



Frontender[1.0] HTML Accessibility (Доступность). Accessibility Tree

🔗 YouTube	https://youtu.be/XUnV0Y-aSwo
🔗 Telegram	https://t.me/Dmitry_Kolotilshikov
# Номер урока	52



Полезные ссылки:

Гайды по Accessibility: <https://www.w3schools.com/accessibility/>

ARIA атрибуты: <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Attributes>

Роли (role): <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Roles>

Подбор контрастности цветов: <https://contrast-ratio.com/>

Сравнение :focus и :focus-visible: <https://css-tricks.com/almanac/selectors/f/focus-visible/>

Про tabindex: <https://dequeuniversity.com/rules/axe/4.6/tabindex>

Дока по aria-labelledby: https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Attributes/aria-labelledby#benefits_and_drawbacks



HTML Accessibility, также известный как "веб-доступность", это процесс создания веб-страниц, которые могут быть использованы и поняты людьми с различными формами инвалидности и ограничений, включая низкое зрение, слабослышащих, людей с ограниченными возможностями движения и т.д.



ARIA (Accessible Rich Internet Applications) - это набор атрибутов и ролей, который помогает создавать веб-приложения более доступными для пользователей с ограниченными возможностями. **Эти атрибуты** позволяют добавлять информацию **о семантике и доступности** элементов на странице, в частности для пользователей с ограниченными возможностями, использующих программы чтения экрана. Например, помочь слабовидящим пользователям понимать, что элемент является кнопкой или ссылкой, а не просто текстом.

Использование ролей:

```
<div role="menu">
  <a href="#" role="menuitem">Пункт меню 1</a>
  <a href="#" role="menuitem">Пункт меню 2</a>
  <a href="#" role="menuitem">Пункт меню 3</a>
</div>
```



Атрибут `role` позволяет задать роль элемента, указывающую на то, как он используется в контексте страницы. Примеры значений для `role` могут включать "button", "link", "menu", "searchbox" и т. д.

Например, для элемента, который выглядит как кнопка, но не имеет соответствующего тега, можно указать `role="button"`, чтобы программы чтения экрана понимали, что это кнопка и должна обрабатываться соответствующим образом.

```
<div role="button" aria-label="Нажмите, чтобы открыть меню">Меню</div>
```

Атрибуты `aria-*` позволяют добавить дополнительную информацию о доступности, такую как описание элемента, состояние элемента или его свойства. Примеры атрибутов `aria-*` могут включать `aria-label`, `aria-hidden`, `aria-checked`, `aria-expanded` и т. д.

```
<button aria-label="Отправить форму" aria-disabled="true">Отправить</button>
```

Здесь мы используем `aria-label` для задания текстового описания элемента, а `aria-disabled` для указания, что кнопка неактивна.

1. Добавление информации о роли элемента и описания его состояния:

```
<div role="progressbar" aria-valuenow="75" aria-valuemin="0" aria-valuemax="100">
  Загрузка...
</div>
```


Здесь мы используем `role="progressbar"` для указания, что элемент является индикатором прогресса, и атрибуты `aria-valuenow`, `aria-valuemin` и `aria-valuemax` для указания текущего значения, минимального и максимального значения соответственно.


1. Улучшение доступности элемента с помощью добавления текстового описания:

```

```

Здесь мы используем атрибут `alt` для указания альтернативного текста для изображения и `aria-label` для добавления дополнительного текстового описания, которое будет использоваться программами чтения экрана.

 **ARIA-атрибуты** могут быть **мощным инструментом** для **улучшения доступности (Accessibility)** веб-приложений. Однако необходимо использовать их правильно, чтобы не нарушать структуру и **семантику HTML** и не ухудшать доступность контента.

 атрибут **accesskey** Как нажимать в разных браузерах и системах:

	Windows	Linux	Mac
Firefox	<div>Alt + Shift + key</div>		<div>On Firefox 57 or newer: Control + Option + key Or Control + Alt + key</div> <div>On Firefox 14 or newer: Control + Alt + key</div> <div>On Firefox 13 or older: Control + key</div>
Edge	N/A	<div>Control + Option + key</div> <div>Control + Option + Shift + key</div>	
Google Chrome	<div>Alt + key</div>		
Safari	N/A		
Opera 15+	<div>Alt + key</div>		<div>Control + Alt + key</div>
Opera 12	<div>Shift + Esc</div> opens a contents list which are accessible by accesskey, then, can choose an item by pressing <div>key</div>		

Accessibility tree в Chrome


Heading 1

Heading 2

Click WClick S

CUSTOM BUTTON

[just empty simple link](#)



▼ RootWebArea "Accessibility" focusable: true focused: true

- ▼ heading "Heading 1"
 - StaticText "Heading 1"
- ▼ heading "Heading 2"
 - StaticText "Heading 2"
- ▼ button "simple button click W" focusable: true keyshortcuts: Alt+w
 - StaticText "Click W"
- ▼ button "simple button click S" focusable: true keyshortcuts: Alt+s
 - StaticText "Click S"
- LineBreak " "
- LineBreak " "
- ▼ button "simple custom button click X" focusable: true keyshortcuts: Alt+x
 - StaticText "CUSTOM BUTTON"
- LineBreak " "
- ▼ link "empty simple link" focusable: true keyshortcuts: Alt+a
 - StaticText "just empty simple link"
- LineBreak " "
- img "boy with goat"
- ▼ heading "Heading 1"
 - StaticText "Heading 1"
- ▼ heading "Heading 2"
 - StaticText "Heading 2"
- ▼ button "simple button click W" focusable: true keyshortcuts: Alt+w
 - StaticText "Click W"
- ▼ button "simple button click S" focusable: true keyshortcuts: Alt+s
 - StaticText "Click S"
- LineBreak " "
- LineBreak " "
- ▼ button "simple custom button click X" focusable: true keyshortcuts: Alt+x
 - StaticText "CUSTOM BUTTON"
- LineBreak " "
- ▼ link "empty simple link" focusable: true keyshortcuts: Alt+a
 - StaticText "just empty simple link"
- LineBreak " "
- img "boy with goat"

Accessibility

Accessibility Tree

☒

Enable full-page accessibility tree

The accessibility tree moved to the top right corner of the DOM tree. [Send](#)

ARIA Attributes

No ARIA attributes

Computed Properties

Accessibility node not exposed
Element is not rendered.

Source Order Viewer

No source order information available

ACCESSIBILITY TREE

Frontender[1.0] HTML Accessibility (Доступность). Accessibility Tree

3

ПРИМЕРЫ КОДА:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Accessibility | Доступность</title>
  <style>
    img {
      width: 300px;
    }
    /* body {
      background-color: darkgray;
    } */
  </style>
</head>
<body>
  <h1 tabindex="-1" aria-label="заголовок 1">Привет</h1>
  <h2 tabindex="0" aria-label="заголовок 2">Мир</h2>

  <!-- tabindex="0" - zero is a normal tab flow -->

  <button aria-label="просто кнопка 1" accesskey="w">
    кнопка 1
  </button>
  <button aria-label="просто кнопка 2" accesskey="e">
    кнопка 2
  </button>

  <br>
  <br>

  <a href="#" aria-label="просто ссылка вникуда">Просто ссылка</a>

  <hr>

  <form aria-labelledby="form-label">
    <label id="form-label" hidden>Form for email</label>

    <label for="email-id" aria-label="Email field">Email</label>
    <input type="email" id="email-id" name="email" placeholder="enter your email" required autocomplete="off">
  </form>

  <hr>

  <section aria-label="Section 1">
    <button aria-describedby="text-id">Просто кнопка</button>

    <p id="text-id" aria-label="text">
      Далеко-далеко за словесными горами в стране гласных и согласных живут рыбные тексты.
    </p>
  </section>

  <hr>

  <!--tricky situation | хитрая ситуация - aria-label будет подавлена aria-labelledby-->
  <section aria-label="Section 2">
    <button aria-label="Синий" aria-labelledby="color-id">красный</button>

    <span id="color-id">желтый</span>
    <!--will be yellow value in accessibility tree, but not on the page and DOM-->
  </section>

  <hr>

  <a href="#">оригинальная ссылка</a>

  <div role="link">моя ссылка</div>

  <div role="button" aria-label="просто кнопка">моя кнопка</div>

  <ul>
    <li>1</li>
    <li>2</li>
    <li>3</li>
  </ul>

  <div role="list">
```



```

        <div role="listitem">1</div>
        <div role="listitem">1</div>
        <div role="listitem">1</div>
    </div>
</body>
</html>

```

Heading 1

Heading 2

Click W Click S

CUSTOM BUTTON

just empty simple link



type your email

show window

A modal window will be shown on the page

Red Yellow

The screenshot shows the Lighthouse Accessibility panel in Chrome DevTools. The left pane displays the DOM tree with the following structure:

- RootWebArea "Accessibility" focusable: true
 - heading "Heading 1"
 - StaticText "Heading 1"
 - heading "Heading 2"
 - StaticText "Heading 2"
 - button "simple button click W" focusable: true shortcuts: Alt+W
 - StaticText "Click W"
 - button "simple button click S" focusable: true shortcuts: Alt+S
 - StaticText "Click S"
 - LineBreak " "
 - LineBreak " "
 - button "simple custom button click X" focusable: true shortcuts: Alt+X
 - StaticText "CUSTOM BUTTON"
 - LineBreak " "
 - link "empty simple link" focusable: true shortcuts: Alt+A
 - StaticText "just empty simple link"
 - LineBreak " "
 - img "boy with goat"
 - form "form with example label and input"
 - LabelText "type your email"
 - textbox "type your email" focusable: true settable: true multiline: false readonly: false required: true
 - separator "" settable: true
 - region "section with example aria-describedby"
 - button "show window" focusable: true
 - paragraph ""
 - region "section with example aria-labelledby"
 - button "Yellow" focusable: true
 - StaticText "Yellow"

The right pane shows the Accessibility Tree, which is currently empty. Below it, the ARIA Attributes section shows the following information:

- aria-labelledby: form-label
- Computed Properties
- Name: "form with example label and input"
- aria-labelledby:
 - label#form-label "form with example label and input"
 - aria-label: Not specified
 - title: Not specified
 - Role: form
 - Labeled by: label#form-label
- Source Order Viewer
 - Show source order

Accessibility audit in Lighthouse

Heading 1

Heading 2

Click W Click S

CUSTOM BUTTON

just empty simple link



type your email

show window

A modal window will be shown on the page

Red Yellow

The screenshot shows the Lighthouse report for the URL `http://127.0.0.1:5500/index.html`. The report is titled "Accessibility" and features a large green circle with the score "100". Below the score, it states: "These checks highlight opportunities to [improve the accessibility of your web app](#). Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged."

The report is divided into three sections:

- ADDITIONAL ITEMS TO MANUALLY CHECK (10)**: This section is currently empty. A link to [conducting an accessibility review](#) is provided.
- PASSED AUDITS (20)**: This section is currently empty.
- NOT APPLICABLE (24)**: This section is currently empty.

At the bottom, the report details the capture information:

- Captured at**: Apr 19, 2023, 2:26 PM GMT+5
- Initial page load**
- Emulated Desktop with Lighthouse 10.0.1**
- Custom throttling**
- Single page load**
- Using Chromium 112.0.0.0 with devtools**

The footer indicates: "Generated by Lighthouse 10.0.1 | [File an issue](#)"

Rule Description

A `tabindex` attribute must never have a value greater than 0 to prevent an unexpected tab order that can give the appearance of skipping some elements entirely.