**Доступность** – это возможность использования интерфейса людьми с различными особенностями. Во многих странах доступность – это часть законодательства, касающегося цифровых сервисов. Подробнее можно почитать [тут](https://www.w3.org/Translations/WCAG20-ru/). Основные принципы доступности интерфейсов:

* **Воспринимаемость** – интерфейс должны суметь воспринять любые люди. Например, важно дублировать источники информации: показывать ошибку не только красным цветом, но и текстом.
* **Управляемость** – нужно думать об альтернативных сценариях управления интерфейсом. Например, не только мышкой, но и клавиатурой.
* **Понятность** – не самое простое правило, ведь даже ребёнок видит мир не так, как взрослый. Например, можно делать единообразную навигацию по сайту.
* **Надёжность** – нужно, чтобы интерфейсы работали с разными технологиями, которые упрощают людям использование интернета. Например, вся страница должна корректно считываться скринридерами.

Ключевые особенности доступности интерфейсов:

* Наличие альтернативных способов донесения информации (субтитры в видео, озвучка кнопок, описание изображений).
* Резкость анимации (отсутствие скримеров, быстрых мерцаний блоков для эпилептиков).
* Читаемость (размер шрифта менее 16 пикселей трудночитаем для людей с дислексией).
* Контрастность, или разница в яркости и цвете, которая делает объект различимым. Для его проверки можно использовать плагины для *Figma* [*Contrast*](https://www.figma.com/community/plugin/748533339900865323) или [*A11y Color Contrast Checker*](https://www.figma.com/community/plugin/733159460536249875/A11y---Color-Contrast-Checker).

Включить эмулятор особенностей зрения можно в инструментах разработчика с помощью команд:

*“Ctrl + Shift + P” -> “Rendering” -> “Emulate vision deficiencies”*

Для корректной работы скринридеров браузер строит из *HTML*-кода **дерево доступности** (*accessibility tree*). Каждый элемент описывается через набор параметров:

* Роль элемента. Например, у заголовка роль “*heading*”, у блока *nav* роль “*navigation*” и т.д. Полный список можно найти [тут](https://doka.guide/a11y/).
* Имя элемента, которое не всегда есть. Например, у заголовков или кнопок оно берется из текста внутри, но у секций или абзацев его нет.

Список популярных скринридеров:

* Для *Windows* – [*JAWS*](https://www.freedomscientific.com/products/software/jaws/) (платный), [*NVDA*](https://www.nvaccess.org/about-nvda/) (бесплатный) и [*Narrator*](https://support.microsoft.com/ru-ru/windows/%D0%BF%D0%BE%D0%BB%D0%BD%D0%BE%D0%B5-%D1%80%D1%83%D0%BA%D0%BE%D0%B2%D0%BE%D0%B4%D1%81%D1%82%D0%B2%D0%BE-%D0%BF%D0%BE-%D0%B8%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8E-%D1%8D%D0%BA%D1%80%D0%B0%D0%BD%D0%BD%D0%BE%D0%B3%D0%BE-%D0%B4%D0%B8%D0%BA%D1%82%D0%BE%D1%80%D0%B0-e4397a0d-ef4f-b386-d8ae-c172f109bdb1) (бесплатный и предустановленный).
* Для *MacOS* и *iOS* – [*VoiceOver*](https://www.apple.com/ru/accessibility/vision/) (предустановлен).
* Для *Android* – [*TalkBack*](https://support.google.com/accessibility/android/topic/3529932) (предустановлен).
* Для *Linux* – [*Orca*](https://help.gnome.org/users/orca/stable/) (предустановлен).

Появление и пропадание важных для понимания смысла элементов по наведению мыши – это **антипаттерн**. На мобильных устройствах нет мыши.

Для сокрытия элементов из дерева доступности используют следующие способы:

*<button hidden>Кнопка Шрёдингера</button> // атрибут hidden в html*

*.hidden-block { // свойство display в css*

*display: none;*

*}*

*.another-hidden-block { // свойство visibility в css*

*visibility: hidden;*

*}*

Атрибут ***hidden*** и свойство ***display*** не только скрывают элемент, но и вырывают его из потока элементов.

В качестве альтернативы можно скрыть элемент визуально, но оставить его в дереве доступности:

*.visually-hidden {*

*position: absolute; // вырвем элемент из потока*

*width: 1px; // сделаем элемент очень маленьким, но чтобы не пропал совсем*

*height: 1px;*

*margin: -1px; // уберем элемент из поля зрения, сдвинув на размер самого себя*

*padding: 0; // обнулим отступы, которые есть у некоторых элементов по умолчанию*

*border: 0; // уберем границу, которая может быть по умолчанию у элементов*

*clip: rect(0 0 0 0); // обрежем видимую область до нулевого значения*

*overflow: hidden; // на всякий случай сокроем видимое за пределами блока*

*}*

*<button class="header\_\_menu-button">*

*<span class="visually-hidden">Открыть меню</span> // сокроем описание кнопки*

*</button>*

Изображение, которые влияют на смысл страницы, называются **контентными**, а те, что оформляют смысл, называются **декоративными**. Контентные изображения должны быть в *HTML*, чтобы их озвучил скринридер. Декоративные изображения не должны озвучиваться скринридером (например, внешний вид кнопки), поэтому они должны быть в *CSS* или скрыты другими способами. Например:

* первая реализация вставляет контентное фото в качестве фона без озвучки скринридером

*<div class="card\_\_image-wrapper dalai-lama-photo">*

*<div class="card\_\_description">*

*<h2 class="card\_\_title">Далай Лама</h2>*

*<p class="photo-copyright">Фото: агентство Reuters</p>*

*</div>*

*</div>*

* вторая реализация вставляет контентное фото картинкой с озвучкой скринридера

*<div class="card\_\_image-wrapper">*

*<img class="card\_\_image"*

*src="./images/dalai-lama.jpg"*

*alt="Далай Лама в монашеской одежде, показывает двумя пальцами символ мира."*

*/>*

*<div class="card\_\_description">*

*<h2 class="card\_\_title">Далай Лама</h2>*

*<p class="photo-copyright">Фото: агентство Reuters</p>*

*</div>*

*</div>*

***Accessible Rich Internet Applications***, или ***ARIA*** – это доступные многофункциональные интернет-приложения, или дополнительные инструменты, которые помогают сделать интерфейс более доступным. Полный список можно посмотреть [тут](https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Attributes).

Атрибут ***role*** управляет именами элементов.

Атрибут ***aria-hidden*** управляет видимость элементов в дереве доступности.

Атрибут ***aria-label*** позволяет задать имена-метки для элементов.

Атрибут ***aria-description*** позволяет задать описание для элементов.

*<header class="header" aria-label="Шапка сайта">*

Атрибут ***aria-labelledby*** задает имя через ссылку на *id*.

Атрибут ***aria-describedby*** задает описание через ссылку на *id*.

*<article class="card"*

*aria-labelledby="hazaiya-heading"*

*aria-describedby="hazaiya-description"*

*>*

*<blockquote class="quote">*

*<p class="quote-text">*

*«Счастлив тот, кто вошел в сокровищницу внутри себя и увидел звезду»*

*</p>*

*<div class="qoute-wrapper">*

*<p class="quote-author" id="hazaiya-heading">Иосиф Хаззайя</p>*

*<p class="quote-description" id="hazaiya-description">*

*Монах, писатель, богослов, восточно-сирийский мистик Ассирийской Церкви.*

*</p>*

*</div>*

*</blockquote>*

*</article>*

Атрибут ***aria-controls*** указывает на *id* элемента, которым управляет кнопка.

Атрибут ***aria-expanded*** сообщает скринридеру статус элемента (свернут *false* или раскрыт *true*).

Атрибут ***aria-selected*** подсказывает через скринридер, выбрана ли в текущий момент опция табуляции (*true*) или нет (*false*).

*<button class="header\_\_menu-button"*

*aria-controls="main-menu"*

*aria-expanded="false"*

*>*