

РУКОВОДСТВО
ПО СОЗДАНИЮ САЙТОВ И КОНТЕНТА
ДЛЯ ЛЮДЕЙ С ОГРАНИЧЕННЫМИ
ВОЗМОЖНОСТЯМИ

ФОНДА «СО-ЕДИНЕНИЕ»

Содержание

Проблемы в повседневной жизни

Ежедневные проблемы 4

Восприятие информации 6

Регуляторика 8

Приемы и примеры реализации

Что нужно 11

Текст 12

Страница 14

Верстка 17

Поля ввода 25

Таблицы 33

Изображения 35

Благодарность 40

О фонде

Благотворительный фонд
поддержки слепоглухих
«Со-единение» 42

Достижения 43

Проблемы в повседневной жизни

Ежедневные проблемы

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

Сайты, которые воспринимаются обычными людьми как четкие и структурированные ресурсы, для людей с ограничениями нередко являются излишне сложными.

Все сайты российских госучреждений должны быть адаптированы для людей с ограничениями. Однако сейчас это требование соблюдается далеко не всегда. На некоторых веб-страницах доступны инструменты для адаптации, но чаще всего этого бывает недостаточно.

По данным ООН, в мире больше 285 млн людей имеют проблемы со зрением, из них около 40 млн — полностью слепы. Каждый день они сталкиваются с трудностями, о которых обычный человек не задумывается: им сложно не только спуститься в метро или поехать в другой город, но и проверить ленту новостей в соцсетях.

По данным ВОЗ, в мире живет более 1 млрд людей с инвалидностью. Это примерно 15% от всего населения планеты. И их число с каждым годом увеличивается.

218 000

человек в России не смогут
пользоваться вашим сайтом
без адаптации

103 000

человек не увидят
ваш сайт никогда

Двумя главными инструментами для считывания информации слабовидящими являются экранный ридер и строка Брайля.

Оба устройства дают возможность считывать информацию с экрана.

Однако сделать это можно не всегда.

Например, когда зрячие пользователи видят на одной фотографии цветущий сад, а на другой — заводской цех, слепые в обоих случаях могут услышать от механического голоса ридера лишь слово «изображение». Такое происходит, если программист не прописал в коде сайта дополнительную текстовую информацию об изображении, которое ридер может считать.

Люди с глухотой, тремором рук и другими заболеваниями сталкиваются с проблемами, которые в совокупности с ослабленным или отсутствующим зрением практически лишают человека доступа к информации.



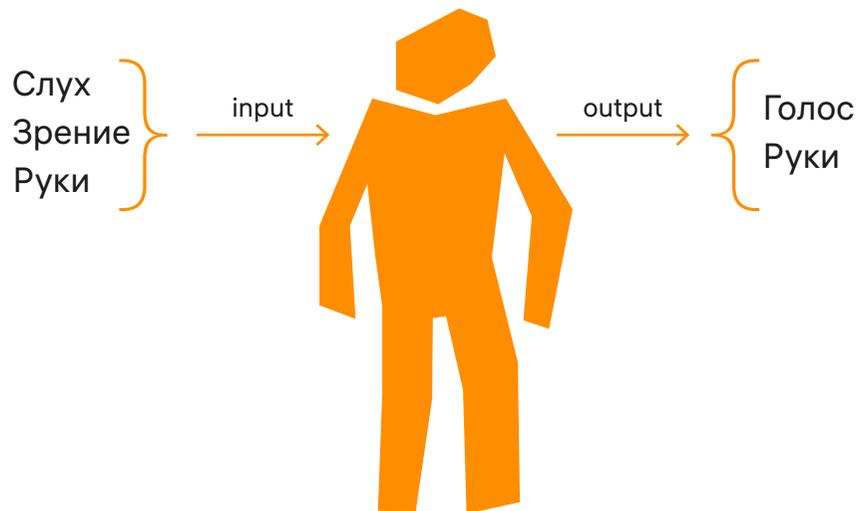
Тактильный дисплей Брайля

6

Восприятие информации

Любого человека можно представить как механизм, который воспринимает (*input*) и воспроизводит (*output*) информацию.

Органы чувств и их передачи — это слух, зрение, руки и голос.



При этом есть разные степени нарушения чувств.

Слух — уровень потери, тугоухость.

Зрение — уровень потери, снижение остроты зрения.

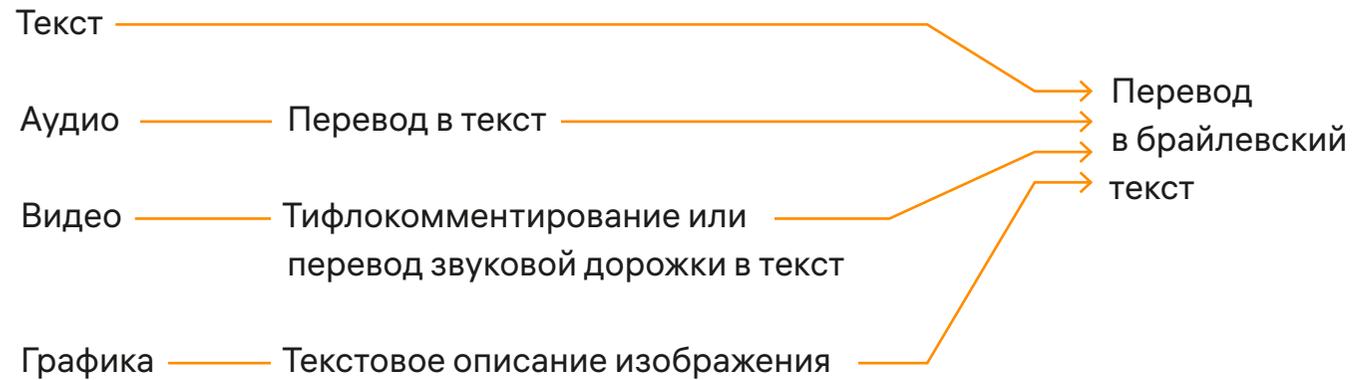
Голос — громкость, качество.

Руки — чувствительность, дрожание, наличие.

Воспринимаемую информацию можно разделить на 4 основных типа.

Текст	Аудио	Графика	Видео
Зрение, руки	Слух	Зрение	Слух, зрение

Человек, полностью потерявший слух и зрение, сможет воспринимать только текстовую информацию, которая трансформируется в шрифт Брайля.



Регуляторика

Веб-доступность в России регулируется тремя законодательными актами и одним стандартом:

- Федеральным законом № 419-ФЗ;
- Федеральным законом № 8-ФЗ;
- Приказом Министерства связи и массовых коммуникаций России № 483;
- Национальным стандартом РФ ГОСТ Р 52872–2012 «Интернет-ресурсы. Требования доступности для инвалидов по зрению».

Федеральный закон № 419-ФЗ

Закон запрещает дискриминацию людей по признаку инвалидности. Он основан на Конвенции ООН о защите прав людей с инвалидностью и на Федеральном законе № 181-ФЗ «О социальной защите инвалидов в Российской Федерации». В законе дано определение дискриминации:

«Любое различие, исключение или ограничение по причине инвалидности, целью либо результатом которых является умаление или отрицание признания, реализации или осуществления наравне с другими всех гарантированных в Российской Федерации прав и свобод человека и гражданина в политической, экономической, социальной, культурной, гражданской или любой иной области».

Основные требования, которые предъявляют к веб-ресурсам регуляторные документы, перечислены ниже.

Федеральные законы и ГОСТ

У всех государственных сайтов должна быть отдельная версия для слепых и слабовидящих. Требования к ней основаны на положениях WCAG 2.1.

Приказ Министерства цифрового развития, связи и массовых коммуникаций

Изображения должны быть описаны в виде текста, если они не носят декоративный характер. На специализированных сайтах для людей с инвалидностью по зрению должна размещаться информация о том, как им воспользоваться госуслугами.

Приемы и примеры реализации

Что нужно

Адаптируя сайты для людей с ограничениями и с потерянными органами чувств, важно в первую очередь создать функциональность считывания. Для этого необходимо использовать следующие приемы (в порядке убывания важности):

- верстка (структурный код);
- увеличение масштаба;
- распознавание экранными считывателями;
- цветовая коррекция;
- текстовая дескрипция;
- аудиодескрипция;
- субтитры и тифлокомментирование.

Текст

1. Декоративный текст

Оформляйте декоративный текст в виде шрифтовых символов, а не показывайте в виде картинки. Расшифровывайте аббревиатуры и объясняйте сложные термины. Это поможет пользователям воспринять текст как единое целое.

Пример

Съешьте еще этих мягких булочек

```
<span class="decor">Съешьте еще этих мягких булочек.</span>
.decor {
  font-family: 'Bad Script', cursive;
  font-size: 2rem;
}
```

2. Аббревиатуры

Расшифруйте сокращения. Добавьте аббревиатуры в тег `<abbr>`

Пример

«ФАП работает с 8 до 20 часов»

```
<abbr title=«Фельдшерско-акушерский пункт»>ФАП</abbr>работает с 8 до 20 часов
```

3. Определения

Для выделения в тексте сложных слов используйте тег `<dfn>`, для оформления ссылок применяйте атрибут `rel`, а с помощью тега `<dl>` оформляйте список определений.

Пример

Тифлокомментирование

лаконичное описание предмета, пространства или действия, которые непонятны незрячему без специальных словесных пояснений

```
<dl><dt>Тифлокомментирование</dt><dd>лаконичное описание предмета,  
пространства или действия, которые непонятны незрячему без специальных  
словесных пояснений</dd></dl>
```

4. Числительные

На письме целые числа в цифровой форме могут разбиваться на группы (разряды). Технические правила «велят» разбивать числительные, начиная с 5-значных. В русской традиции отделять группы друг от друга можно *только пробелами*, но в других системах допустимо разделение разрядов точками или запятыми. При этом скринридеру проще считать число, написанное слитно. Поэтому визуально для всех пользователей *двадцать пять миллионов* будут выглядеть как 25 000 000. Но для скринридера нужно скрыть тег с этим числом с помощью атрибута `aria-hidden="true"` и написать само число слитно: `aria-label="25000000"`. Только в этом случае скринридер интерпретирует его как «двадцать пять миллионов», а не как «двадцать пять, ноль, ноль, ноль, ноль, ноль».

Страница

5. Заголовок страницы

Добавьте в код каждой страницы метатеги `<title>`. Атрибут этого тега должен описывать название и назначение страницы. При навигации по сайту без перезагрузки страницы содержимое тега `<title>` также следует изменять.

Пример



О фонде ▾ Слепоглухим/Представителям ▾

```
<title>Соединение. Фонд поддержки слепоглухих</title>
```

6. Язык содержимого

Укажите основной язык страницы в атрибуте `lang` тега `<html>` — так синтезатор речи сможет сразу определить, на каком языке разработан сайт. Примеры значений атрибута `lang: ru` — русский язык, `en` — английский язык, `de` — немецкий язык и т. д. Если же на странице встречается текст на другом языке (отличном от основного), его нужно заключить в тег ``, добавив атрибут `lang` со значением, соответствующим языку вставки.

Пример

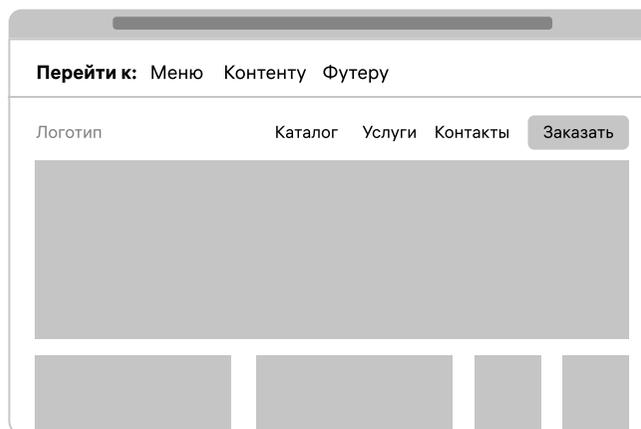
Чтобы узнать, как добраться в центр города, скажите:
How can I get to the city center?

```
<span lang="en">How can I get to the city center?</span>
```

7. Повторяющиеся блоки

Добавьте в начало сайта ссылку для перехода к основному содержимому, чтобы пользователь при желании мог пропустить повторяющиеся блоки (шапку, навигацию).

Другие варианты — добавить в начало страницы ссылки на все блоки сайта или разместить в начале каждого блока ссылку на его конец.



8. Поиск содержимого

Дайте пользователю возможность искать содержимое несколькими способами: добавьте список связанных страниц, оглавление, карту сайта, инструмент поиска, список всех страниц сайта и т. д.

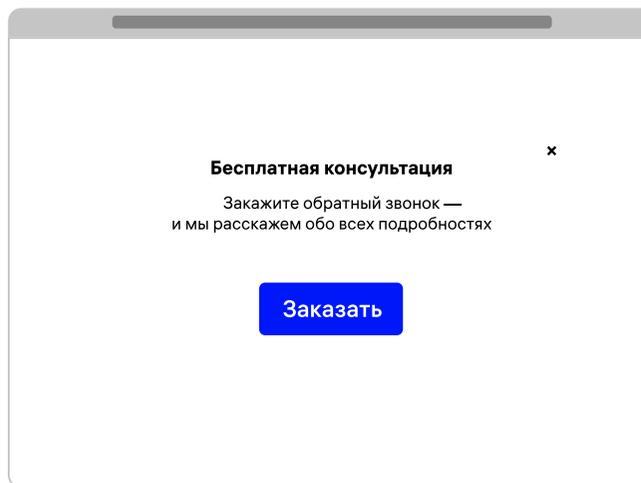
9. Изменение страницы

Перенос фокуса и изменение настроек компонента не должны вызывать изменений страницы, кроме случаев, когда пользователя уведомили об этом заранее, до начала работы с компонентом.

Распространенный случай — переход на другую страницу или открытие нового окна по событию `onchange` в элементах `<select>`, радиокнопках или чекбоксах. Избегайте такого поведения, переходите на другую страницу только по нажатию на экранную кнопку или на клавишу Enter.

Изменения страницы должны происходить только по запросу пользователя, либо он должен иметь возможность отменить изменение.

Например, у пользователя может открыться модальное окно.



Верстка

10. Семантика

Верстка должна быть семантически правильной: используйте элементы HTML по смысловому назначению. Помните, что не все браузеры отображают содержимое специальных тегов одинаково.

Для заголовков используйте теги `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`.

Соблюдайте четкую иерархию заголовков.

Для маркированных списков используйте тег ``, для нумерованных — ``.

Используйте следующие теги HTML5:

- `<header>` для шапки сайта; поместите внутрь тега вводные элементы сайта, которые есть на каждой странице — логотип, навигацию, заголовок страницы, строку поиска;
- `<nav>` для навигации по сайту;
- `<section>` для разделения страницы на смысловые части;
- `<aside>` для боковой колонки со списком статей, рекламы, архивных записей;
- `<footer>` для нижнего блока страницы с информацией об авторе статьи, копирайта и т. д.;
- `<figure>` для группировки элементов, например изображения и текста;
- `<main>` для основного содержимого страницы.

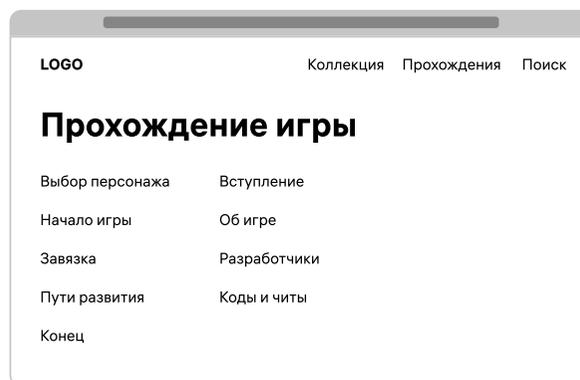
11. Адаптивность

Используйте адаптивную верстку и относительные единицы размера шрифта, чтобы дать пользователю возможность увеличивать масштаб страницы не менее чем в два раза. При масштабировании страницы на экране отображается элемент горизонтальной прокрутки, что приводит к потере более половины информации. Адаптивная верстка решит эту проблему.

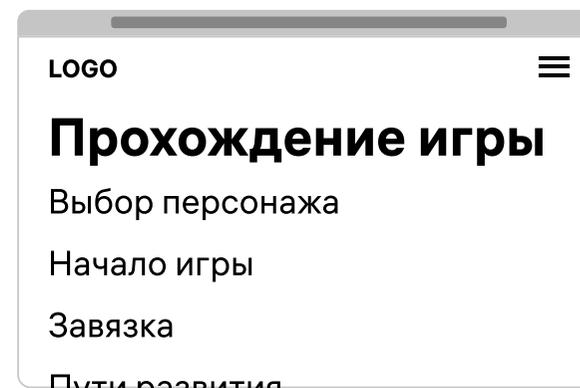
Относительные единицы для задания размеров шрифтов:

- rem** размер шрифта рассчитывается относительно значения, прописанного для тега `<html>`;
- em и %** размер шрифта рассчитывается относительно значения, прописанного для ближайшего родителя текущего элемента.

Масштаб страницы 100%



Масштаб страницы 300%



Помимо масштабирования, сайт должен «уметь» менять цветовую гамму для людей с нечетким или искаженным зрением. Для этого можно создать «панель для слабовидящих». Например, настройка отображения цвета на сайте поможет людям с дальтонизмом, а увеличение интервалов между буквами и строками, а также настройка шрифта с засечками — людям с дислексией, для которых рядом стоящие буквы при чтении меняются местами.

Инструмент по изменению размера шрифта дублирует масштабируемость, но старшее поколение не всегда бывает знакомо с таким инструментом. Многие ищут ту самую кнопку для увеличения кегля. Поэтому ее стоит оставить, несмотря на альтернативу.

Пример



The image shows a user interface for accessibility settings, presented as a rounded rectangle. It contains five rows of settings, each with a label on the left and one or more buttons on the right. The 'Инверсия' button is highlighted in black, while the others are light gray.

Цвет шрифта и фона	Стандартный	Контрастный	Инверсия
Изображения	Цветные	Отключить	
Расстояние между строками	Рекомендуемое	Большое	Маленькое
Расстояние между буквами	Рекомендуемое	Большое	Очень большое
Использовать шрифт	Arial	Times New Roman	

12. Валидность

Соблюдайте валидность верстки — соответствие спецификации W3C. Проверяйте вложенность тегов, закрывайте все парные теги, проверяйте написание атрибутов и кавычек, не повторяйте один и тот же `id` несколько раз на странице. Скринридеры могут неправильно воспринимать невалидную верстку.

Проверьте валидность верстки с помощью онлайн-валидатора W3C: validator.w3.org

13. Навигация

Сделайте навигацию на всех страницах одинаковой. Правильно верстайте вложенные меню. Дайте посетителю информацию о его текущем положении на странице. Это поможет слепым и слабовидящим пользователям свободно перемещаться по сайту, всегда зная свое положение.

14. Меню

У элементов меню, которое всегда отображается на сайте в развернутом виде, следует указать атрибут `role` со следующими значениями:

`menubar` горизонтальная панель меню;

`menu` меню второго уровня;

`menuitem` отдельный пункт меню;

`separator` разделитель между двумя группами элементов меню.

Чтобы настроить фокус на пункты меню с помощью клавиатуры, используйте атрибут `tabindex="-1"`. Первому элементу главного меню («Шапки, шарфы, варежки» в примере выше) добавьте значение `tabindex="0"`. При этом дополнительные пункты меню делаются недоступными.

Особо важным моментом является управление поведением пользователя при его работе с клавиатурой. Внедрение на сайт горизонтальной панели меню вынуждает пользователя сменить режим и перейти от классического обзора скринридера к редактированию. Подобный переход совсем не очевиден как для посетителя сайта, так и для разработчиков.

Дальнейшее взаимодействие с пунктами меню осуществляется уже без использования горячих клавиш скринридера, вертикальных стрелок `↑` `↓` и клавиши `Tab`, а при помощи горизонтальных стрелок `←` `→` для перемещения по родительским пунктам. Вертикальные стрелки `↑` `↓` при этом используются только для работы с подпунктами меню. Но такая схема взаимодействия с ресурсом имеет смысл при наличии глобального меню, состоящего только из подменю, содержащих, в свою очередь, по несколько подпунктов.

Стандарт WAI-ARIA — это вовсе не панацея и не волшебная палочка, чудесным образом облегчающая работу с сайтом незрячих и слабовидящих людей. Это инструмент, созданный специально для особых — критичных — случаев. Однако такие случаи возникают крайне редко. И там, где есть возможность не использовать стандарт ARIA, лучше обойтись без него.

15. Единообразие

Не меняйте последовательность навигационных элементов, которые повторяются на всех страницах сайта. Элементы страницы со схожей функциональностью называйте однотипно.

16. Текущая страница

Добавьте на страницу дублирующую навигацию («хлебные крошки») или карту сайта, чтобы показать, где сейчас находится пользователь относительно общей иерархии и других страниц.

17. Управление с клавиатуры

Дайте пользователю возможность управлять сайтом с клавиатуры без ограничений по времени на нажатие клавиши. Это поможет людям, которые не могут пользоваться мышкой, работать со страницей без препятствий — пролистывать, переходить по ссылкам, заполнять формы.

Рекомендация не относится к случаям, когда управление с клавиатуры невозможно, например при рисовании. Управление с клавиатуры не должно мешать управлению мышью.

Обеспечьте правильную последовательность перехода фокуса — проверьте семантику верстки, правильную последовательность содержимого на странице и добавьте визуальное отображение фокуса на активных элементах. Переход на следующий элемент — одно нажатие на клавишу *Tab*, на предыдущий — *Shift+Tab*. Если программно заданы другие клавиши, укажите это в начале сайта.

WGAC 2.1 рекомендует использовать атрибут `tabindex` — его следует заполнить положительными числами в порядке возрастания. Не рекомендуется использовать `tabindex` для всех элементов. Если на странице появятся новые элементы, в установленных значениях можно будет легко запутаться.

Полагайтесь на естественный порядок элементов на странице. Используйте `tabindex` для элементов только в специальных случаях, когда важна нестандартная последовательность. Используйте `tabindex="-1"`, чтобы пропустить элемент, и `tabindex="0"`, чтобы элемент оказался в фокусе. Выделите активный элемент визуально с помощью псевдокласса: `focus` для ссылок, кнопок и полей ввода.

Пример

Электронная почта

Пароль

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

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

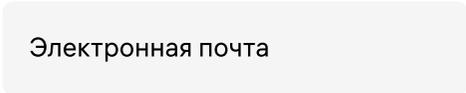
Избегайте использования обработчиков `mousedown` или `mouseup`, потому что они недоступны для клавиатуры. Если это невозможно, добавьте обработку событий клавиатуры.

Поля ввода

18. Поля ввода

Добавьте названия ко всем полям ввода в форме, включая радиокнопки, чек-боксы и выпадающие списки. Название должно быть кратким и отражать назначение поля. Добавьте название с помощью тега `<label>`, в атрибуте `for` пропишите `id` поля.

Пример



Электронная почта

Плохо

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

Хорошо

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

Если лейбл нужно скрыть, расположите его перед полем, используйте привязку атрибута `for` и специальный класс для вспомогательных технологий.

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

Этот класс скрывает лейбл визуально, но его прочтут скринридеры. Обратите внимание: скринридеры и другие вспомогательные технологии, как и браузеры, скрывают элементы с использованием `display: none` и `visibility: hidden`.

Еще один способ связывания поля и подписи — атрибут `aria-labelledby`. Он используется как атрибут `for` для тега `<label>`, значение этого атрибута совпадает со значением `id` поля.

Пример

Электронная почта

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

Название поля можно указать в атрибутах `aria-label` или `title`. Второй метод менее надежен, название будет видно всем пользователям в виде белого всплывающего элемента при наведении курсора на поле.

Добавьте к форме инструкции по заполнению полей, отметьте обязательные поля, чтобы избежать ошибок. Обязательные поля можно указать в названии поля `label`, добавив к полю атрибуты `required` или `aria-required="true"`.

Электронная почта

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

Часто разработчики применяют способ, позволяющий избежать использования атрибутов `for` и `id`. Для этого нужно «обернуть» тег `<input>` тегом `<label>`.

```
<label>Имя:<input type="text" name="firstname"></label><br>  
<label><input type="checkbox" name="subscribe">  
    Подписаться на рассылку</label>
```

19. Группы полей

Группировка связанных полей помогает разделить форму на части для облегчения восприятия.

Группируйте поля с помощью тегов `<fieldset>` и `<legend>`.

Тег `<fieldset>` — контейнер для связанных элементов формы, `<legend>` — заголовок группы, где можно прописать не только назначение группы, но и общие характеристики полей. Например, указать, что все поля группы обязательны.

Пример

Контакты

Электронная почта

Телефон

```
<fieldset>
  <legend>Контакты</legend>
  <label for="input-mail" id="label-mail">Электронная почта</label>
  <input type="email" id="input-mail" aria-labelledby="label-mail">

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

Другой способ группировки полей формы — использование атрибутов `role="group"` и `aria-labelledby`, для которых нужно указать `id` элемента, содержащего описание группы полей.

Пример

Контакты

Электронная почта

Телефон

```
<div role="group" aria-labelledby="user-data">  
  <div id="user-data">Контакты</div>  
  <label for="input-mail" id="label-mail">Электронная почта</label>  
  <input type="email" id="input-mail" aria-labelledby="label-mail">  
  
  <label for="input-phone" id="label-phone">Телефон</label>  
  <input type="text" id="input-phone" aria-labelledby="label-phone">  
</div>
```

Предоставьте пользователю понятные инструкции по вводу информации. Укажите обязательные поля, допустимые форматы данных и ограничения времени. Разместите инструкцию перед тегом `<form>`, чтобы скринридер прочитал инструкцию, прежде чем войдет в режим чтения форм.

Для простых инструкций используйте подсказки прямо в описании поля. Например, для поля даты можно написать его название.

Пример

Электронная почта
(обязательно)

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

20. Уведомления

Предоставьте пользователю информацию о результатах отправки формы. Если после отправки формы страница обновляется, измените для нее тег `<title>` на ошибку или добавьте заголовок `h1` с сообщением об ошибке.

Если страница не обновляется, добавьте список ошибок с атрибутом `role="alert"` над элементом формы. Свяжите каждую ошибку с полем при помощи атрибута `aria-describedby`. Атрибут `aria-describedby` можно обновлять динамически — в обычном состоянии он указывает на подсказку или описание поля, а в случае ошибки ввода — на текст ошибки (вместо подсказки или вместе с ней — `aria-describedby` может содержать несколько идентификаторов).

Пример

Неправильный формат электронной почты

Электронная почта	<input type="text" value="example.a"/>
Телефон	<input type="text" value="+7 921 123-45-67"/>

```
<p class="txt-danger" id="danger-mail">Неправильный формат электронной почты</p>
<p>
  <label for="input-mail">Электронная почта</label>
  <input type="email" id="input-mail" aria-describedby="danger-mail"
    class="textbox" aria-required="true">
</p>
<p>
  <label for="input-phone">Телефон</label>
  <input type="email" id="input-phone" class="textbox" aria-required="true">
</p>
```

21. Уведомления к полям

Уведомления к полям можно выводить после отправки формы или непосредственно во время ввода данных.

После отправки формы установите фокус в первое поле, чтобы пользователь узнал о результатах проверки каждого поля. Если произошла ошибка ввода, отображайте сообщение о ней в текстовом виде после поля. Свяжите ошибку с полем с помощью атрибута `aria-describedby`.

Для демонстрации сообщений в случае, когда поле проверяется во время ввода, добавьте после поля область с атрибутом `aria-live="polite"`. Если поле проверяется после потери фокуса, добавьте после поля атрибут `aria-live="assertive"`. Программно добавляйте ошибку в этот элемент. Отличие подходов заключается в том, что `aria-live="assertive"` прерывает все текущие задачи скринридера. Если применить его для проверки поля во время ввода, после каждого изменения скринридер будет переходить к этому элементу и читать его несколько раз.

Пример

Электронная почта	<input type="text" value="example.a"/>
	<small>Неправильный формат</small>
Телефон	<input type="text" value="+7 921 123-45-67"/>

```
[...]  
<label for="input-mail" id="label-mail">Электронная почта</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">  
  Неправильный формат электронной почты</small>  
[...]
```

Таблицы

22. Таблицы

Укажите взаимосвязи между ячейками, для обозначения ячеек используйте семантически правильные теги. Для ячеек-заголовков применяйте тег `<th>`. Для связывания ячейки с данными используйте атрибут `scope`. Это поможет слепым и слабовидящим пользователям ориентироваться в таблице — узнавать заголовки таблицы, правильно перемещаться по строкам и столбцам.

Для простых таблиц с одним заголовком используйте значения `row` или `col`.

Пример

Ф. И. О.	Возраст	Профессия
Иванов Александр Александрович	23	Врач
Китаева Анна Сергеевна	34	Учитель
Колосова Александра Иванова	32	Программист

```
<thead>
  <tr>
    <th scope="col">ФИО</th>
    <th scope="col">Возраст</th>
    <th scope="col">Профессия</th>
  </tr>
</thead>
```

Для таблиц с двумя заголовками используйте одновременно `row` для верхних заголовков и `col` для горизонтальных строк. Добавьте тег `<caption>` или атрибут `aria-describedby` для заголовка таблицы.

Пример

Участники проекта «Курьер»

Номер	Ф. И. О.	Возраст	Профессия
1	Иванов Александр Александрович	23	Врач
2	Китаева Анна Сергеевна	34	Учитель
3	Колосова Александра Иванова	32	Программист

```
<caption>Участники проекта</caption>
<thead>
  <tr>
    <th scope="col">№ строки</th>
    <th scope="col">ФИО</th>
    <th scope="col">Возраст</th>
    <th scope="col">Профессия</th>
  </tr>
</thead>
<tbody>
  <tr>
    <th scope="row">1</th>
    <th>Иванов Александр Александрович</th>
    <th>23 года</th>
    <th>Врач</th>
  </tr>
</tbody>
```

Изображения

23. Информативные изображения

Информативные изображения иллюстрируют текстовую информацию в наглядном виде, дополняя или обозначая ее. Для таких изображений добавьте атрибут `alt`.

Пример



Вкручивайте лампочку по часовой стрелке.

Плохо

```
 Вкручивайте лампочку по часовой стрелке.
```

Хорошо

```
 Вкручивайте лампочку по часовой стрелке.
```

24. Декоративное изображение

Декоративные изображения следует добавлять с помощью свойства CSS `background-image` — тогда скринридеры будут их игнорировать. Если декоративный элемент представлен в виде изображения, например внутри ссылки, добавьте к нему пустой `alt=""` или атрибуты `role="presentation"` и `aria-hidden="true"`.

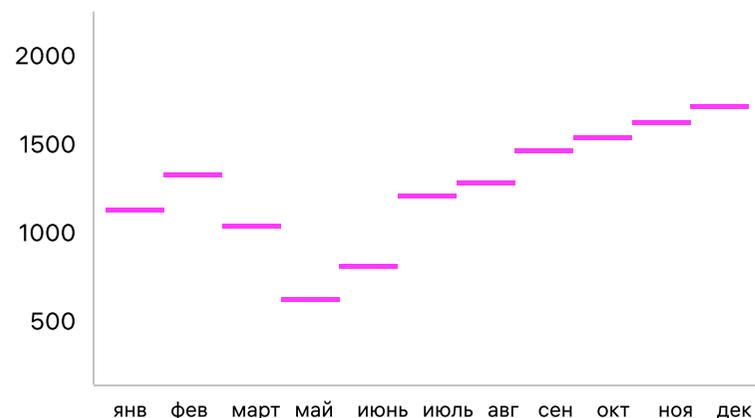
25. Графики, диаграммы

Для сложных изображений — карт, диаграмм, графиков — добавьте в атрибут `alt` краткое описание, а подробности вынесите на отдельную страницу, поместив их в тег `<figcaption>`, в атрибуты `longdesc` или `aria-describedby`.

Добавьте ссылку на другую страницу или на фрагмент текущей страницы с полным описанием. Минус этого подхода — ссылка будет не связана с изображением по смыслу.

Пример

Посетителей на сайте



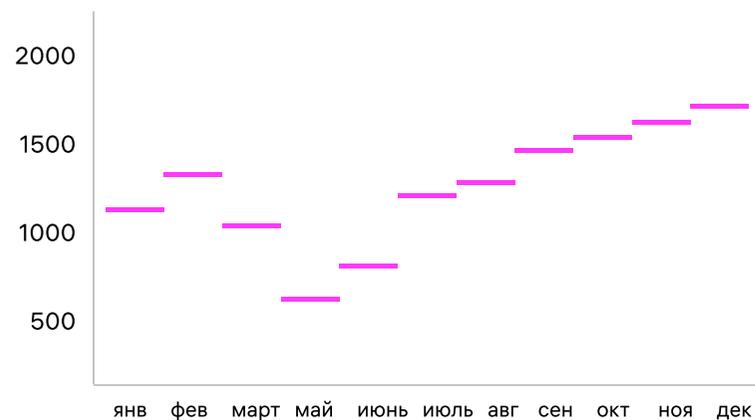
```

<a href="chart-info.html">Описание изображения</a>
```

Добавьте в `<figure>` с атрибутом `role="group"` само изображение и его описание. Описание «оберните» в тег `<figcaption>`.

Пример

Посетителей на сайте

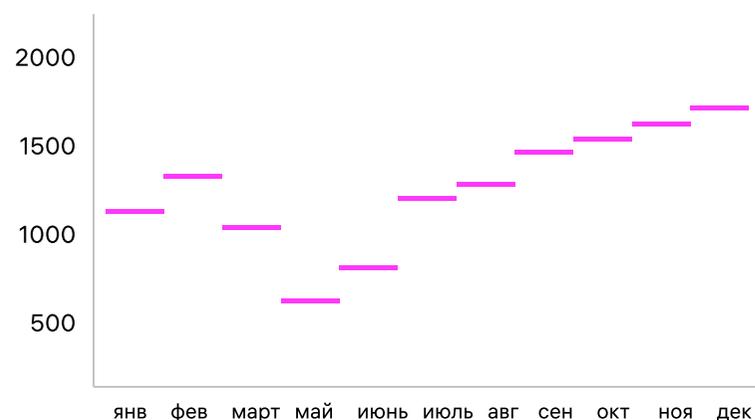


```
<figure role="group">
  
  <figcaption>
    <a href="chart-bar.html">
      График посещений сайта so-edinenie.org за 2021 год</a>
    </figcaption>
</figure>
```

Добавьте к изображению атрибут `longdesc` или `aria-describedby`. Пропишите в нем `id` элемента, содержащего подробное описание изображения. Различия между примерами заключаются в том, что в элементе с `id` из `aria-describedby` должен быть только текст, в описание элемента с `id` из `longdesc` можно добавлять дополнительные теги.

Пример

Посетителей на сайте



```

<p id="chart-longdesc">
  График посещений сайта so-edinenie.org за 2021 год: в октябре сайт посетили
  26 тысяч человек, это самое большое количество посещений в 2021 году.
</p>
```

26. Группы изображений

Для групп изображений, представляющих одну и ту же информацию, например, звездочек рейтинга, добавьте описание в **alt** только к первому изображению.

Пример

Рейтинг фильма 4 из 5



```
<img src=>images/i-star.png> alt=>Рейтинг фильма 4 из 5>  
  
  
  

```

Благодарность

Приемы и примеры реализации были составлены на основе разработок команды проекта «[Веблайнд](#)» при поддержке «[Школы Бюро Горбунова](#)».

Автор Анастасия Суховерхова.

Эксперты Алексей Любимов, Вадим Макеев и Александр Нефедов.



О фонде

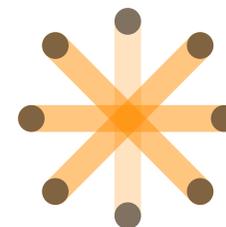
Благотворительный фонд поддержки слепоглухих «Со-единение»

Основание фонда в 2014 году стало переломным моментом в жизни слепоглухих людей и позволило им выйти из социальной изоляции.

Курсы компьютерной грамотности и социальной реабилитации, центры сопровождаемого проживания и повышение квалификации тифлосурдопереводчиков, разработка технических средств реабилитации, гаджетов и мобильных приложений, облегчающих жизнь людям с нарушениями слуха и зрения, развитие инклюзивного театра в России — все это стало элементами масштабного пазла, который называется системой поддержки людей с одновременным нарушением слуха и зрения.

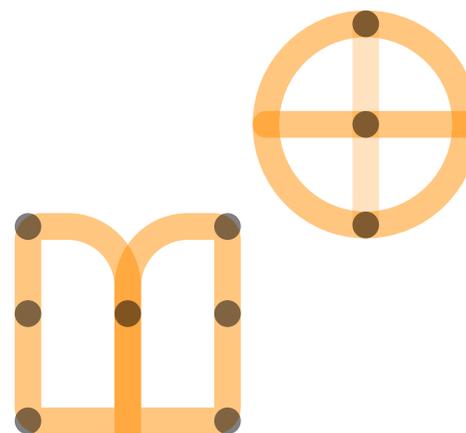
В условиях пандемии важнейшей задачей для фонда является формирование доступной цифровой образовательной среды.

Ключевым проектом 2021 года станет разработка кортикального импланта. В этом году мы планируем представить прототип первой технологии восстановления зрения с помощью импланта, который вживляется в головной мозг.



На сегодняшний день мы выявили более 4500 слепоглухих людей (по оценке экспертов, всего в России их около 15 000).

Результат нашей ежедневной работы имеет значение как для слабовидящих и слепых людей (около 218 000), так и для слабослышащих и глухих (более 13 млн).



Достижения

- Разработана методика сопровождаемого проживания слепоглухих взрослых: две площадки в Москве и области, в 2020 году открыт третий «Дом сопровождаемого проживания» в Северо-Западном федеральном округе, который станет и центром реабилитации и обучения.
- Открыто 8 театральных школ «Инклюзион» (Москва, Санкт-Петербург, Екатеринбург, Казань, Новосибирск, Калининград, Орск); 25 февраля 2021 года будет открыта девятая школа в Кургане.
- Качественный технологический прорыв для России: в 2017 году были проведены две первые операции по имплантации бионического глаза totally blind человеку, что позволило частично вернуть зрение двум подопечным фонда.
- Открыт 41 досуговый центр в 27 регионах России.
- Основано 15 ресурсных центров для слепоглухих по всей России.
- 38 региональных отделений «Сообщества семей слепоглухих» помогают людям сегодня.
- Работают 3 обучающих центра для взрослых слепоглухих.
- В рамках выхода на международный уровень было налажено сотрудничество с Sense International, Институтом Перкинса, Институтом Кенталис, DBI.
- Брайлевские дисплеи включены в перечень ТСР, предоставляемых государством.



Досуговые, обучающие и ресурсные центры в регионах России

