

Accessibility

WAI

WAI is a **Web Accessibility Initiative** that was created by W3C comand.

This iniciative provede a few standarts and specification that alow make web application more accessible for whole peoples.

The **W3C Web Accessibility Initiative** (WAI) provides a set of guidelines that are internationally recognized as the standard for web accessibility. These include:

- **Web Content Accessibility Guidelines (WCAG)**
- **User Agent Accessibility Guidelines (UAAG)**
- **Authoring Tool Accessibility Guidelines (ATAG)**

WAI guidelines grouped for four principles:

- perceivable - information must be perceivable for users in ways that it can perceive
- operable - interfaces must be operatable (navigation, inputs, buttons)
- understandable - content information and interact actions must be understhandable
- robust - content must be robust for assistive technologies (screen readers ...)

ATAG

Authoring tools are software and services that “authors” (web developers, designers, writers, etc.) use to produce web content. For example: HTML editors, content management systems (CMS), and websites that let users add content, such as blogs and social networking sites.

UAAG

User agents include browsers, browser extensions, media players, readers, and other applications that render web content.

WCAG

Web “content” generally refers to the information in a web page or web application

[WCAG2.1](#)

[WIA-ARIA](#)

[Accessibility tree](#)

[Web accessibility laws in EU and USA](#)

WCAG2.1

WCAG is a **Web Content Accessibility Guidelines**.

WCAG is a documentation that describe how to make web applications suitable and accessible for people with disabilities.

WCAG2.1 include 12 guidelines.

For each guideline WCAG has success test criterias (A, AA, AAA):

- “A” has 25 success criterias
- “AA” has 45 success criterias (include previous)
- “AAA” has 78 success criterias (include previous)

Web accessibilities is not only for peoples with disabilities. You can also use some accesibility things to make your website more comfortable for all user.

WCAG has guidelines organized under 4 principles: perceivable, operable, understandable, and robust.

There are there basic usage of accessibility tools for people with disabilities:

- screen readers (for blind or people with bad vision).
- voice or keyboard controls to input and manipulate with data (alternative for mouse and basic usage for people with limited move control).
- audio transcripts (for people who cant hear).

Some basics accsesability considerations:

- Keyboard compatibility (usage without mouse)
- Video caption (subtitles and transcript)
- Colors and good contrast (design patterns to make information more acceptable/suitable)
- Customizable text (people with bad vision or disleksia could customise text on the page to make it more comfortable and clarity without lossing context)

- Clear layout and design (without unreadable design, layout should be with clear headers,, navigation, content and footer)
- Voice Recognition (use voice recognition or voice assistants to control or input information. It is good for people with less or limited move control)
- Text to speech (converting text to speech for blind or bad vision people or for people with disleksia)
- Undersandable content (do not use complex language, jargon, unexpained achronyms)
- Large Links, Buttons, Controls (do not make these thinks small and check it on different devices, for instanse, on mobile)
- Notification and feedback (use understandable suces, error or information notification)

Tips:

10 WCAG Tips for designers

10 WCAG Tips for developers

10 WCAG Tips for developers

1. Associate each input (formControl) with label

```
<label for="username">Username</label>
<input id="username" type="text" name="username">
```

2. Include alternative text for images
3. Identify language and language changes for each page

Use lang attribute in html teg (<html lang="en">)

4. Use markUp to convey structure and information

For instance, use semantic tegs: nav, side, article, section and other to provide a structure and information in HTML.

Use WIA ARIA roles to indentify functionality (role="search"):

```
<form action="#" method="post">
  <div role="search">
    <label for="search">Search for</label>
    <input type="search" id="search" aria-describedby="search-help">
    <div id="search-help">Search records by customer id or name</div>
    <button type="submit">Go</button>
  </div>
</form>
```

5. Provide user clear description and error information to avoid mistakes (error messages, examples)

```
<label for="phone">Phone</label>
<input id="phone" name="phone" type="tel"
  pattern="^\(?(?0[1-9]{1}\)?)?[0-9 -]*$"
  aria-describedby="phone-desc">
<p id="phone-desc">For example, (02) 1234 1234</p>
```

6. Reflect the reading order in the code order


Avoid missing order of the elements during reading. You can check it while disabling styles.

Example: Reflecting the logical reading order in the code



Space trainers

Space trainer for a classic and stylish look.

 Add to cart

✖ Image before heading may be missed

```

<h3>Space trainers</h3>
<p>Space...</p>
<a href="...">Add to cart</a>
```

 View complete code example

✔ Heading marks the start of the section

```
<h3>Space trainers</h3>

<p>Space...</p>
<a href="...">Add to cart</a>
```

7. Write code to adapt content for user technology (Responsive, and different viewports)

```
/* On narrow viewports,
make the navigation full width */
@media screen and (min-width: 25em) {
  #nav {
    float: none;
    width: auto;
  }
  #main {
    margin-left: 0;
  }
}
```

```
/* On wider viewports,
put the navigation on the left */
@media screen and (min-width: 43em) {
  #nav {
    float: left;
    width: 24%;
  }
  #main {
    margin-left: 27%;
  }
}
```

8. Provide meaning for non-standard interactive elements

You should use WIA ARIA to describe non standard interactive elements and their state

```
<nav aria-label="Main Navigation" role="navigation">
  <ul>
    <li><a href="...">Home</a></li>
    <li><a href="...">Shop</a></li>
```

```

<li class="has-submenu">
  <a aria-expanded="false" aria-haspopup="true" href="#">SpaceBears</a>
  <ul>
    <li><a href="#">SpaceBear 6</a></li>
    <li><a href="#">SpaceBear 6 Plus</a></li>
  </ul>
</li>
<li><a href="#">MarsCars</a></li>
<li><a href="#">Contact</a></li>
</ul>
</nav>

```

9. Provide keyboard accesable for all alternative interactive elements

Think about keyboard access, especially when developing interactive elements, such as menus, mouseover information, collapsable accordions, or media players.

Use `tabindex="0"` to add an element that does not normally receive focus, such as `<div>` or ``, into the navigation order when it is being used for interaction. Use scripting to capture and respond to keyboard events.

[Article about tabindex=0](#)

10. Avoid CAPTCH where possible

CAPTCHAs create problems for many people.

If it not posible to avoid capture you should make CAPTHCHA uderstandable and clear

Additional rules:

Heading and landmarks

You should use corect h1, h2, h3 order at your website. Ordinary, you should use one h1 for name of the website or website logo, h2 for sections name and h3 for sections titles.

Do not worry about css styles for h3 or h2. You can make then larger than h1 and prewious ordered teg.

Landmarks tags are **main**, **nav**, **aside**. Define major section of your website by using lanmark tags. Do not use alot of landmarks, for instance, one **main** for whole page.

Use **skiplinks** to avoid focus on navigation twice and go to the main content

```
<a class="skip-link" href="#main">Skip to main</a>
...
<main id="main">
  [Main content]
</main>
```


10 WCAG Tips for designers

1. Provide sufficient contrast between foreground and background

Example: Contrast ratio

✖ Insufficient

Some people cannot read text if there is not sufficient contrast between the text and background. For others, bright colors (high luminance) are not readable; they need low luminance.

✔ Sufficient

Some people cannot read text if there is not sufficient contrast between the text and background. For others, bright colors (high luminance) are not readable; they need low luminance.

2. Do not use colors alone to convey information

It means that you can't use only color to show information. You should provide an additional symbols or numbers to show information unique (not only with color)

Example: Using color to convey meaning

✖ Color only

Required fields are in red

Name

Email

✔ Color and symbol

Required fields are in red and marked with an *

Name

Email *

Example: Refer to something using color alone

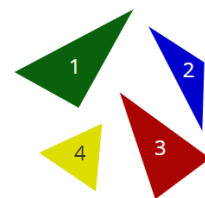
✖ Color only



Which is the right-angled triangle?

- ☐ Green
- ☐ Blue
- ☐ Red
- ☐ Yellow
- ☐ Don't know

✔ Color and number



Which is the right-angled triangle?

- ☐ Green (1)
- ☐ Blue (2)
- ☐ Red (3)
- ☐ Yellow (4)
- ☐ Don't know

3. Interactive elements should be easily identifiable

For instance, when you use links on the page, you should make it interactive (change color, cursor, shadow) while user hovers on it.

4. Clear and consistent navigation options

Make navigation consistent and make understandable where user situated.

5. Form input elements should have clear and good associated labels

Labels should be on the left side or above input element. Avoid having too much spacing between labels and fields.

6. Provide easily identifiable feedback

Use colors with special symbols. Notify user about his interactions with website.

Add a comment

Required fields are in red and marked with an *

Name

Superbear

⚠ E-mail *

superbear@@hq.example.com

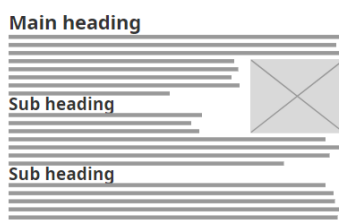
Website

⚠ Comment *

7. Use spacing and clear relationship between content

Example: Spacing highlights relationship between content

⊗ Little spacing and unclear relationship

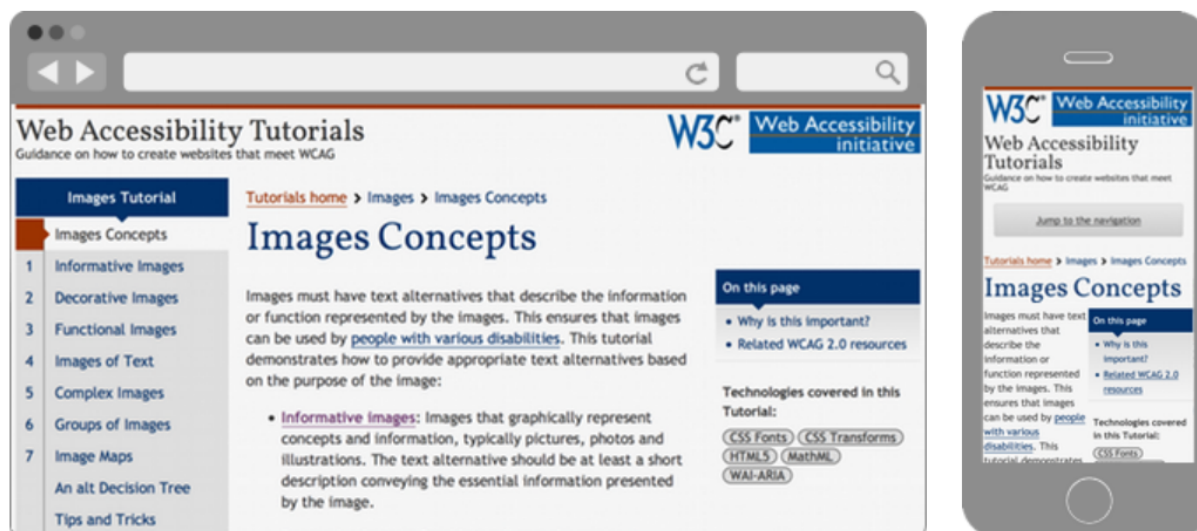


✔ More spacing and clearer relationship



8. Create or adapt design for different viewports and devices

Example: Content and navigation adapt to smaller mobile screen

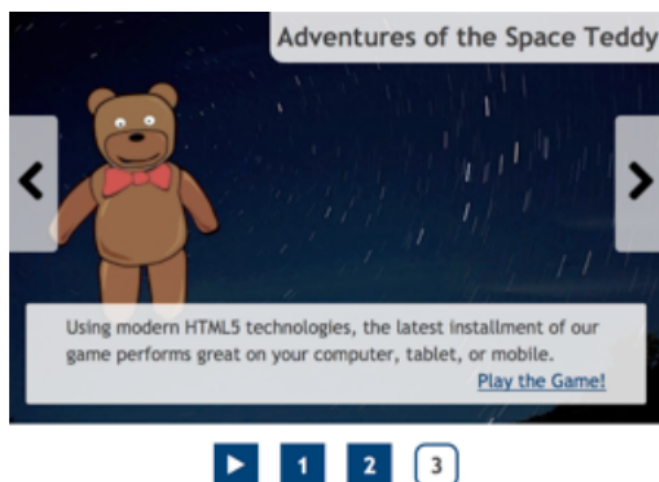


9. Use images and media alternatives in your content

That means you could provide information in different variations and for different users. You should use alternative text, captions and transcriptions, text labels for images.

10. Provide controls for content that starts automatically

Example: Show play/stop and selection controls in carousel design



Accessibility tree

Accessibility tree is a tree that browser make from DOM tree and provide information about elements for assistive tools (screen readers and e.c.t).

Why it needs?

Accessibility tree allow special assistive api's to make alternative interfaces (hear and spoken representation, voice input, mouse click emulation and other...) It is allow users with disabilities to interact and use your website.

Semantic in HTML

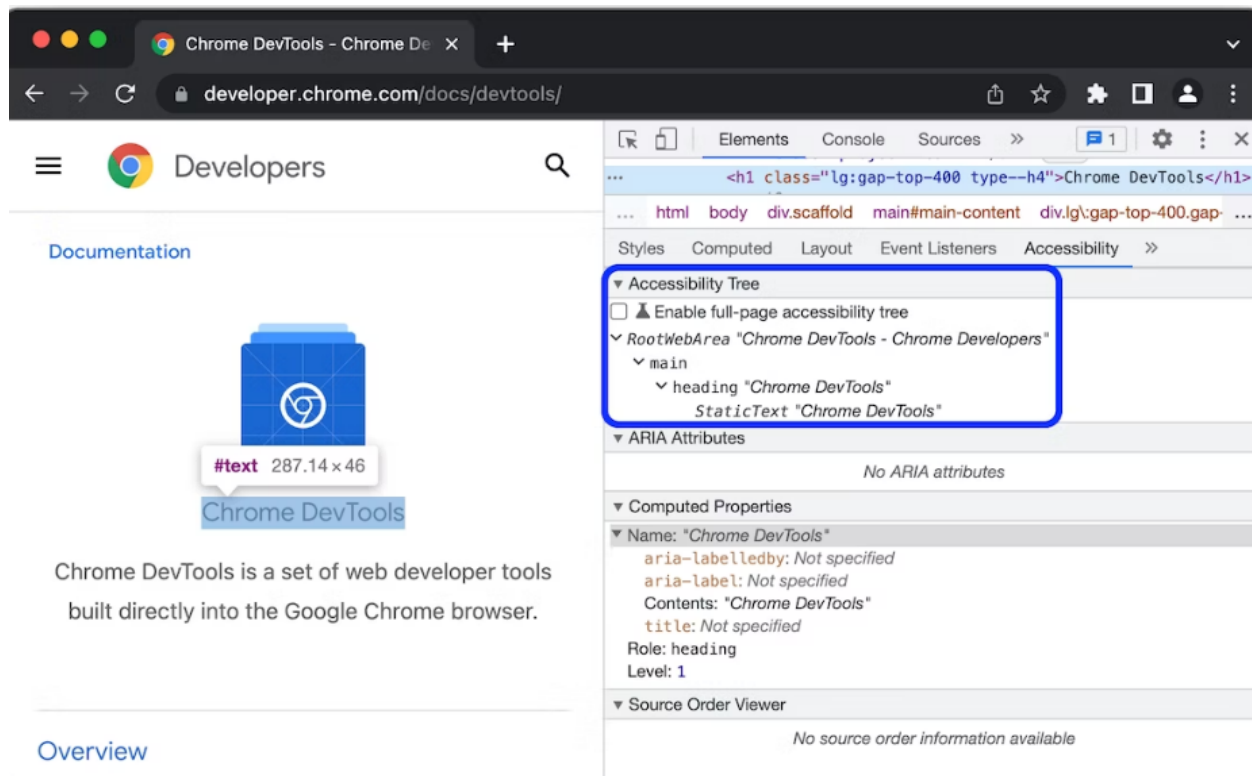
To make accessibility tree browsers take info from tegs elements and nodes. To describe roles and provide browser information about scematic and conten of your application your should use **semantic/landmarks** tags like **main, nav, section, acticle, button, side...**

For some tags, basicly for interactive tags, wich havent got description, you should add this description by usings alt attribute for images, type, placeholder, name and other attributes (**necessary code**).

Your custom interactive elements (<div>Button</div>) havent got semantic information. You should add necessary desctiptions attribute that describe name, state and value of element, for instance: role, arria-expanded (see WIA ARIA) (**extracode**).

Crome devtools Accessibility

At chrome browser devtoolse exist an accessibility tab wich can help you to check you application accessibility.



Crome devtools LightHouse

LightHouse is a crome devtool that allow you to improve your performace, quality, accessibility at your website. This tool can check some accessibility features at your website and show result.

Skip to Content | Change Text Size or Colors | All Translations

W3C® Web Accessibility Initiative WAI

Strategies, standards, resources to make the Web accessible to people with disabilities

Get Involved | About W3C WAI

Search

Accessibility Fundamentals

Planning & Policies

Design & Develop

Test & Evaluate

Teach & Advocate

Standards/Guidelines

[Home](#) / [Standards/Guidelines](#) / [WAI-ARIA](#)

Standards/Guidelines

WAI-ARIA Overview

13:49:59 - www.w3.org

https://www.w3.org/WAI/standards-guidelines/aria/

100

Performance

100

Accessibility

100

Best Practices

100

SEO

PWA

PWA

100

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. [See calculator.](#)

W3C WAI-ARIA Overview

Accessibility tree

3

WIA-ARIA

WIA-ARIA is an **WEB INTERFACE ACCESSIBILITY-Accessible Rich Internet Application** web-standards which define ways to make web application more accessible for people with disabilities. It helps with dynamic content and custom interactive user interfaces.

WIA-ARIA provide developers a set of new HTML attributes which add additional semantics and make elements more accessible.

Three main things in specification of WIA-ARIA:

1. Roles - define special attribute role to identify role, group, structure of element (button, article, tab, menu...)
2. Properties - define attributes to manage elements (aria-required, aria-describedby, aria-controls...)
3. State - define attributes to show element condition (aria-disabled, aria-checked, aria-hidden...)

Roles contain:

- roles to identify widget role (button, radio, tab...)
- composite roles to compose element in group (radiogroup, menu, tree...)
- structure roles to define structure of element (article, heading, img...)
- landmarks (navigation, banner, main...)

*for landmarks try to use HTML5 semantic landmarks tags. They have implemented landmarks roles