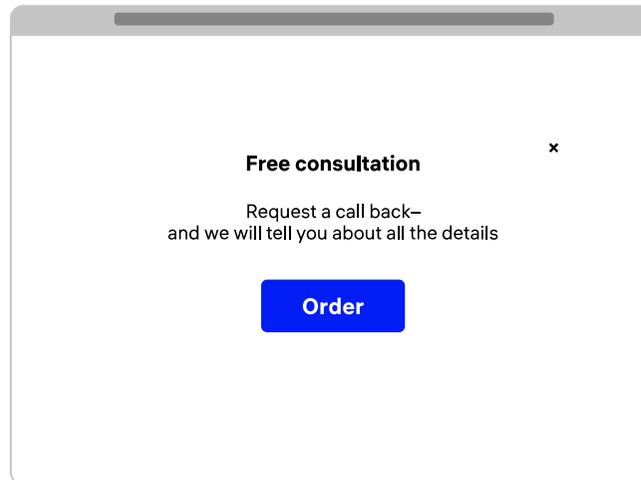
9. Changing the page

Shifting focus and changing the settings of a component should not result in changes to the page, unless the user has been notified of this in advance, before starting to work with the component.

A common case is navigating to another page or opening a new window on the `onchange` event in `<select>` elements, radio buttons, or checkboxes. Avoid this behavior, and go to another page only by pressing the on-screen button or pressing the Enter key.

Changes to the page should only occur at the request of the user, or they should be able to undo the change.

For example, a user might have a modal window open.



Layout

10. Semantics

The layout must be semantically correct: use HTML elements for their meaning. Please be aware that not all browsers render the content of special tags in the same way.

For headings, use the tags `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`. Maintain a clear heading hierarchy.

Use the `` tag for bulleted lists and `` for numbered lists.

Use the following HTML5 tags:

- `<header>` for the site header; inside the tag, insert the introductory site elements that are on each page—logo, navigation, page title, search bar;
- `<nav>` for site navigation;
- `<section>` for dividing the page into semantic parts;
- `<aside>` for a sidebar listing articles, advertisements, archived entries;
- `<footer>` for the footer block of the page with information about the author of the article, copyright, etc.;
- `<figure>` for grouping elements such as images and text;
- `<main>` for the main content on the page.

11. Adaptability

Use a responsive layout and relative font size units to give the user the ability to zoom in on the page at least twice. When the page is scaled, a horizontal scroll element is displayed on the screen, which leads to the loss of more than half of the information. Responsive layouts solve this problem.

Relative units for setting font sizes:

- rem** font size is calculated relative to the value specified for the `<html>` tag;
- em и %** font sizes are calculated relative to the value written for the closest parent of the current element.

Page scale 100%



Page scale 300%



In addition to scaling, the site must be able to change its color scheme for people with blurry or distorted vision. To do this, you can create a panel for the visually impaired. For example, adjusting the display of color on the site will help people with color blindness, and increasing the spacing between letters and lines, as well as adjusting the serif font for people with dyslexia when reading.

The font resizing tool duplicates scalability, but older generations are not always familiar with this kind of tool. Many people are looking for the same button to increase the size. As such, it should be left regardless of any alternative.

Example

The image shows a rounded rectangular panel with a light gray border. It contains five sections of settings, each with a label on the left and one or more buttons on the right. The 'Inversion' button is highlighted in black, while the others are light gray with black text. The 'Recommended' buttons are also highlighted with a slightly darker gray background.

Font and background color	Standard	Contrasting	Inversion
Images	Colored	Disable	
Line spacing	Recommended	Large	Small
Distance between letters	Recommended	Large	Very large
Use font	Arial	Times New Roman	

12. Validity

Observe the validity of the layout—compliance with the W3C specification. Check the nesting of tags, close all paired tags, check the spelling of attributes and quotes, do not repeat the same id several times on the page. Screen readers may misinterpret an invalid layout.

Check the validity of the layout using the online W3C validator:

validator.w3.org

13. Navigation

Make sure the navigation is the same on all pages. Lay out nested menus correctly. Give the visitor information about their current position on the page. This will help blind and visually-impaired users to move freely around the site, and make sure that they always know where they are.

14. Menu

The elements of the menu, which is always displayed in expanded form on the site, should have the `role` attribute with the following values:

- `menubar` horizontal menu bar;
- `menu` second level menu;
- `menuitem` item is a separate menu item;
- `separator` separator between two groups of menu items.

To focus on menu items using the keyboard, use the `tabindex="-1"` attribute. Add the value `tabindex="0"` to the first element of the main menu (“Hats, scarves, mittens” in the example above). In this case, additional menu items are set as unavailable.

A particularly important point is the management of user behavior when working with the keyboard. The introduction of a horizontal menu bar on the site forces the user to change the mode and switch from the classic screen reader review to editing. Such a transition is not at all obvious to both the site visitor and the developers.

Further interaction with menu items is carried out without using the hotkeys of the screen reader, vertical arrows $\uparrow \downarrow$ and the *Tab* keys, and with the horizontal arrows $\leftarrow \rightarrow$ to move through the parent items. In this case, vertical arrows $\uparrow \downarrow$ are used only for working with menu sub-items. But this kind of way of interacting with a resource makes sense if there is a global menu consisting only of submenus, which, in turn, contain several sub-items.

The WAI-ARIA standard is not a panacea or silver bullet which miraculously facilitates all the necessary work on a website for blind and visually-impaired people. This is a tool designed specifically for special-critical-cases. However, these kinds of cases are extremely rare. And where it is possible not to use the ARIA standard, it is better to do without it.

15. Uniformity

Do not change the sequence of navigation elements that are repeated on all pages of the site. Name the elements of the page with similar functionality in the same way.

16. Current page

Add duplicate navigation (breadcrumbs) or a sitemap to the page to show where the user is in relation to the overall hierarchy and other pages.

17. Keyboard controls

Give the user the ability to control the site from the keyboard with no time limit on keystrokes. This will help people who cannot use a mouse, work with the page without obstacles—scroll, follow links, fill out forms.

The recommendation does not apply to cases where control over the keyboard is impossible, for example, when drawing. Keyboard controls should not interfere with mouse controls.

Ensure the correct sequence of focus transition—check the semantics of the layout, the correct sequence of content on the page, and add a visual display of focus on active elements. Go to the next element—one press on the Tab key, to the previous one—*Shift+Tab*. If other keys are programmatically designated, indicate this at the beginning of the site.

WGAC 2.1 recommends using the `tabindex` attribute—it should be filled with positive numbers in ascending order. It is not recommended to use `tabindex` for all elements. If new elements appear on the page, it will be easy to get confused in the set values.

Rely on the natural order of elements on the page. Use `tabindex` on elements only in special cases where non-standard consistency is important. Use `tabindex="-1"` to skip the element and `tabindex="0"` to focus. Highlight the active element visually using the: `focus` pseudo-class for links, buttons and input fields.

Example



```
<label id="label-mail">Email</label>
<input type="email" aria-labelledby="label-mail" aria-required="true"
tabindex="0">

<label id="label-phone">Password</label>
<input type="email" aria-labelledby="label-phone" aria-required="true">
```

Avoid using `mousedown` or `mouseup` as they are not accessible for keyboards. If this is not possible, add keyboard event handling.

Input fields

18. Input fields

Add titles to all input fields on the form, including radio buttons, checkboxes, and drop-down lists. The title should be short and reflect the purpose of the field. Add a name using the `<label>` tag, in the `for` attribute write the field `id`.

Example

Badly

```
<label>Email</label>
<input type="text">
```

Good

```
<label for="input-mail">Email</label>
<input type="text" id="input-mail">
```

If you want to hide the label, place it in front of the field, use the `for` attribute binding and a special class for assistive technologies.

```
.visually-hidden {
  position: absolute;
  width: 1px; height: 1px; margin: -1px; border: 0; padding: 0;
  clip: rect(0 0 0 0);
  overflow: hidden;
}
```

This class hides the label visually, but screen readers will read it. Please note that screen readers and other assistive technologies, like browsers, hide elements using `display: none` and `visibility: hidden`.

Another way to associate a field and a label is with the `aria-labelledby` by attribute. It is used as a `for` attribute for the `<label>` tag, the value of this attribute is the same as the value of the `id` of the field.

Example

```
<label for="input-mail" id="label-mail">Email</label>  
<input type="text" id="input-mail" aria-labelledby="label-mail">
```

The field name can be specified in the `aria-label` or `title` attributes. The second method is less reliable, and the name will be visible to all users as a white window when they hover over the field.

Add instructions for filling in the fields to the form, check the required fields to avoid mistakes. Required fields can be specified in the name of the `label` field by adding the `required` or `aria-required=>true<>` attributes to the field.

```
<label for="input-mail" id="label-mail">Email</label>  
<input type="email" id="input-mail" aria-labelledby="label-mail"  
  aria-required="true" required>
```

Often, developers use a method to avoid using the `for` and `id` attributes. To do this, you need to “wrap” the `<input>` tag with the `<label>` tag.

```
<label>Name:<input type="text" name="firstname"></label><br>  
<label><input type="checkbox" name="subscribe">  
Subscribe to newsletter </label>
```

19. Field groups

Grouping related fields helps to divide the form into parts for easier readability.

Group fields using the `<fieldset>` and `<legend>` tags.

The `<fieldset>` tag is a container for related form elements, `<legend>` is a group heading, where you can specify not only the purpose of the group, but also the general characteristics of the fields. For example, indicate that all fields in a group are required.

Example

Contacts

Email

Telephone

```
<fieldset>
  <legend>Contacts</legend>
  <label for="input-mail" id="label-mail">Email</label>
  <input type="email" id="input-mail" aria-labelledby="label-mail">

  <label for="input-phone" id="label-phone">Phone</label>
  <input type="email" id="input-phone" aria-labelledby="label-phone">
</fieldset>
```

Another way to group form fields is to use the `role="group"` and `aria-labelledby` by attributes, for which you need to specify the `id` of the element that contains the description of the field group.

Example

Contacts

Email

Telephone

```
<div role="group" aria-labelledby="user-data">
  <div id="user-data">Contacts</div>
  <label for="input-mail" id="label-mail">Email</label>
  <input type="email" id="input-mail" aria-labelledby="label-mail">

  <label for="input-phone" id="label-phone">Phone</label>
  <input type="email" id="input-phone" aria-labelledby="label-phone">
</div>
```

Provide the user with clear instructions for entering information. Specify required fields, valid data formats, and time limits. Place the statement before the `<form>` tag so that the screen reader reads the statement before entering form reading mode.

For simple instructions, use hints right in the field description. For example, for a date field, you can write its name.

Example

Email
(required)

```
<label for="input-mail"
  id="label-mail">Email (required)</label>
<input type="email"
  id="input-mail" aria-labelledby="label-mail" aria-required="true">
```

20. Notifications

Provide the user with information about the results of submitting the form. If the page refreshes after submitting the form, change the tag for it `<title>` for the error, or add an `h1` heading with the error message.

If the page does not refresh, add an error list with the `role="alert"` attribute above the form element. Associate each error with a field using the `aria-describedby` attribute. The `aria-describedby` attribute can be updated dynamically—normally it points to a hint or field description, and in the case of a typing error, to the error text (instead of or together with a hint, `aria-describedby` can contain multiple identifiers).

Example

Invalid email format

Email	<input type="text" value="example.a"/>
Telephone	<input type="text" value="+7 921 123-45-67"/>

```
<p class="txt-danger" id="danger-mail">Invalid email format</p>
<p>
  <label for="input-mail">Email</label>
  <input type="email" id="input-mail" aria-describedby="danger-mail"
    class="textbox" aria-required="true">
</p>
<p>
  <label for="input-phone">Phone</label>
  <input type="email" id="input-phone" class="textbox" aria-required="true">
</p>
```

21. Notifications to fields

You can display field notifications after submitting the form or directly during data entry.

After submitting the form, set the focus to the first field so that the user is aware of the results of each field's validation. If an input error occurs, display a message about it in text after the field. Associate the error with the field using the `aria-describedby` attribute.

To display messages when the field is validated during input, add an area after the field with the `aria-live="polite"` attribute. If the field is validated after losing focus, add the attribute after the field `aria-live="assertive"`. Programmatically add the error to this element. The difference between these approaches is that `aria-live="assertive"` interrupts all of the current screen reader tasks. If you use it to validate a field as you type, after each change the screen reader will go to this item and then read it several times.

Example

Email

Invalid email format

Telephone

```
[...]  
<label for="input-mail" id="label-mail">Email</label>  
<input type="email" id="input-mail"  
      aria-describedby="label-mail error-message-mail"  
      class="textbox" aria-required="true">  
<small aria-live="polite" id="label-mail, error-message-mail">  
  Invalid email format</small>
```

[...]

Tables

22. Tables

Specify relationships between cells, use semantically correct tags to identify cells. Use the `<th>` tag for heading cells. Use the `scope` attribute to bind a cell to data. This will help blind and visually-impaired users navigate the table—recognize the table headings, and navigate the rows and columns correctly.

For simple tables with a single header, use the `row` or `col` values.

Example

FULL NAME	Age	Profession
Ivanov Alexander Alexandrovich	23	Doctor
Kitaeva Anna Sergeevna	34	Teacher
Kolosova Alexandra Ivanovna	32	Programmer

```
<thead>
  <tr>
    <th scope="col">Full name</th>
    <th scope="col">Age</th>
    <th scope="col">Profession</th>
  </tr>
</thead>
```

For tables with two headers, use a [row](#) at the same time for top headers and [col](#) for horizontal lines. Add a tag `<caption>` or the [aria-describedby](#) attribute for the table caption

Example

Project members

Number	FULL NAME	Age	Profession
1	Ivanov Alexander Alexandrovich	23	Doctor
2	Kitaeva Anna Sergeevna	34	Teacher
3	Kolossova Alexandra Ivanovna	32	Programmer

```
<caption>Project members</caption>
<thead>
  <tr>
    <th scope="col">№ line</th>
    <th scope="col">Full name</th>
    <th scope="col">Age</th>
    <th scope="col">Profession</th>
  </tr>
</thead>
<tbody>
  <tr>
    <th scope="row">1</th>
    <th>Ivanov Alexander Alexandrovich</th>
    <th>23 years old</th>
    <th>Doctor</th>
  </tr>
</tbody>
```

Images

23. Informative images

Informative images illustrate textual information in a visual form, complementing or denoting it. Add an `alt` attribute for these kinds of images.

Example



Screw in the bulb clockwise.

Bably

```
 Screw  
the buld clockwise.
```

Good

```
 Screw in the light  
buld clockwise.
```

24. Decorative image

Decorative images should be added using the CSS `background-image` property so that screen readers will ignore them. If the decorative element is an image, such as within a link, add an empty `alt=""` or the `role="presentation"` and `aria-hidden="true"` attributes to it.

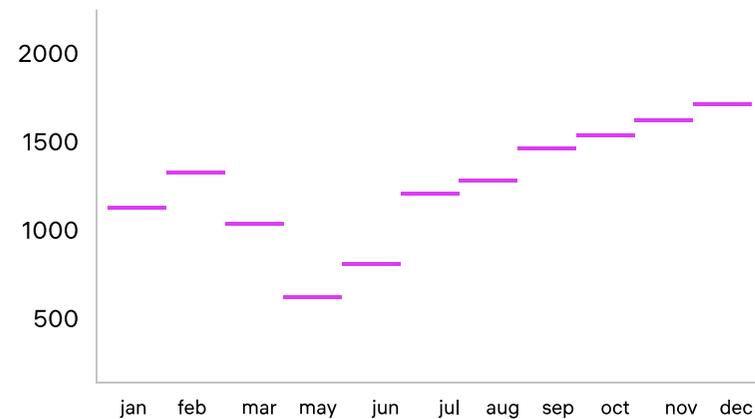
25. Graphs, diagrams

For complex images—maps, charts, graphs—add a short description to the `alt` attribute, and put the details on a separate page by placing them in the `<figcaption>` tag, in the `longdesc` or `aria-describedby` attributes.

Add a link to another page or to a fragment of the current page with a full description. The disadvantage of this approach is that the link will not be associated with the image by meaning.

Example

Visitors to the site



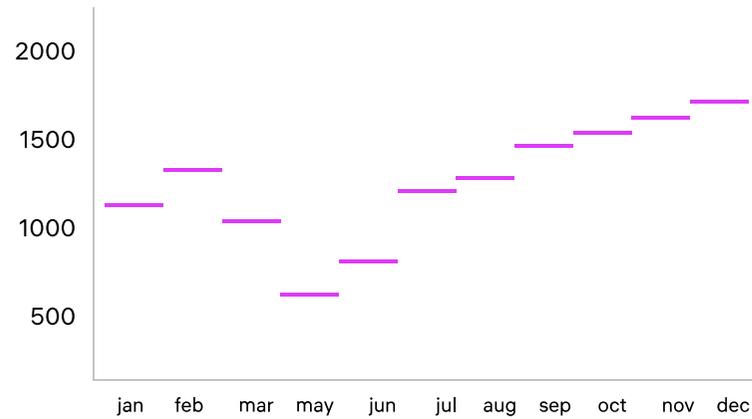
```

<a href="chart-info.html">Image Description</a>
```

Add the image and its description to `<figure>` with `role="group"` attribute. Description “wrap” in `<figcaption>` tag.

Example

Visitors to the site

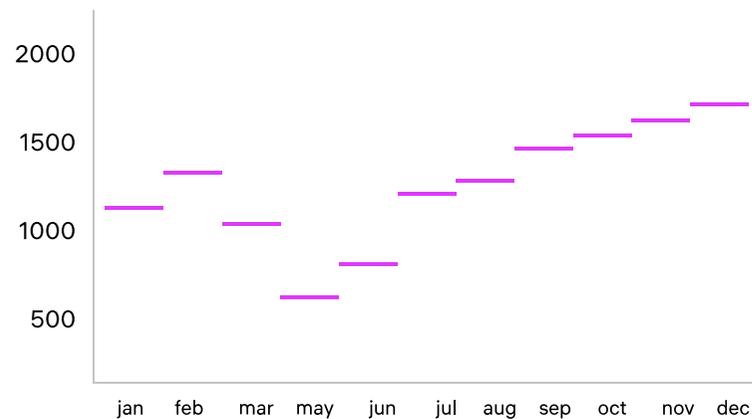


```
<figure role="group">  
    
  <figcaption>  
    <a href="chart-bar.html">  
      So-edinenie.org website visit schedule for 2021</a>  
  </figcaption>  
</figure>
```

Add the **longdesc** or **aria-describedby** attribute to the image. Write the **id** of the element within it, containing a detailed description of the image. The differences between the examples are that the element with **id** from **aria-describedby** must contain only text; additional tags can be added to the description of the element with **id** from **longdesc**.

Example

Visitors to the site



```

<p id="chart-longdesc">
  So-edinenie.org site visits schedule for 2021: site visited in October 26
  thousand people, this is the highest number of visits in 2021.
</p>
```

26. Image groups

For groups of images that represent the same information, such as rating stars, add an `alt` description to the first image only.

Example

Movie rating 4 out of 5



```
<img src=>images/i-star.png> alt=>Movie rating 4 out of 5<>  
  
  
  
  

```

Thanks

The techniques and examples for implementation were compiled on the basis of the development of the [Weblind](#) project team with the support of the [Gorbunov Bureau School](#).

Author: Anastasia Sukhoverkhova.

Experts: Alexey Lyubimov, Vadim Makeev and Alexander Nefedov.



About the foundation

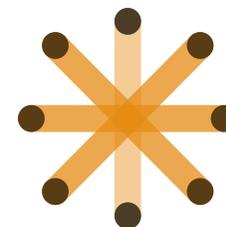
Charitable Foundation for the Support of the Deafblind «So-edinenie»

The launch of the foundation in 2014 was a turning point in the lives of deaf-blind people and allowed them to find a way out of social isolation.

Computer literacy and social rehabilitation courses, assisted living centers and advanced training for tiflosurdo translators, development of technical means of rehabilitation, gadgets and mobile apps that make life easier for people with hearing and vision impairments, the development of an inclusive theater in Russia—all these have become elements of the large-scale puzzle called a support system for people with simultaneous hearing and vision impairments.

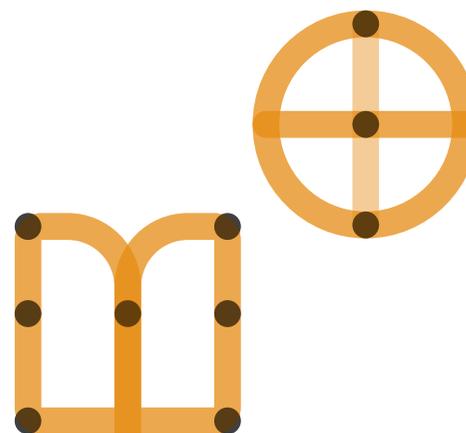
During the pandemic, the most important task for the fund has been to create an accessible digital educational environment.

A key project in 2021 will be the development of a cortical implant. This year we plan to present a prototype of the first vision restoration technology using a brain implant.



To date, we have identified more than 4,500 deaf-blind people (according to experts, there are about 15,000 of them in Russia).

The result of our daily work is of importance to both for visually-impaired and blind people (of which there are about 218,000), and for the hearing impaired and deaf (of which there are 13 million).



Achievement

- Established a methodology for the accompanied living of deaf-blind adults with two sites in Moscow and the Moscow region, a third was opened in 2020, “The house of accompanied residence” in the Northwestern Federal District, which will also serve as a center for rehabilitation and training.
- Opened 8 *drama schools* (in Moscow, St. Petersburg, Yekaterinburg, Kazan, Novosibirsk, Kaliningrad, Orsk); On 25 February, the ninth school opened in Kurgan in 2021.
- A qualitative technological breakthrough for Russia: in 2017, *the first two operations were carried out to implant a bionic eye in a blind person*, making it possible to partially restore vision to two wards of the foundation.
- Opened 41 leisure centers in 27 regions of Russia.
- Founded 15 resource centers for the deaf-blind throughout Russia.
- Today, 38 regional branches of the “Community of Families of the Deafblind” support people.
- There are 3 training centers for deaf-blind adults.
- Established a partnership with Sense International, The Perkins Institute, Kentalis Institute, and DBI as part of an effort to cooperate with organisations based abroad.
- Braille displays included in the list on government lists.



Leisure, training and resource centers
in regional Russia



Designed by Art. Lebedev Studio