



# Mastering Keyboard Accessibility

by Tanja Ulianova



# Tanja Ulianova

Fullstack Web Developer @ inovex GmbH  
based in Karlsruhe, Germany  
all things web, a11y nerd, clean architecture

 @techtanja

 tulianova@inovex.de

 @inovexgmbh

 @inovexlife

**What is a11y?**

a11y = a{ccessibilit}y

**11** letters



making web content usable for any people  
(regardless of their abilities)

part of universal design and good usability

# Users!



low vision, blindness, color blindness  
*Screen Reader*

contrast, text alternatives,  
people who can't see a mouse cursor  
rely on screen reader



low hearing, deafness

somebody who cannot hear (well)  
needs captions



motor impairments  
*people who can't operate a mouse*

keyboard access



cognitive impairments, dyslexia,  
migraines

simple language, uncomplicated GUIs,  
reader mode



seizure disorders

avoid flashing animations

# "us" vs. "them"

natural • illness • situational • aging

**not needing a11y is only temporary**

Guidelines

**Abstract****Status of This Document****Introduction**

- .1 Background on WCAG 2
- .2 WCAG 2 Layers of Guidance
- .3 WCAG 2.2 Supporting Documents
- .4 Requirements for WCAG 2.2
- .5 Comparison with WCAG 2.1
  - .5.1 New Features in WCAG 2.2
  - .5.2 Numbering in WCAG 2.2
  - .5.3 Conformance to WCAG 2.2
- .6 Later Versions of Accessibility Guidelines

**1. Perceivable**

- 1.1 Text Alternatives
  - 1.1.1 Non-text Content
- 1.2 Time-based Media
  - 1.2.1 Audio-only and Video-only (Prerecorded)
  - 1.2.2 Captions (Prerecorded)
  - 1.2.3 Audio Description or Media Alternative (Prerecorded)
  - 1.2.4 Captions (Live)
  - 1.2.5 Audio Description (Prerecorded)
  - 1.2.6 Sign Language (Prerecorded)
  - 1.2.7 Extended Audio Description (Prerecorded)
  - 1.2.8 Media Alternative (Prerecorded)
  - 1.2.9 Audio-only (Live)
- 1.3 Adaptable
  - 1.3.1 Info and Relationships
  - 1.3.2 Meaningful Sequence
  - 1.3.3 Sensory Characteristics
  - 1.3.4 Orientation
  - 1.3.5 Identify Input Purpose

# Web Content Accessibility Guidelines (WCAG) 2.2

W3C Working Draft 21 May 2021

**This version:**

<https://www.w3.org/TR/2021/WD-WCAG22-20210521/>

**Latest published version:**

<https://www.w3.org/TR/WCAG22/>

**Latest editor's draft:**

<https://w3c.github.io/wcag/guidelines/22/>

**Previous version:**

<https://www.w3.org/TR/2021/WD-WCAG22-20210513/>

**Latest Recommendation:**

<https://www.w3.org/TR/WCAG/>

**Editors:**

[Chuck Adams](#) (Oracle)  
[Alastair Campbell](#) (Nomensa)  
[Rachael Montgomery](#) (Invited Expert)  
[Michael Cooper](#) (W3C)  
[Andrew Kirkpatrick](#) (Adobe)

**Participate:**

[GitHub w3c/wcag](#)  
[File a bug](#)  
[Commit history](#)  
[Pull requests](#)

Copyright © 2020-2021 W3C® (MIT, ERCIM, Keio, Beihang). W3C liability, trademark and document use rules apply.

## Abstract

Web Content Accessibility Guidelines (WCAG) 2.2 covers a wide range of recommendations for making Web content more accessible. Following these guidelines will make content more accessible to a wider range of people with disabilities, including accommodations for blindness and low vision, deafness and hearing loss,

# WCAG 2.2

**13 Guidelines** based on four principles:

*perceivable, operable, understandable, robust*

# WCAG 2.2 - Success Criteria

Testable [Success Criteria](#) per Guideline

**A** basic a11y features

**AA** worldwide legal standard ([list of laws](#))

**AAA** extended a11y features

<https://www.w3.org/TR/WCAG22/>

# How to Meet WCAG (Quick Reference)

A customizable quick reference to Web Content Accessibility Guidelines (WCAG) 2 requirements (success criteria) and techniques.

[Show About & How to Use](#)

Contents

Filter

Hide

## 1. Perceivable

### 1.1 Text Alternatives

#### 1.1.1 Non-text Content

### 1.2 Time-based Media

#### 1.2.1 Audio-only and Video-only (Prerecorded)

#### 1.2.2 Captions (Prerecorded)

#### 1.2.3 Audio Description or Media Alternative (Prerecorded)

#### 1.2.4 Captions (Live)

#### 1.2.5 Audio Description (Prerecorded)

#### 1.2.6 Sign Language (Prerecorded)

#### 1.2.7 Extended Audio Description (Prerecorded)

#### 1.2.8 Media Alternative (Prerecorded)

#### 1.2.9 Audio-only (Live)

### 1.3 Adaptable

#### 1.3.1 Info and Relationships

#### 1.3.2 Meaningful Sequence

#### 1.3.3 Sensory Characteristics

#### 1.3.4 Orientation

#### 1.3.5 Identify Input Purpose

#### 1.3.6 Identify Purpose

Selected Filters: **WCAG 2.1**: all success criteria and all techniques.

Clear filters

Expand all sections

Share

## Principle 1 – Perceivable

Information and user interface components must be presentable to users in ways they can perceive.

### Guideline 1.1 – Text Alternatives

Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language.

#### 1.1.1 Non-text Content — Level A

All non-text content that is presented to the user has a text alternative that serves the equivalent purpose, except for the situations listed below. [Show full description](#)

Understanding 1.1.1

[Show techniques and failures for 1.1.1](#)

SHARE | BACK TO TOP

### Guideline 1.2 – Time-based Media

Provide alternatives for time-based media.

#### 1.2.1 Audio-only and Video-only (Prerecorded) — Level A

For prerecorded audio-only and prerecorded video-only media, the following are true, except when the audio or video is a media alternative for text and is clearly labeled as such: [Show full description](#)

Understanding 1.2.1

How to Meet WCAG: <https://www.w3.org/WAI/WCAG21/quickref/>

**Keyboard Support**

# Why Keyboard?

- People who can't use a mouse or trackpad
- Other assistive technology
- Convenience or preference — [r/MechanicalKeyboards](https://www.reddit.com/r/MechanicalKeyboards)



**Abstract****Status of This Document****Introduction**

- .1 Background on WCAG 2
- .2 WCAG 2 Layers of Guidance
- .3 WCAG 2.2 Supporting Documents
- .4 Requirements for WCAG 2.2
- .5 Comparison with WCAG 2.1
  - .5.1 New Features in WCAG 2.2
  - .5.2 Numbering in WCAG 2.2
  - .5.3 Conformance to WCAG 2.2
- .6 Later Versions of Accessibility Guidelines

**1. Perceivable**

- 1.1 Text Alternatives
  - 1.1.1 Non-text Content
- 1.2 Time-based Media
  - 1.2.1 Audio-only and Video-only (Prerecorded)
  - 1.2.2 Captions (Prerecorded)
  - 1.2.3 Audio Description or Media Alternative (Prerecorded)
  - 1.2.4 Captions (Live)
  - 1.2.5 Audio Description (Prerecorded)
  - 1.2.6 Sign Language (Prerecorded)
  - 1.2.7 Extended Audio Description (Prerecorded)
  - 1.2.8 Media Alternative (Prerecorded)
  - 1.2.9 Audio-only (Live)
- 1.3 Adaptable
  - 1.3.1 Info and Relationships
  - 1.3.2 Meaningful Sequence
  - 1.3.3 Sensory Characteristics
  - 1.3.4 Orientation
  - 1.3.5 Identify Input Purpose

# Web Content Accessibility Guidelines (WCAG) 2.2

W3C Working Draft 21 May 2021

**This version:**

<https://www.w3.org/TR/2021/WD-WCAG22-20210521/>

**Latest published version:**

<https://www.w3.org/TR/WCAG22/>

**Latest editor's draft:**

<https://w3c.github.io/wcag/guidelines/22/>

**Previous version:**

<https://www.w3.org/TR/2021/WD-WCAG22-20210513/>

**Latest Recommendation:**

<https://www.w3.org/TR/WCAG/>

**Editors:**

[Chuck Adams](#) (Oracle)  
[Alastair Campbell](#) (Nomensa)  
[Rachael Montgomery](#) (Invited Expert)  
[Michael Cooper](#) (W3C)  
[Andrew Kirkpatrick](#) (Adobe)

**Participate:**

[GitHub w3c/wcag](#)  
[File a bug](#)  
[Commit history](#)  
[Pull requests](#)

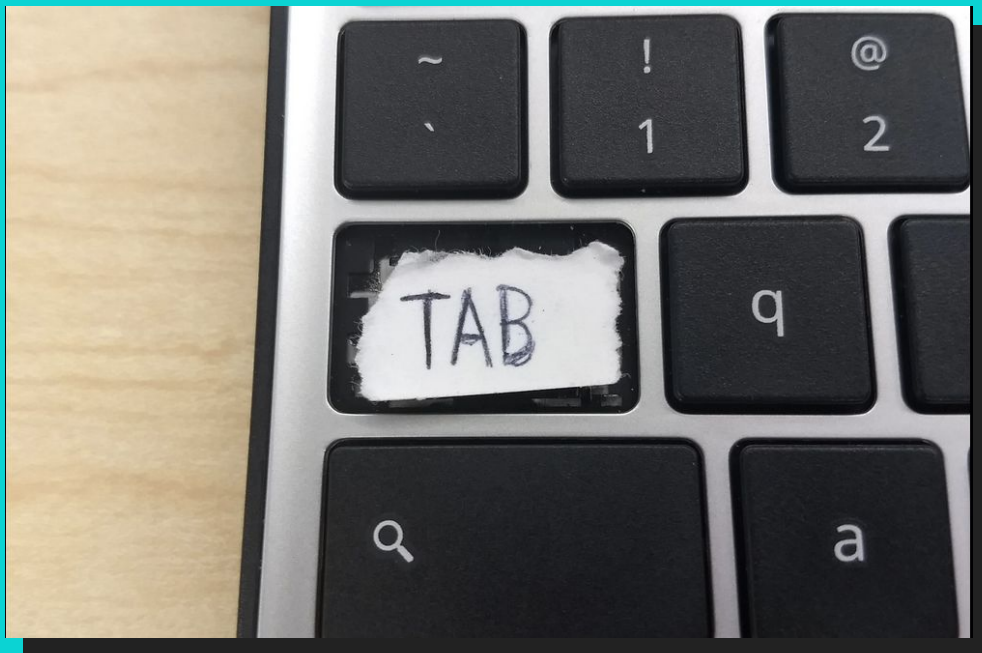
Copyright © 2020-2021 W3C® (MIT, ERCIM, Keio, Beihang). W3C liability, trademark and document use rules apply.

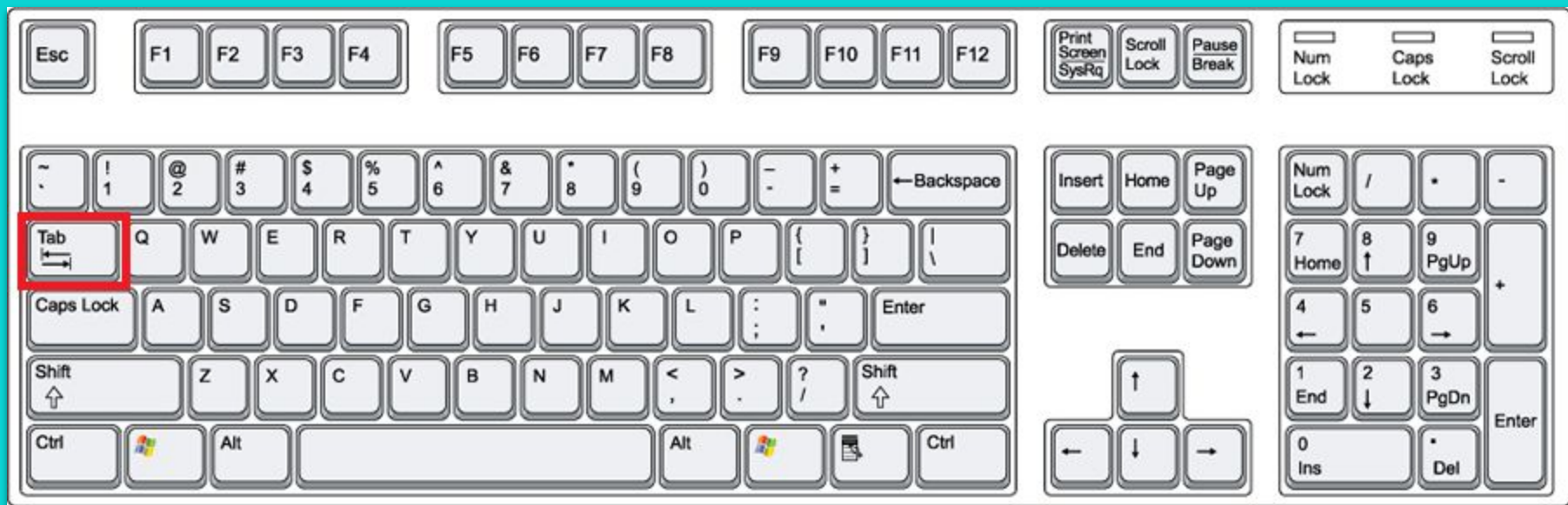
## Abstract

Web Content Accessibility Guidelines (WCAG) 2.2 covers a wide range of recommendations for making Web content more accessible. Following these guidelines will make content more accessible to a wider range of people with disabilities, including accommodations for blindness and low vision, deafness and hearing loss,

## Guideline 2.1 – Keyboard Accessible

*"Make all functionality available from a keyboard."*





*Demo Time*

**Keyboard accessibility = *focus  
management***

# ***Tabbing order***

In HTML, the default tabbing (=focus) order follows the order in which elements appear in the content source.

The default behaviour can be modified with the ***tabindex*** attribute.

# tabindex

## **tabindex="0"**

Make an otherwise unfocusable element focusable.

Use its position in DOM for focus order.

## **tabindex="-1"**

Exclude an otherwise focusable element from focus order.

Make an unfocusable element, focusable via `element.focus()`

### **Avoid tabindex 1+**

Creates a page global focus order (all >0 in their order, before all with 0).  
Very hard to manage and bad for non vision impaired keyboard users.

# ***Tab order should match logical reading order.***

Don't destroy it by using floating style

Place **position:absolute** elements in their logical place in DOM

***Put the element in its logical place  
in the DOM.***

***Want to ignore input events on  
element all together?***

# inert Attribute

..makes the browser "ignore" user input events for the element, including focus events..

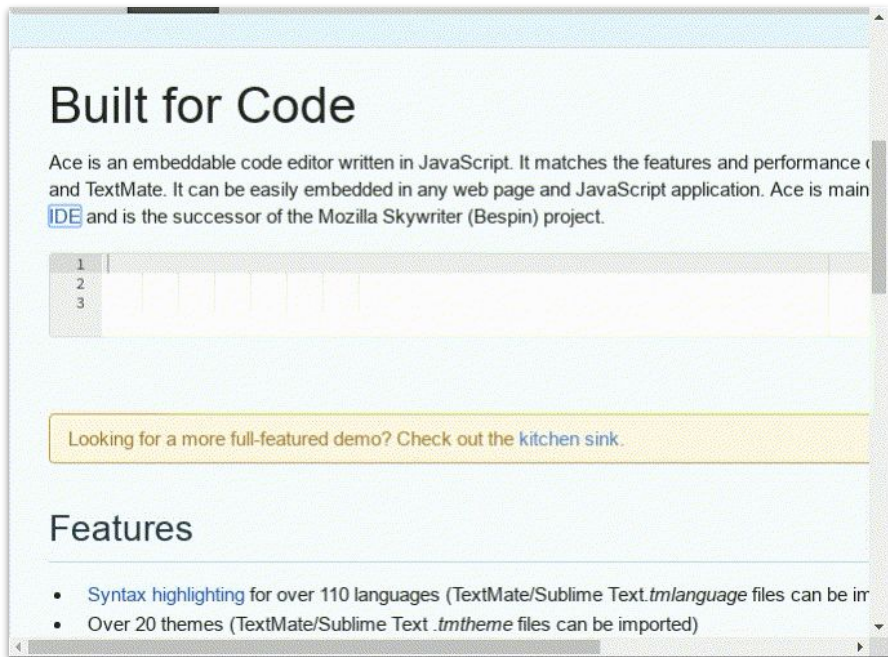
<https://developer.mozilla.org/en-US/docs/Web/API/HTMLElement/inert>

	🖥️					📱					
	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android
inert	✓ 102	✓ 102	✗ No	✓ 88	✓ 15.5	✓ 102	✗ No	✓ 70	✓ 15.5	✓ 19.0	✓ 102

Spec: <https://html.spec.whatwg.org/multipage/interaction.html#the-inert-attribute>

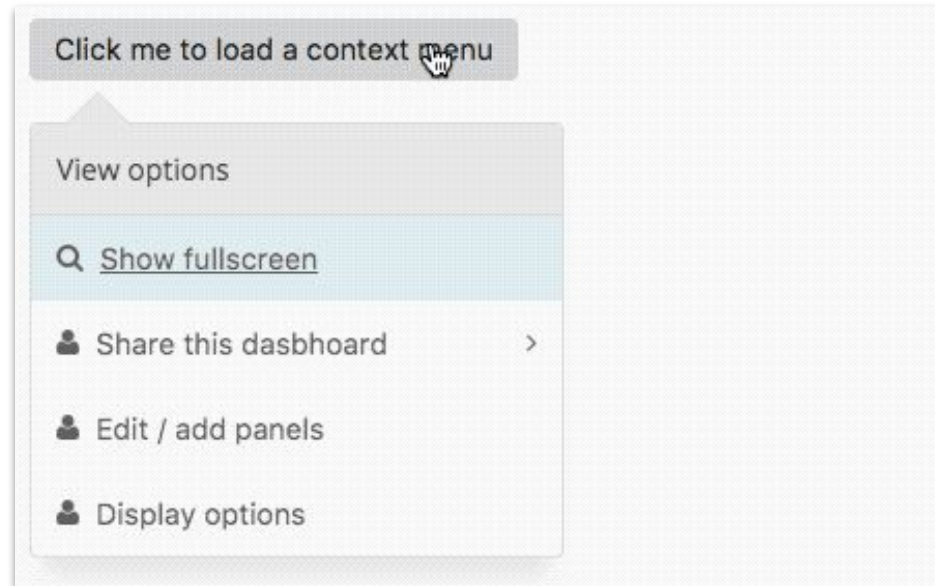
# Don't trap focus

User should be able to navigate within the page without being trapped (e.g. inside a code editor).



## ...unless you need to trap focus

In modal elements, e.g. dialogs, contextmenus, usually everything, that closes as soon as you click outside of it.



**Focus *should always be visible***

Make  focus ring visible!

(no `outline: none`)

mouse, touch,  
keyboard

```
el:focus {  
  outline: 2px solid black;  
}
```

keyboard only

```
el:focus-visible {  
  outline: 4px dashed blue;  
}
```

<https://codepen.io/techtanja/pen/vLEaeZm>

# :focus-visible

	🖥️					📱					
	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android
<code>:focus-visible</code>	< 86	< 86	< 85 ::	< 72	< 15.4	< 86	< 85 ::	< 61	< 15.4	< 14.0	< 86

Sign me up

```
<div class="button" onclick="..">  
  Sign me up  
</div>
```

## Please don't do this.

- No focus via tab
  - **tabindex="0"**
- No keyboard events
  - *Trigger on Enter keydown event*
- Not announced as a button
  - **role="button"**

*instead*

Sign me up

```
<button class="button">  
  Sign me up  
</button>
```

Much better! 🙌

Extensive keyboard support for free:

- Events
- Focus

Style as you wish

Sign me up

**<button>**    **actions**

**<a>**    **navigation & links**

**P.S.**

**<div>**

**is a container!**

**<span>**

**is an *inline* container!**

**80%** of the a11y issues can be fixed with ***semantic HTML***

Recap

# Keyboard a11y Checklist

- ❑ Semantic HTML
- ❑ Focus handling
  - ❑ tab order is logical
  - ❑ DOM = structure
  - ❑ **tabindex** with caution
  - ❑ **inert** attribute
  - ❑ focus traps only if explicitly needed
- ❑ Visible focus handling
  - ❑ Avoid **outline: none**
  - ❑ **:focus-visible** for only keyboard focus
- ❑ [Test your app!](#)

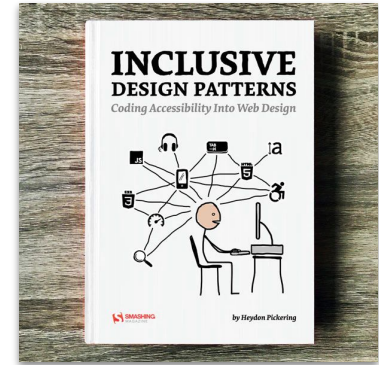
# Further Resources



<https://inclusive-components.design/>  
by Heydon Pickering



[a11y Casts](#)  
by Rob Dodson



[Inclusive Design Patterns](#)  
by Heydon Pickering

**“For most people technology makes things easier. For people with disabilities, technology makes things possible.”**

Mary Pat Radabaugh, Director of IBM National Support Center for Persons with Disabilities

# Thank you for your Attention!

Tanja Ulianova



@techtanja



tulianova@inovex.de



@inovexgmbh



@inovexlife



inovex

