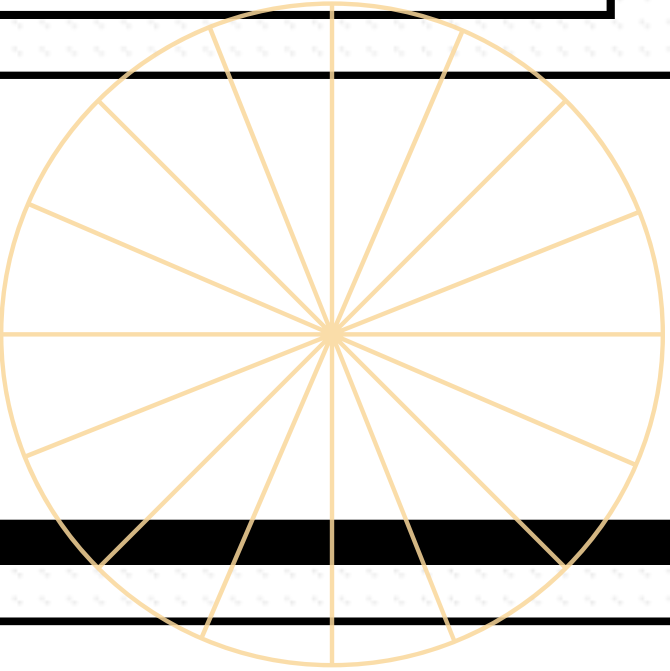




Intro React Accessibility 🔍

Essentials  
For Developers



How to make website accessible for everyone

Start



# Raihan Nismara

Frontend Dev @ Telkom Indonesia

 [raihan.my.id](https://raihan.my.id)

 [raihan.nismara](https://www.instagram.com/raihan.nismara)

 [codevcast](https://www.youtube.com/codevcast)

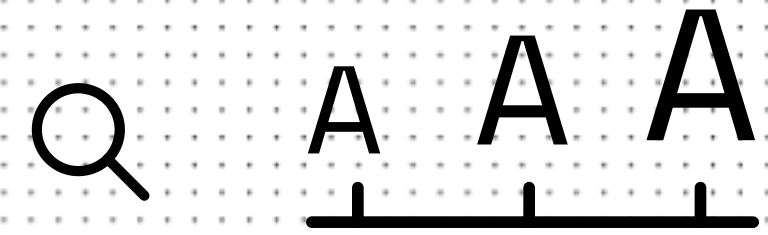
 [Raihan Nismara](https://www.linkedin.com/in/RaihanNismara)

# Today's Agenda:

*To understand what accessibility is, and why is it important to consider?*

1. What is Accessibility (a11y)
2. How users navigate a website
3. Accessibility Principles
4. What are the benefits
5. How to measure/audit the web for accessibility
6. Unit testing



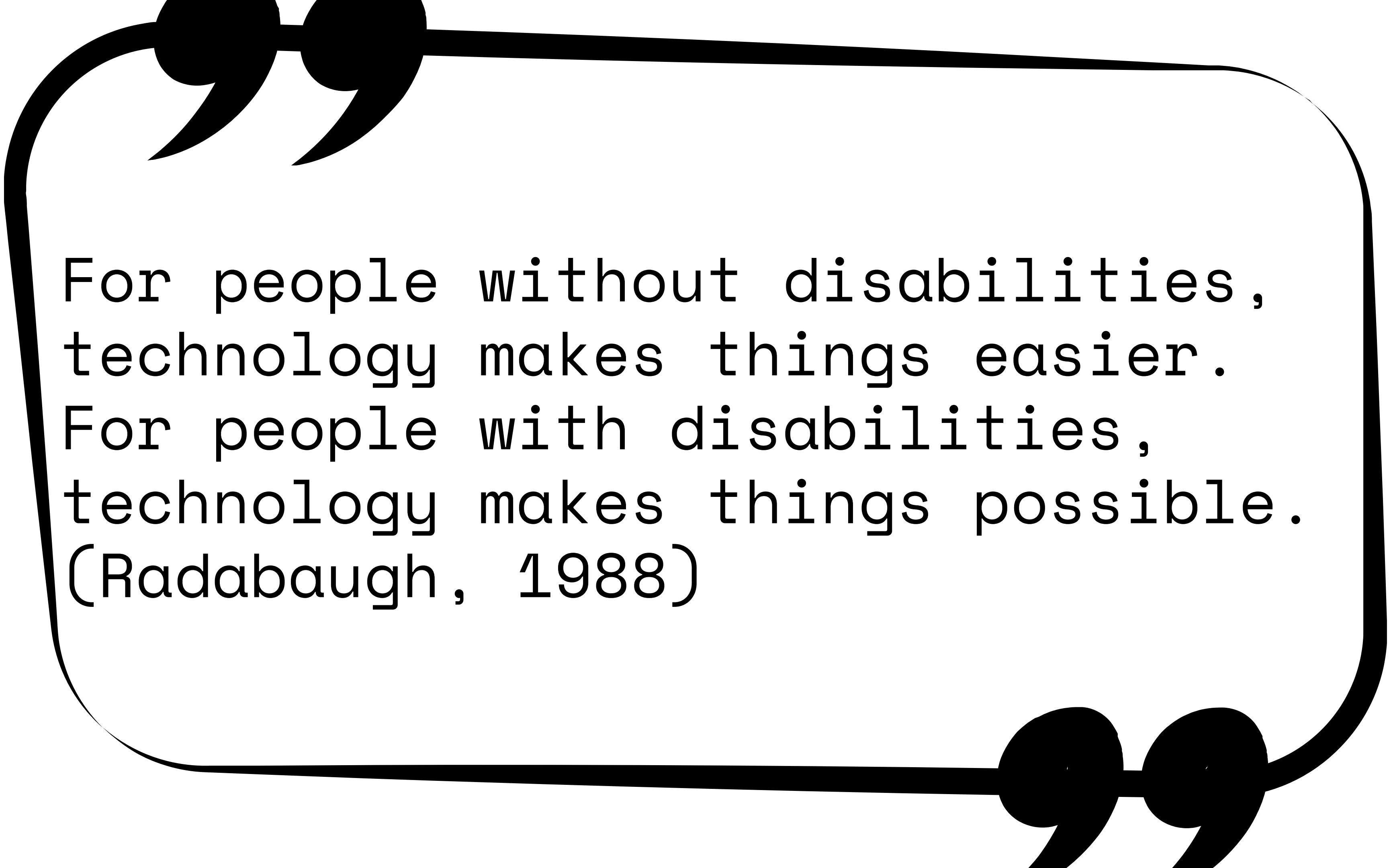


What is Web Accessibility (a11y)

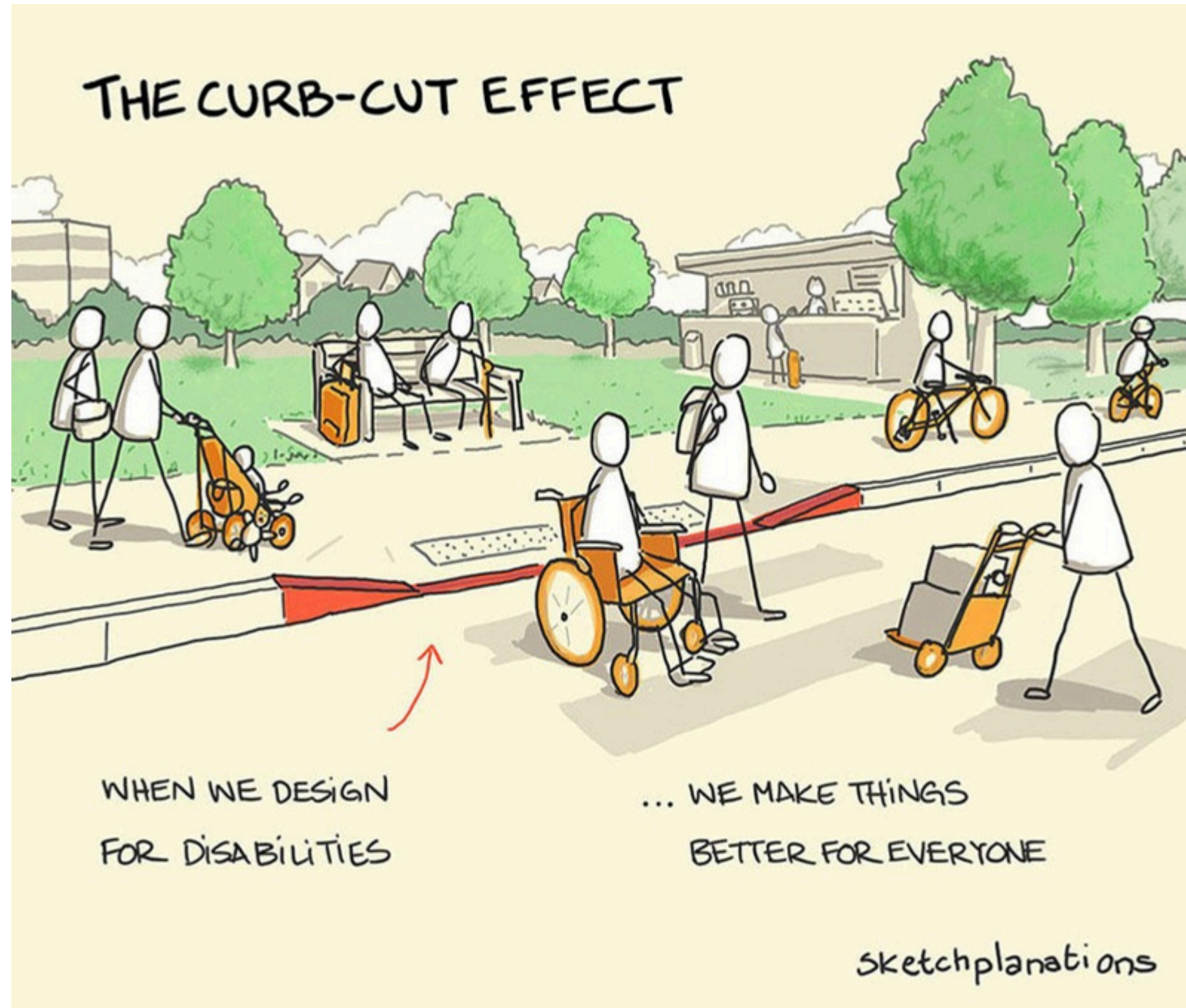


Web accessibility means that websites, tools, and technologies are designed and developed so that people with disabilities can use them. More specifically, people can: perceive, understand, navigate, and interact with the Web. contribute to the Web.





For people without disabilities,  
technology makes things easier.  
For people with disabilities,  
technology makes things possible.  
(Radabaugh, 1988)

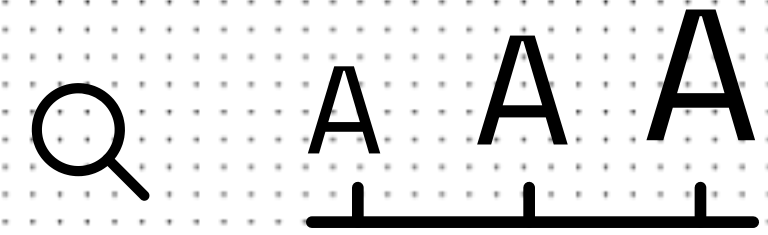


Source: <https://sketchplanations.com/the-curb-cut-effect>









How user with special needs  
navigate a website



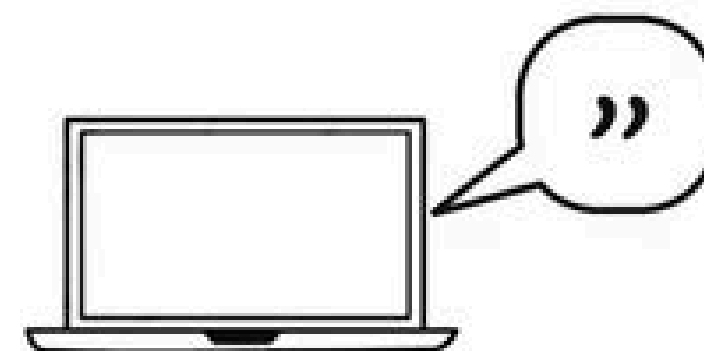
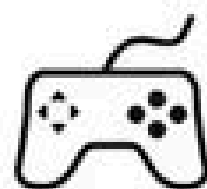
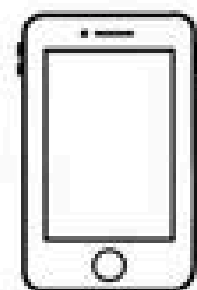
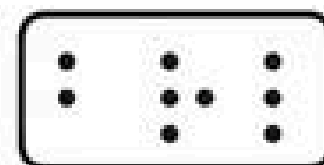
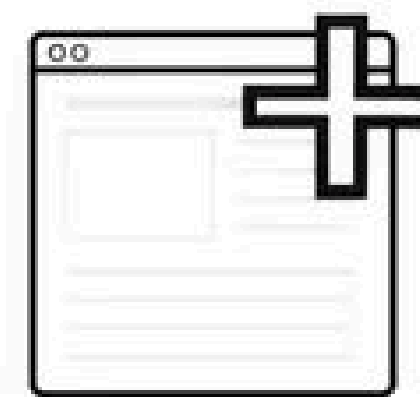
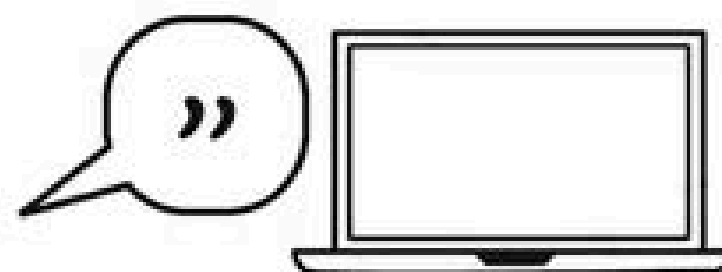
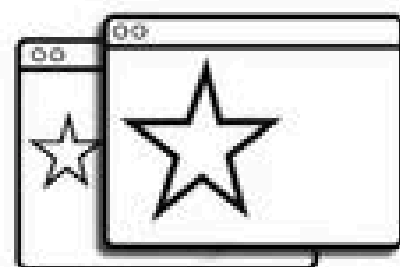
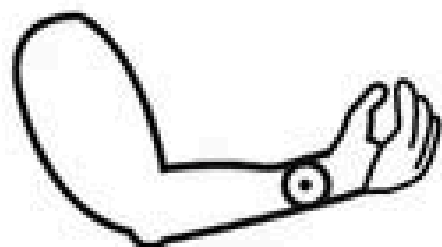
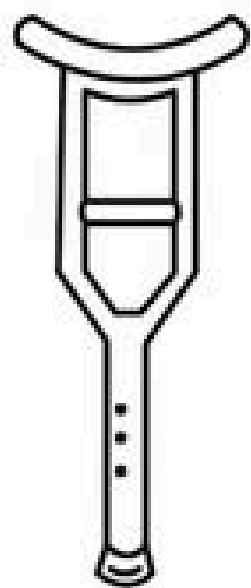
How user with special needs  
navigate a website ?

**1. Assistive Technology**

**2. Adaptive Strategy**



# Assistive technology



# How user with special needs navigate a website ?

Physical disabilities

Common problem :

1. Keyboard accessibility
2. Lack of visible focus
3. Focus order



# How user with special needs navigate a website ?

Physical & visual disabilities

Common problem :

1. Keyboard accessibility
2. Lack of visible focus
3. Focus order



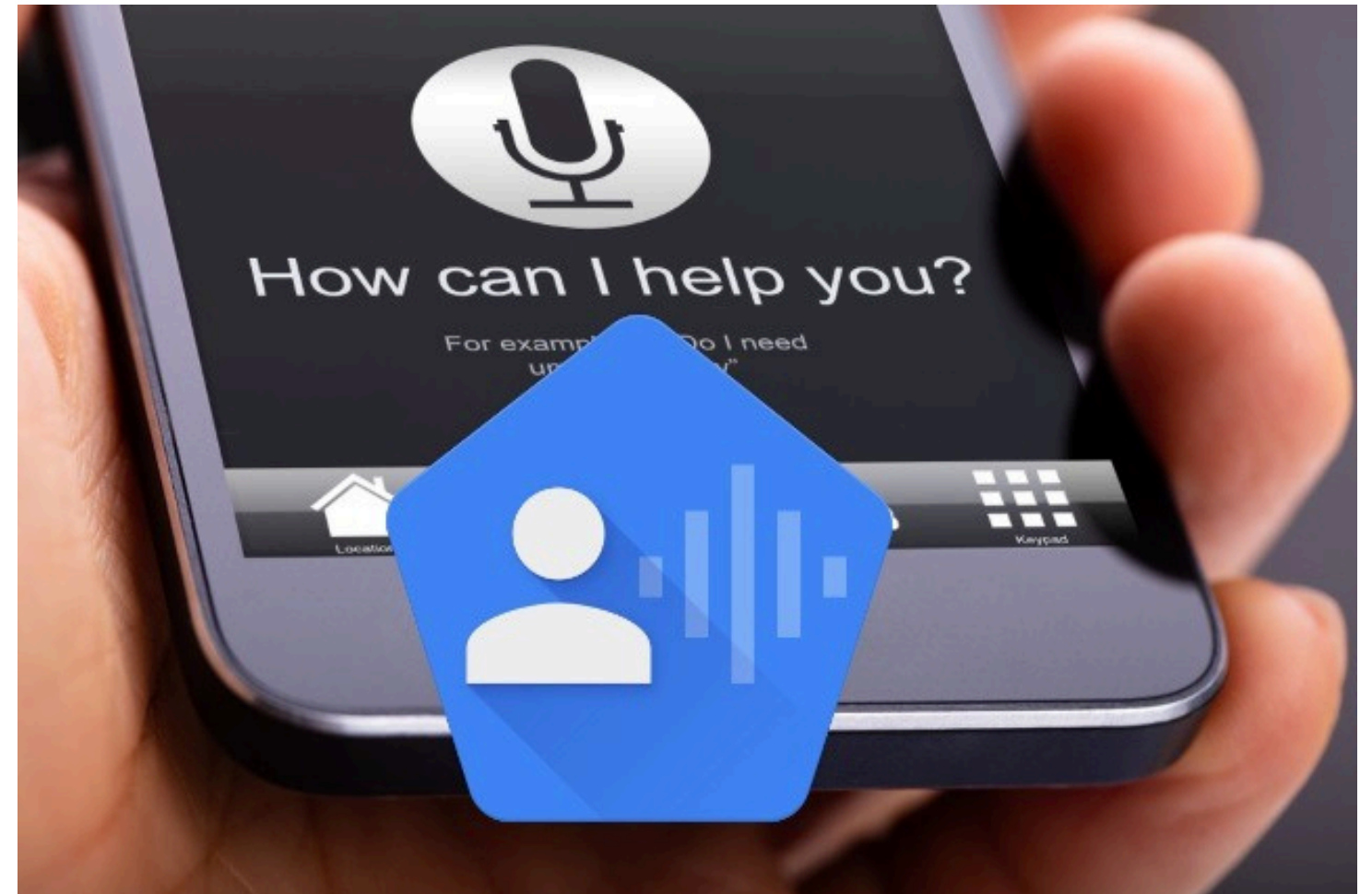
Source: <https://accessibleweb.com/assistive-technologies/assistive-technology-focus-sip-and-puff-devices/>

# How user with special needs navigate a website ?

Physical & visual disabilities

Common problem :

1. Unclear links and buttons
2. Lack of keyboard access
3. Illogical focus order
4. Poor visible focus
5. Alternatives



# How user with special needs navigate a website ?

Physical & visual disabilities(blind)

Common problem :

1. Alternatives
2. Lack of keyboard accessibility
3. Unclear Structure



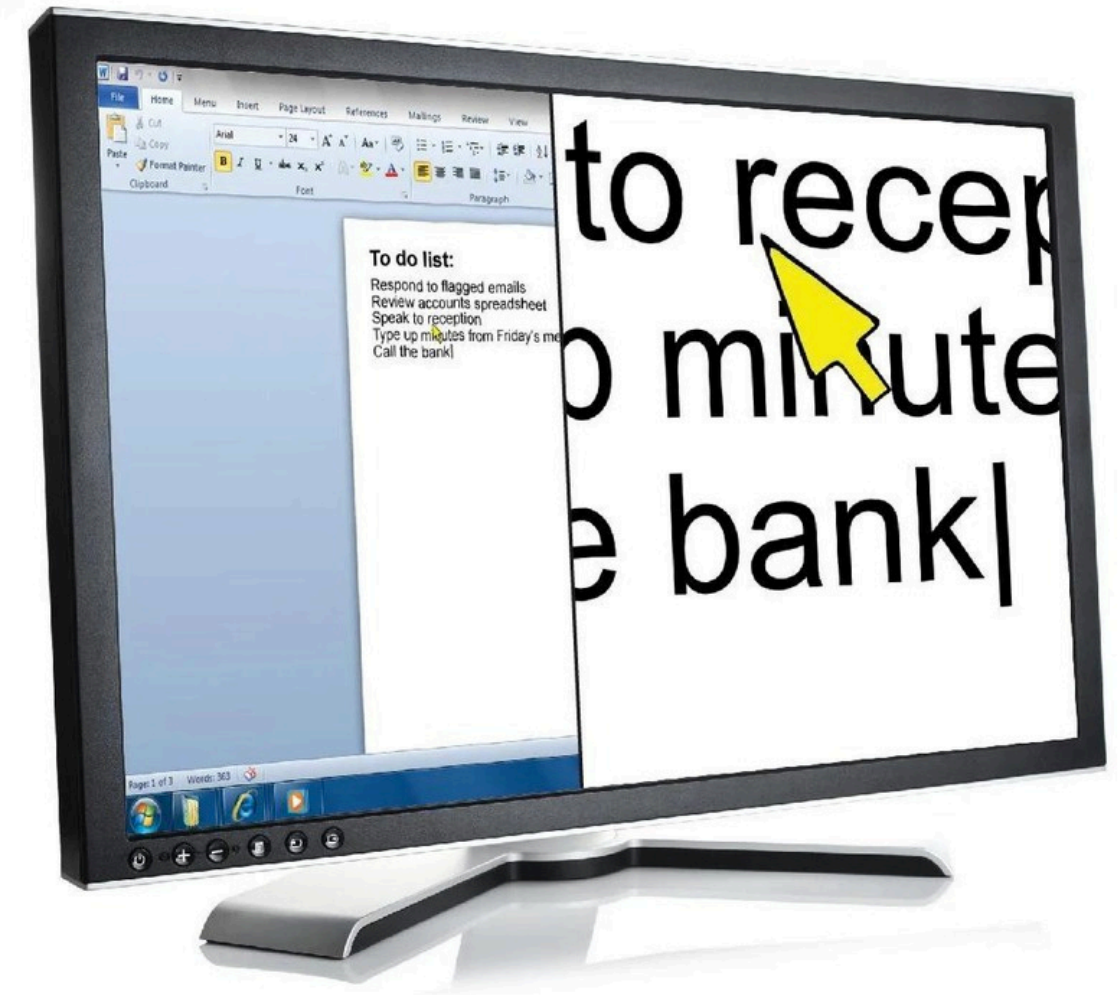


# How user with special needs navigate a website ?

Physical & visual disabilities(low vision)

Common problem :

1. Font size
2. Poor contrast
3. Relying on color
4. Animations and moving content

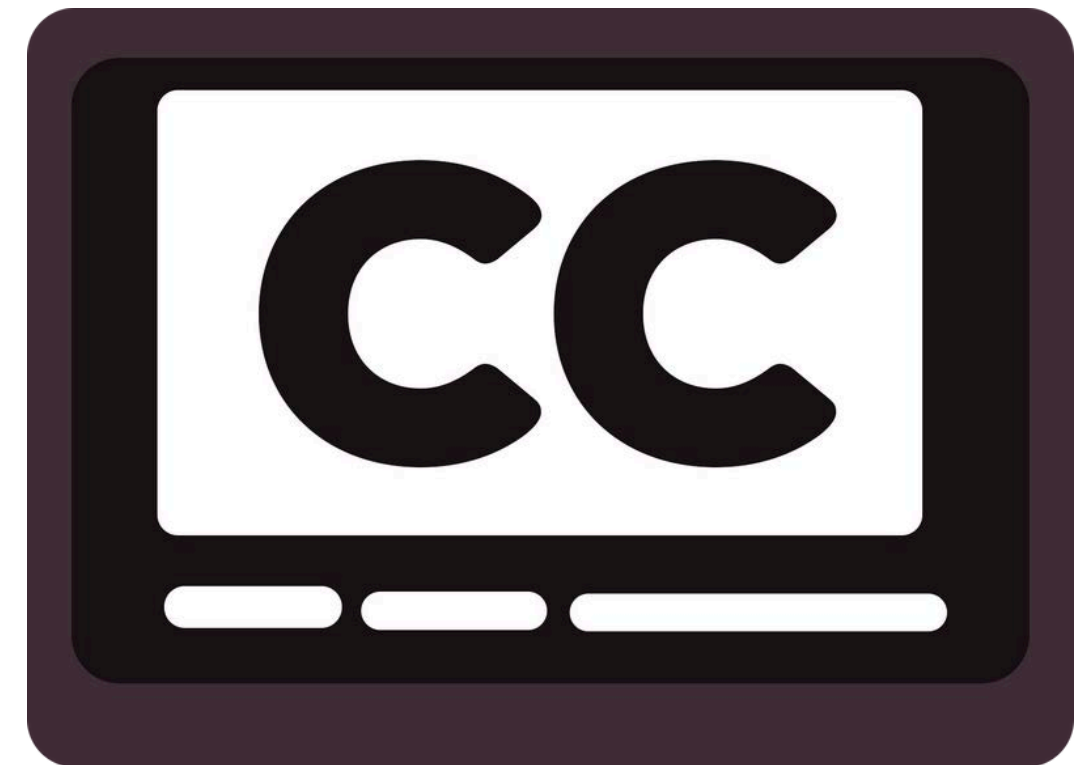


# How user with special needs navigate a website ?

Hearing disabilities

Common problem :

1. No closed captions
2. Subtitles
3. No transcripts
4. No sign language



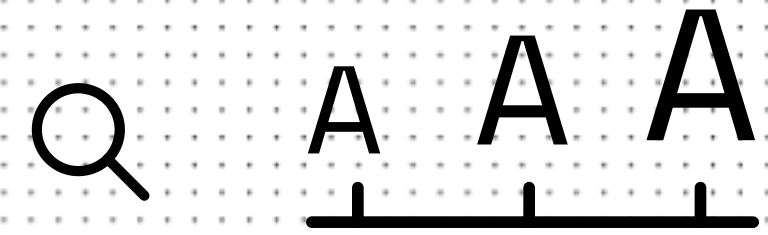
# How user with special needs navigate a website ?

Hearing, Cognitive and learning disabilities

Common problem :

- 1.No closed captions
- 2.Subtitles
- 3.No transcripts
- 4.No sign language





# Accessibility Principle



Web Content Accessibility Guidelines (WCAG)  
international standard, including WCAG 2.0, WCAG  
2.1, and WCAG 2.2.

1. Perceivable
2. Operable
3. Understandable
4. Robust



# **Perceivable**

Information can be presented in different ways. for example, in braille, different text sizes, text-to-speech, or symbols, etc.

## 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.*



```
["id":"defect-image-offensive","value":"Offensive or adult content"),  
{"id":"defect-image-extra-items","value":"Shows additional items"),  
{"id":"defect-image-not-clear","value":"Is not clear"}, {"id":"defect-  
other-image-issue","value":"Other"}]]" data-metadata="IMAGE" data-  
feature-container-id="imageBlockEDPOverlay" data-custom-event-handler=  
"imageBlockEDPCustomEventHandler" data-display-name="Images" data-edit-  
data-state="imageBlockEDPEditData" data-position="0" data-resolver=  
"CQResolver"/>/span>  
<!-- Creating a custom overlay for image not available experience -->  
<!-->  
<div class="variationUnavailable unavailableExp" style="display:  
none;"></div>  
<div class="a-hidden" id="auImmersiveView0lv"></div>  
<ul class="a-unordered-list a-nostyle a-horizontal list maintain-  
height">  
  <li><span id="imageBlockEDPOverlay"></span>  
<li class="image" item itemNo0 maintain-height  
selected" style="cursor: pointer;">  
  <span class="a-list-item">  
    <span class="a-declarative" data-action="main-image-click" data-  
main-image-click="{}">  
      <div id="imgTagWrapperId" class="imgTagWrapper" style="height:  
500px;">  
        <img alt="Doritos Tortilla Chips, Nacho Cheese, 1.75-Ounce  
Large Single Serve Bags (Pack of 64)" src="https://images-  
na.ssl-images-amazon.com/images/I/  
71Br1LeeJGL_5Y679SY_0_0_488_679_P1bundle=  
64,TopRight,0_0_5X488_5Y679_CR,0_0_488_679_S1Q0_.jpg" data-  
old-hires="https://images-na.ssl-images-amazon.com/images/I/  
71Br1LeeJGL_5L1366_.jpg" class="a-dynamic-image a-stretch-  
vertical" id="landingImage" data-a-dynamic-image="("https://  
images-na.ssl-images-amazon.com/images/I/  
71Br1LeeJGL_5Y450SX322_5Y450_CR,0_0_322_450_P1bundle=  
64,TopRight,0_0_5X322_5Y450_CR,0_0_322_450_S1Q0_.jpg":  
[450,322],"https://images-na.ssl-images-amazon.com/images/I/
```

## Time-based media

*Provide alternatives for time-based media.*

### Content that is dependent on time


- audio-only files: podcasts
- video-only files
- video plus audio: movies
- slide show carousel

### What is the problem?

- Do not hear the audio information
- Do not see the visual information

### How to?

- captions
- transcripts
- audio descriptions
- sign language

Transcript 

00:23 here's 30 thousand dollars in a duffel

00:25 bag oh oh no

00:26 can i touch it yeah you can touch it of

00:28 course you can pay my views enough

00:30 i think it's safe to say you don't need

00:31 **this site anymore right oh no not today**

00:34 next place up next i'm paying for

00:36 everyone's rent in this apartment

00:37 complex

00:38 even me you don't live here this is

00:40 jessica the property manager of the



# Adaptable content

Create content that can be presented in different ways (for example simpler layout) without losing information or structure.

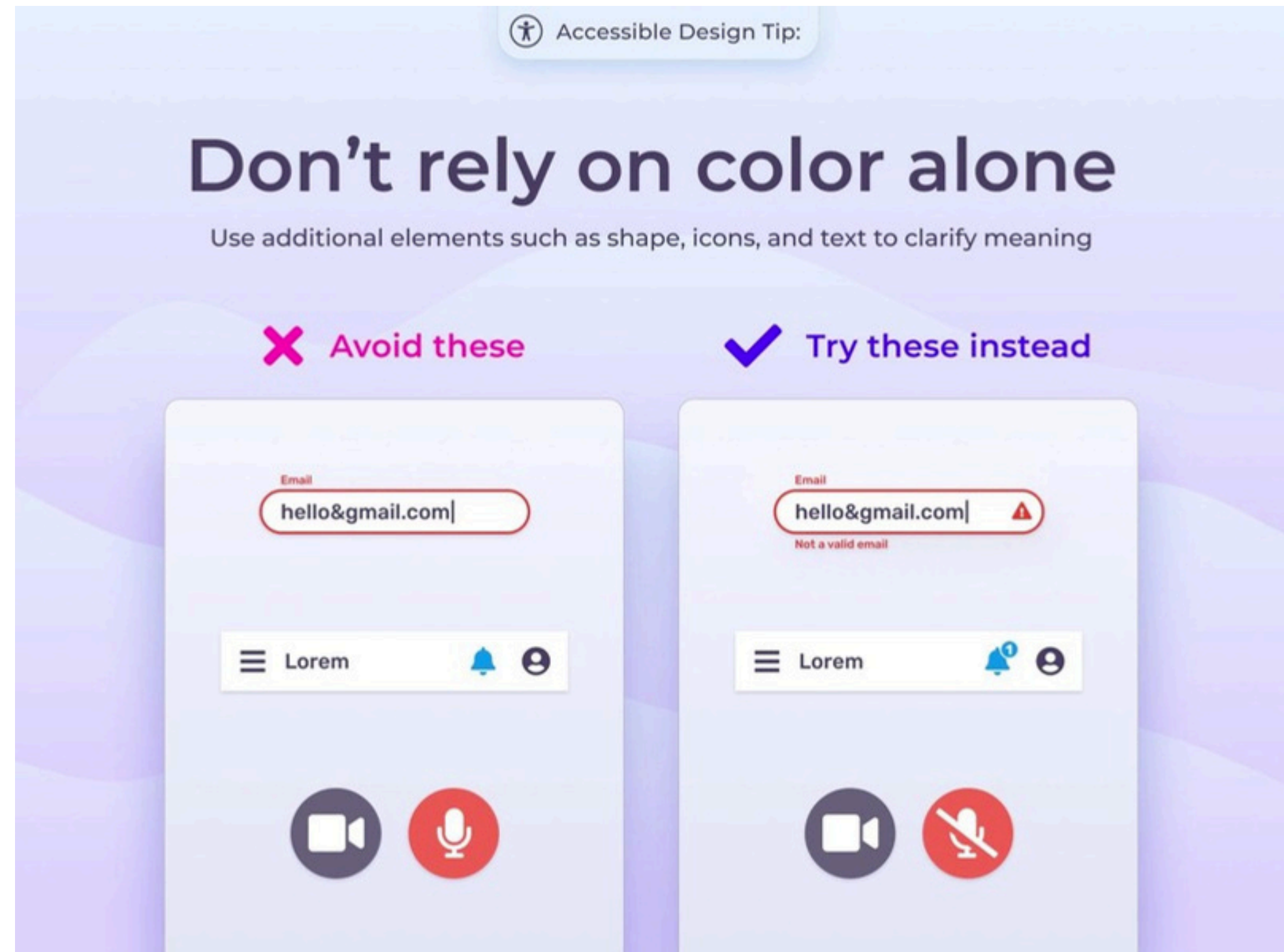


## Distinguishable content

*Make it easier for users to see and hear content including separating foreground from background.*

### Examples of making content distinguishable

- Color is not used as the only way of conveying information or identifying content
- Default foreground and background color combinations provide sufficient contrast
- When users resize text up to 200% or change text spacing, no information is lost
- Background audio is low or can be turned off, to avoid interference or distraction

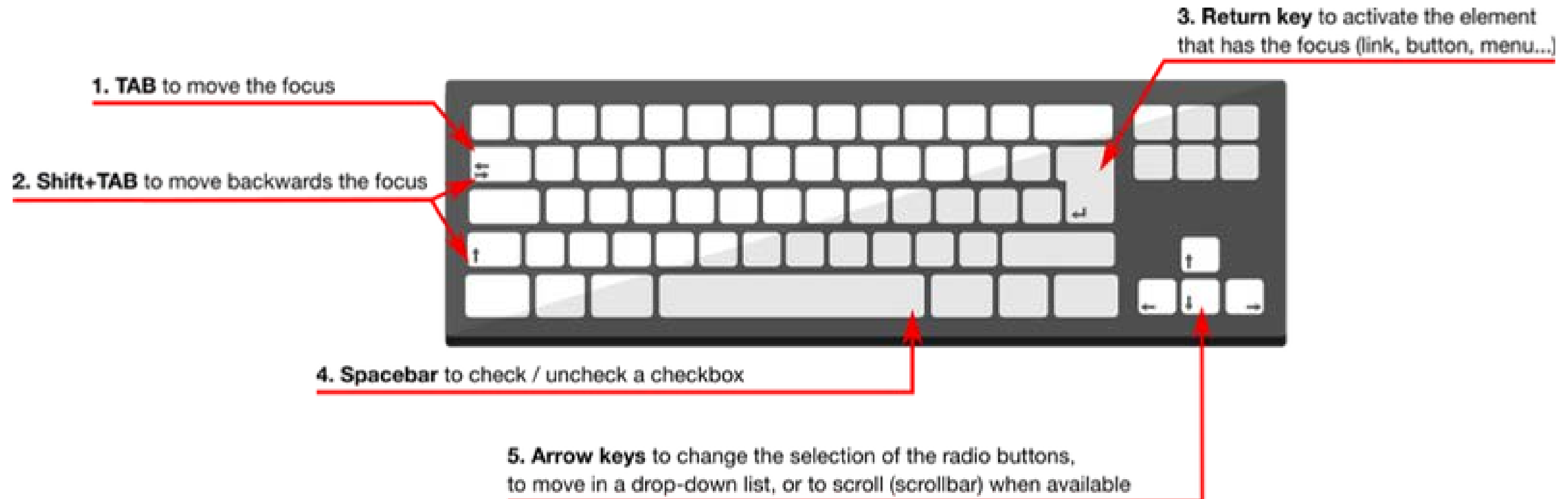


# Operable

Functionality can be used in different modalities; for example, keyboard, mouse, sip-and-puff, speech input, touch, etc

# Keyboard accessible

*Make all functionality available from a keyboard.*



## Enough time

*Provide users enough time to read and use content.*



## **Avoid seizures and physical reactions**

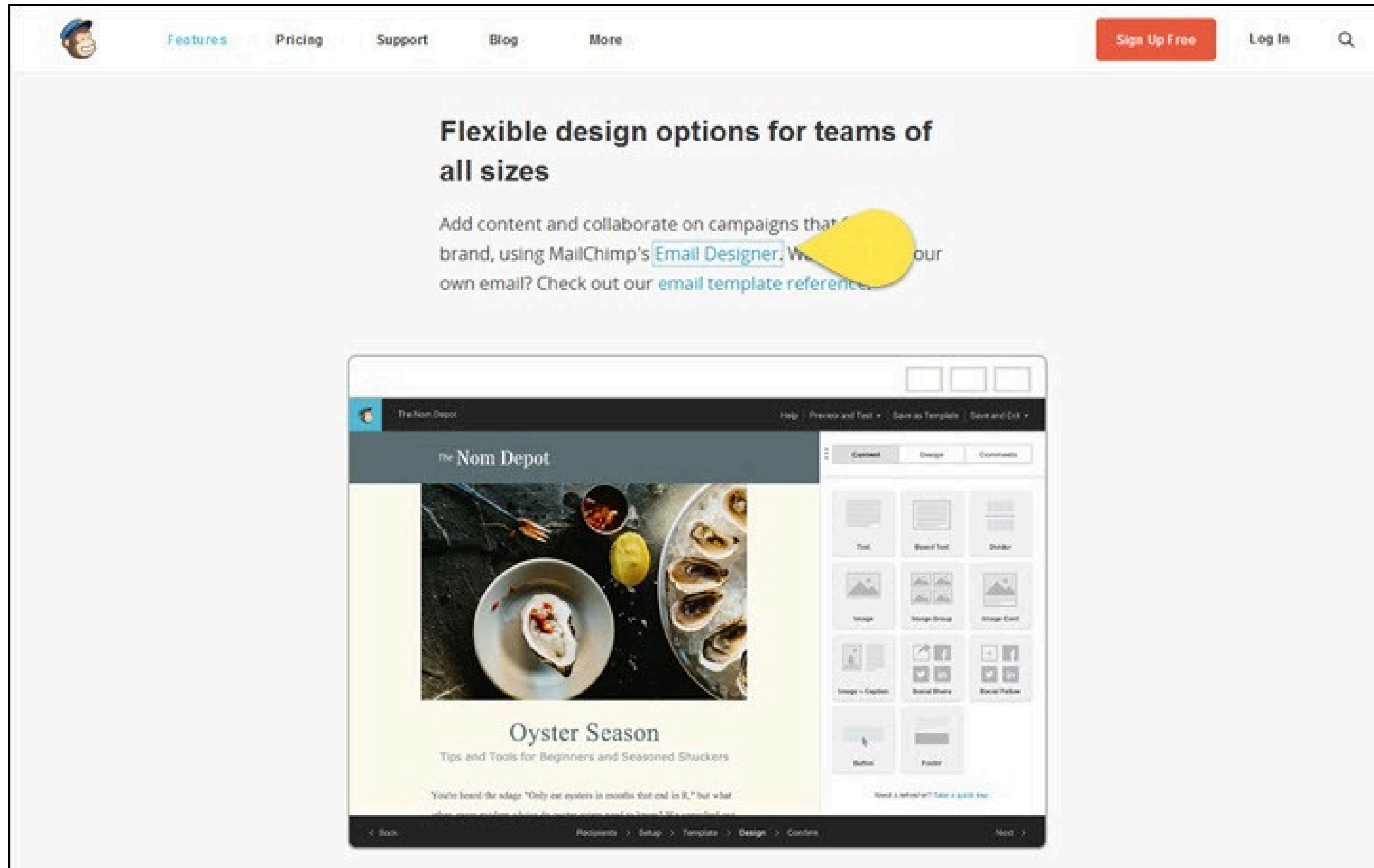
*Do not design content in a way that is known to cause seizures or physical reactions.*



Web  
Animation

# Navigable content

*Provide ways to help users navigate, find content, and determine where they are.*



## Input modalities

*Make it easier for users to operate functionality through various inputs beyond keyboard.*





# **Understandable**

Information and functionality are understandable; for example consistent navigation, simple language, etc.

## Readable content

*Make text content readable and understandable.*

- Identifying the primary language of pages and of page parts
- Explaining abbreviations, unusual words, and phrases
- Using simple language or providing simplified alternatives

### Trust your copy and let your prospects read in peace

Your website visitors **don't appreciate** your skills, and your offers leave them **indifferent**. You put **so much hard work** into your website, but it's just **not paying off**.



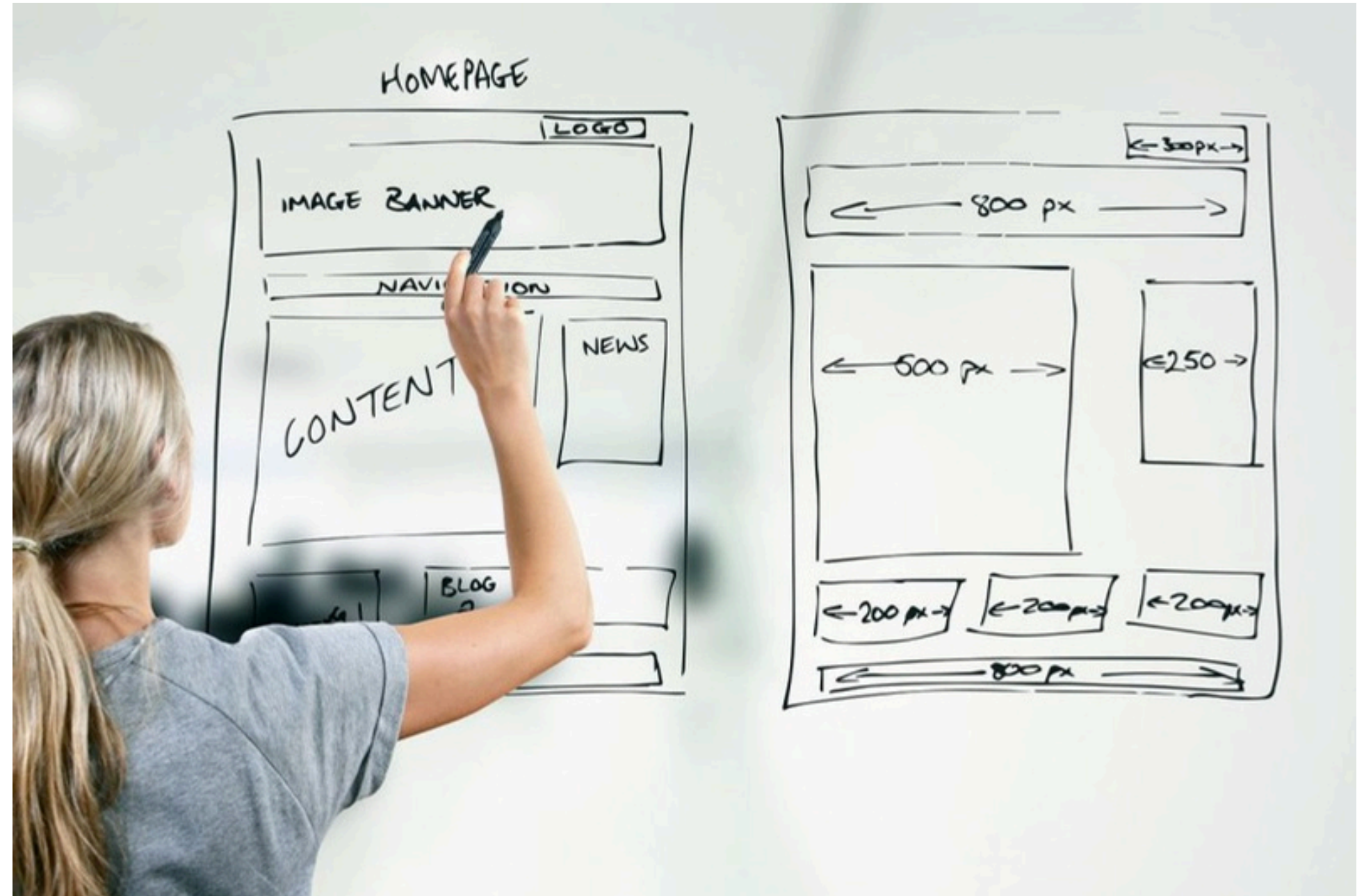
Your website visitors don't appreciate your skills, and your offers leave them indifferent. You put so much hard work into your website, but it's just not paying off.



## Predictable content

*Make Web pages appear and operate in predictable ways.*

- Keep the navigation consistent across your website
- Name buttons and controls consistently too
- Avoid changing the page simply when actionable elements receive focus or input



# Input assistance

*Help users avoid and correct mistakes.*


Getting started with Delicious is easy!  
Please start by entering your account details below.

First name	<input type="text" value="Anders"/>	✓
Last name	<input type="text" value="Toxboe"/>	✓
Email Address	<input type="text" value="info@ui-patterns"/>	✗ Please provide a valid email address.
User name	<input type="text" value="toxboe"/>	✓
Password	<input type="password"/>	✗ You must supply a password.
Retype Password	<input type="password"/>	✗ You must verify your password.

**Password Requirements**

- at least 6 characters required
- at least one letter and one symbol or number required
- may not contain any part of your username or full name

Please type the letters below  
This helps us prevent automated registrations



If you require assistance with registration our [support team](#) can help you.

I have read and agree to the [Terms of Service](#) ✓

# Robust

Content can be interpreted reliably by a wide variety of browsers, media players, and assistive technologies.

## Compatible content

*Maximize compatibility with current and future user agents, including assistive technologies.*

- Ensuring markup can be reliably interpreted, for instance by ensuring it is valid
- Providing a name, role, and value for non-standard user interface components

- Default
- Checked
- Disabled
- Disabled & Checked

[View source >>](#)

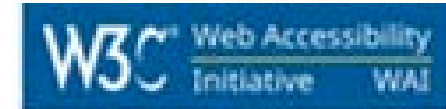
---

```
name="Default" role="checkbox" value="check"
```

# WCAG Guideline

## 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](#)



Selected Filters: **WCAG 2.0**: all success criteria and all techniques. [Clear filters](#) [Expand all sections](#) [Share](#)

### 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
- 1.4 Distinguishable
  - 1.4.1 Use of Color
  - 1.4.2 Audio Control
  - 1.4.3 Contrast (Minimum)
  - 1.4.4 Resize text
  - 1.4.5 Images of Text
  - 1.4.6 Contrast (Enhanced)
  - 1.4.7 Low or No Background Audio

### 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](#)

[Show techniques and failures for 1.2.1](#)

[Share](#) | [Back to Top](#)

#### 1.2.2 Captions (Prerecorded) — Level A

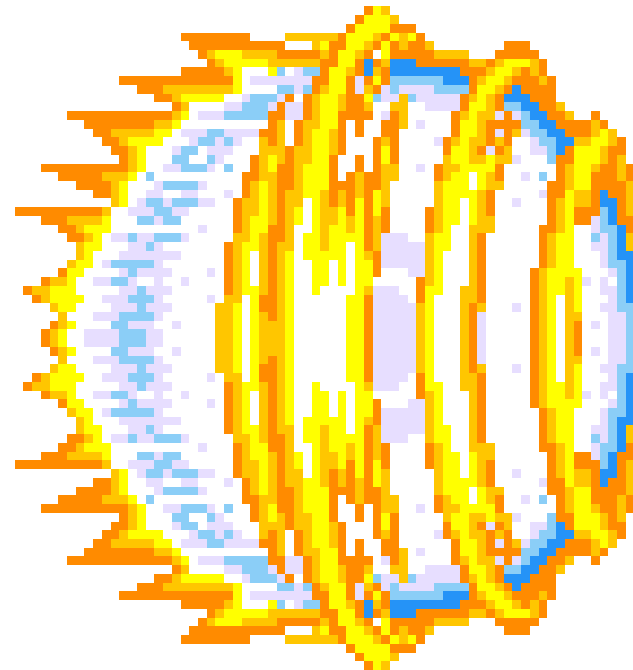
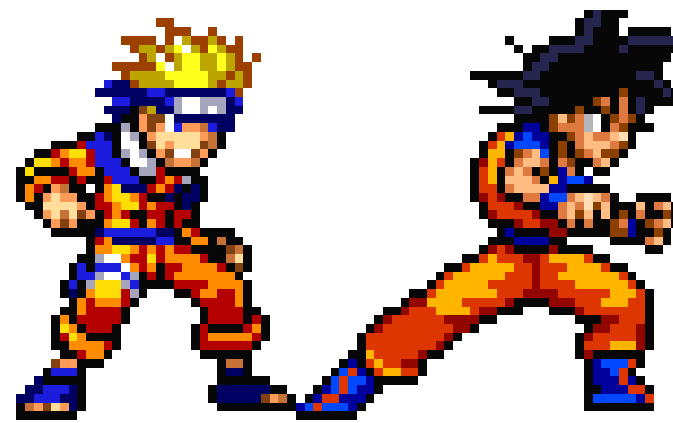
Captions are provided for all prerecorded audio content in synchronized media, except when the media is a media alternative for text and is clearly labeled as such. [Show full description](#) [Understanding 1.2.2](#)

[Show techniques and failures for 1.2.2](#)

[Share](#) | [Back to Top](#)

<https://www.w3.org/WAI/WCAG21/quickref/?versions=2.0#text-alternatives>

Let's talk more technically and see the real action in ReactJS...





```
<a href="#maincontent" class="skip-link" >Skip to main content</a>
```

```
<nav>
```

```
...
```

```
</nav>
```

```
<main id="maincontent" tabindex="-1">
```

```
...
```

```
</main>
```

`tabindex="-1"`

- Not in the natural tab order
- Can be programmatically focused with `focus()` method

*tabindex="-1"*

- *Not in the natural tab order*
- *Can be programmatically focused with `focus()` method*

```
document.querySelector('#modal').focus()
```

**Oh Hai!**

Join our newsletter?

DECLINE

ACCEPT

*tabindex="0"*

- *In the natural tab order*
- *Can be programmatically focused*



```
<body>
```

```
<label>This element will be focused second</label>
```

```
<input name="field1" type="email" tabindex="2" />
```

```
<label>This element will be focused first</label>
```

```
<textarea rows="4" cols="5" tabindex="1"></textarea>
```

```
<label>The second focused last</label>
```

```
<input name="field3" type="text" tabindex="3"/>
```

```
</body>
```



```
button: hover,  
button: focus {  
    background: #E91E63;  
    color: #FFFFFF;  
    text-decoration: underline;  
}
```

```
button: focus {  
    outline: 0;  
    box-shadow: 0 0 8px 3px  
    rgba(255, 255, 255, 0.8);  
}
```

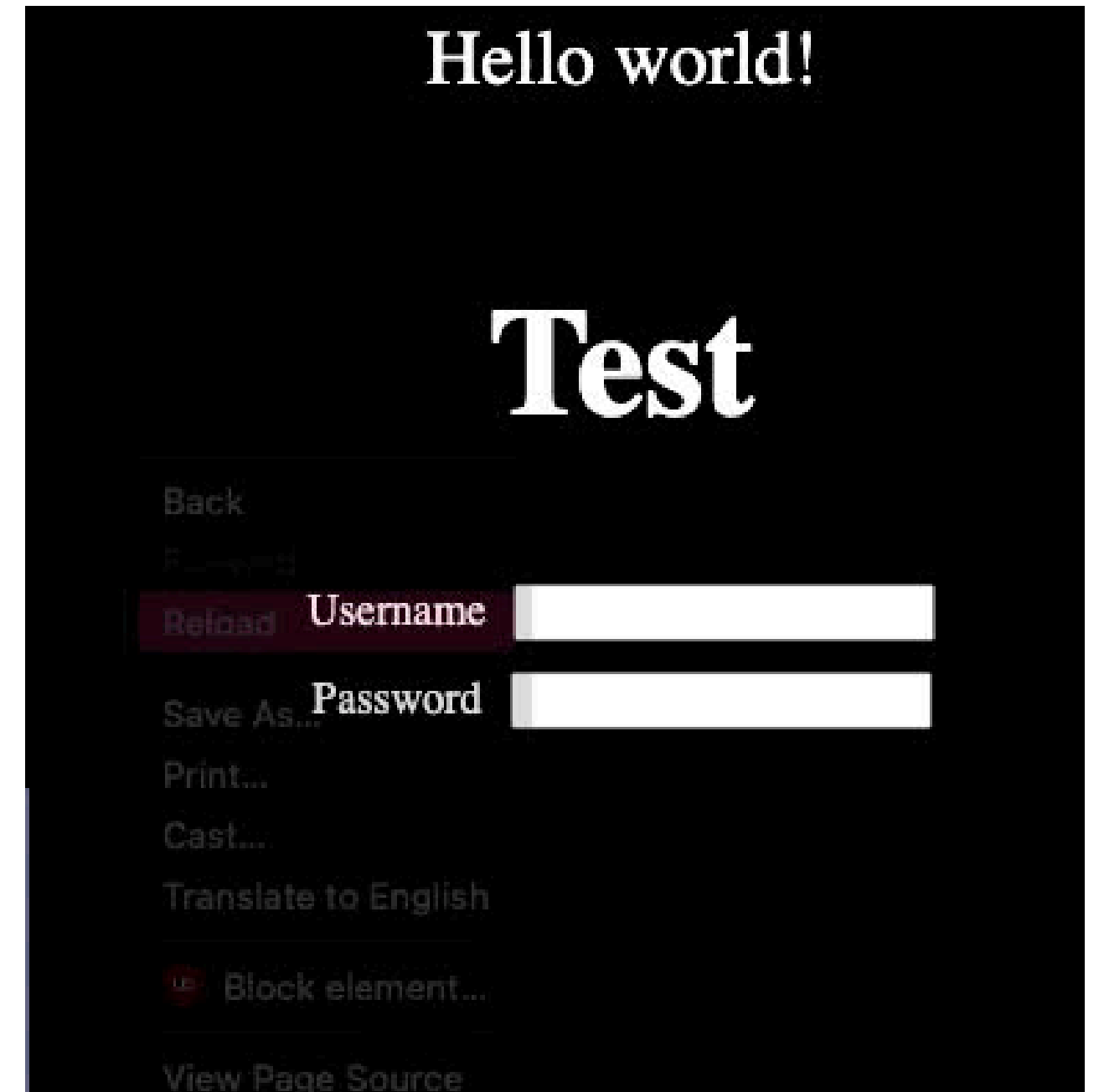
WOMBAT

QUOKKA

WALLABY

# Autofocus

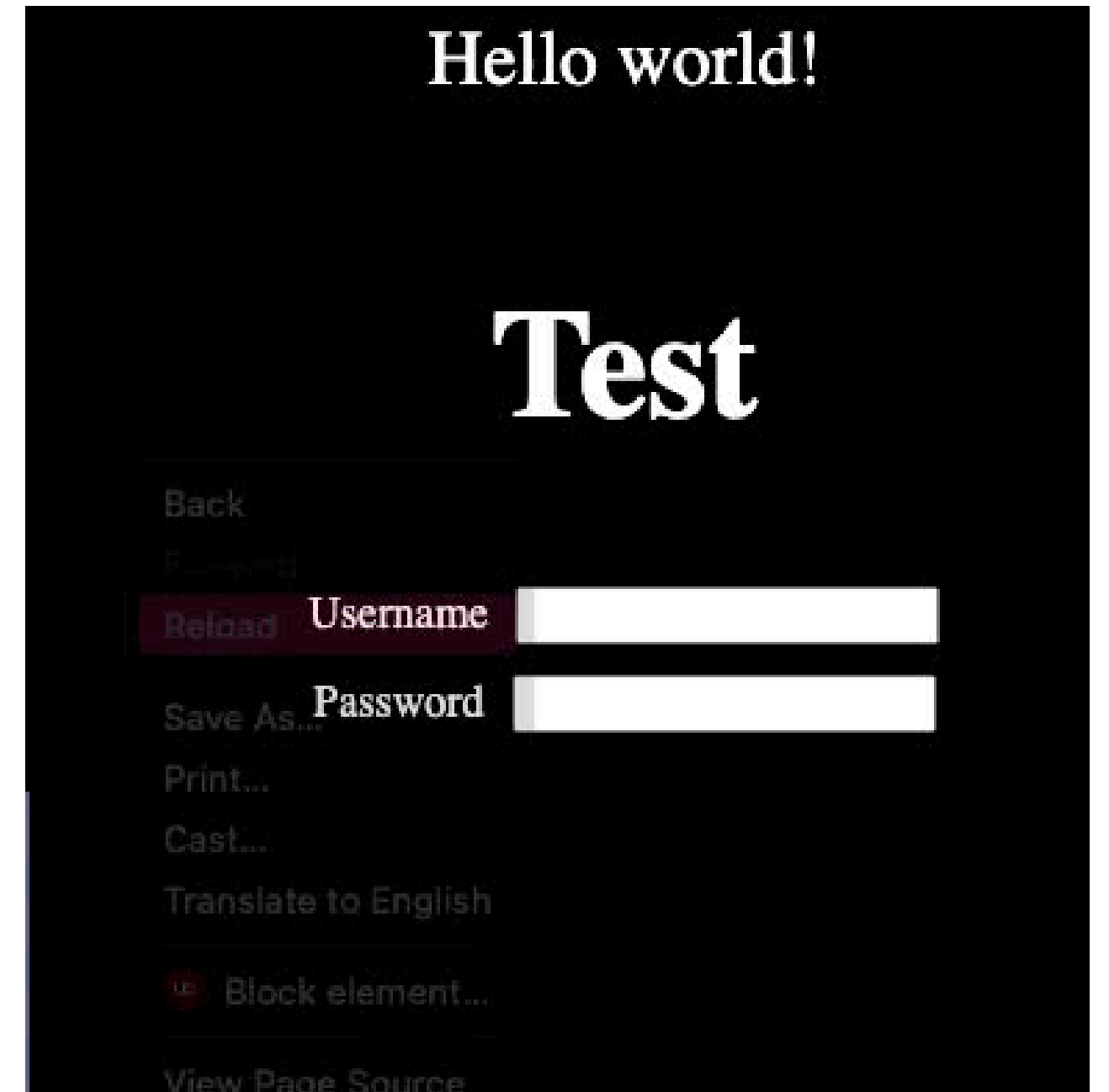
```
<form>
  <label>
    Username
    <input name="user" type="user" />
  </label>
  <label>
    Password
    <input name="user" type="password"
autoFocus />
  </label>
</form>
```





# focus with Ref & hooks

```
export default function Login(props) {
  const pwinput = useRef(null);
  useEffect(() => {
    pwinput.current.focus();
  })
  return (
    <form>
      <label>
        Username
        <input name="user" type="user" />
      </label>
      <label>
        Password
        <input name="user" type="password"
          ref={pwinput} />
      </label>
    </form>
  )
}
```



```
<a href="#internal">Now I'm a link</a>
```

```
<a href="#internal">Now I'm a link</a>
```

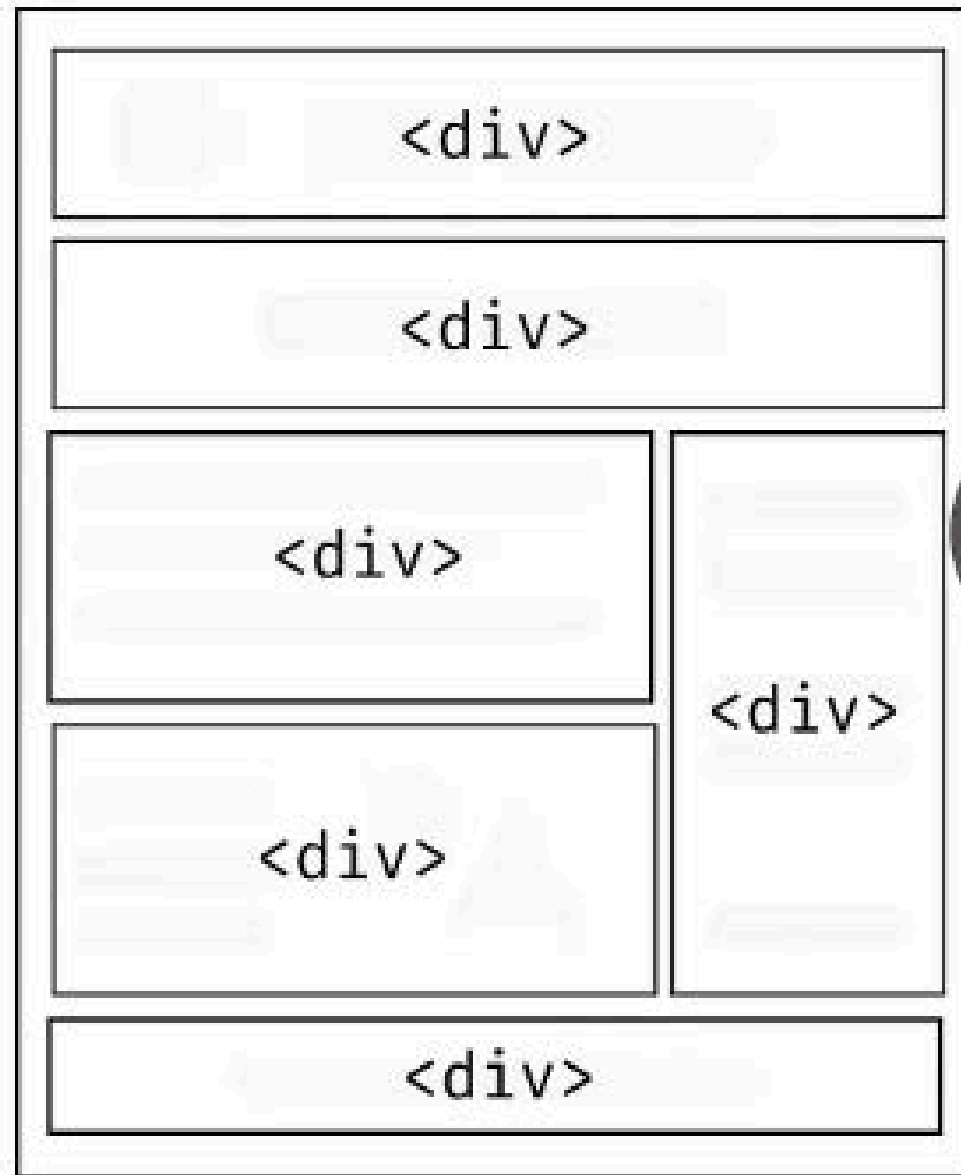
✓ *shows up in links list*

```
<a href="#internal">Now I'm a link</a>
```

- ✓ *shows up in links list*
- ✓ *works with the keyboard*

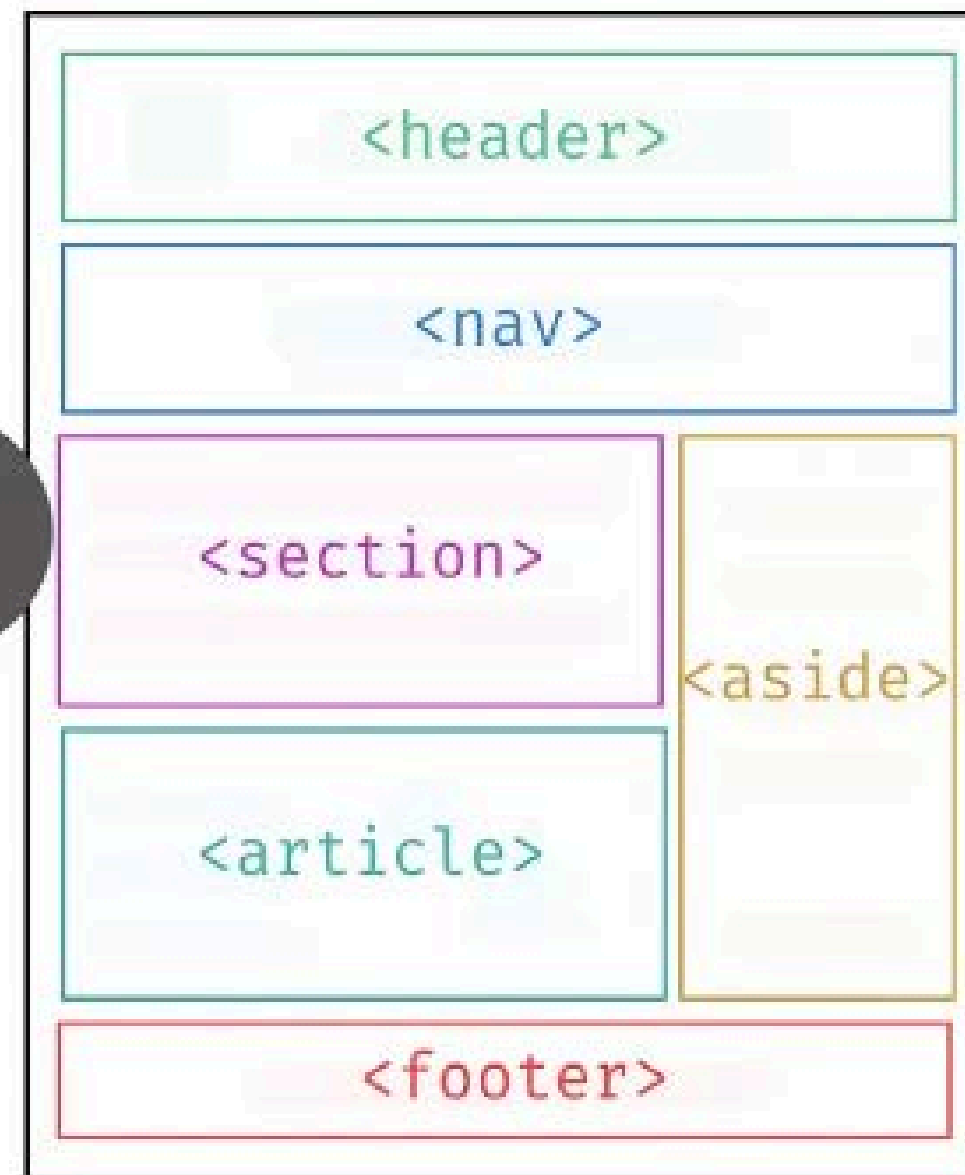
```
<a href="#internal">Now I'm a link</a>
```

- ✓ *shows up in links list*
- ✓ *works with the keyboard*
- ✓ *can be bookmarked*



**Ambiguous Sections**

VS



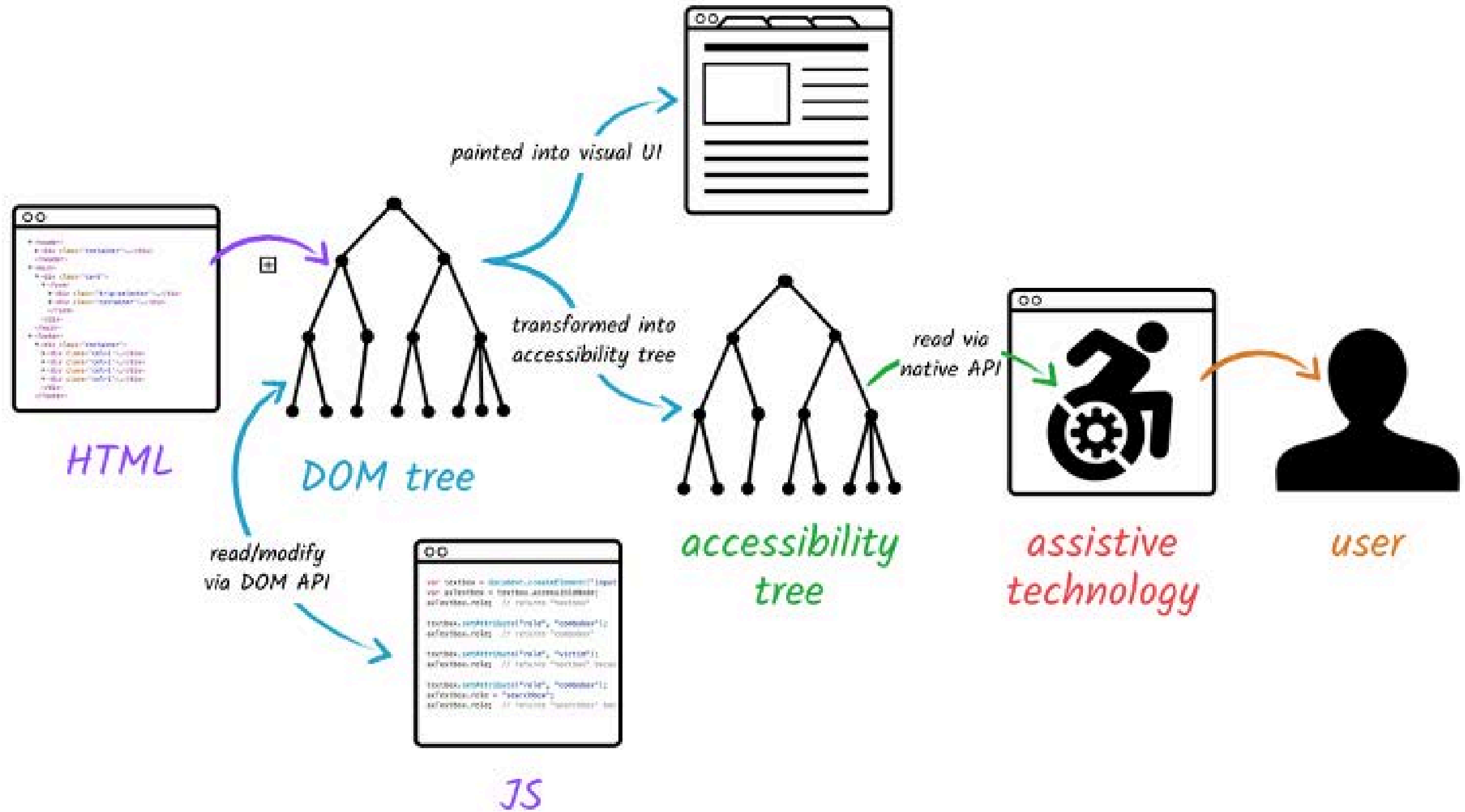
**Clear Sections**

compiles into:

```
• • •  
<MyButton color="blue" shadowSize={2}>  
  Click Me  
</MyButton>
```

```
• • •  
React.createElement(  
  MyButton,  
  {color: 'blue', shadowSize: 2},  
  'Click Me'  
)
```







```
<html lang="en">
  <head> ... </head>
  <body data-theme="light">
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root">
      <div class="JuTe6q_spectrum _5QszkG_spectrum _5QszkG_i18nFontFamily PFjRbG_spectrum--light xSyFOq_spectrum--medium zA6MfG_spectrum zA6MfG_spectrum--dark zA6MfG_spectrum--darkest zA6MfG_spectrum--large zA6MfG_spectrum--light zA6MfG_spectrum--lightest zA6MfG_spectrum--medium" lang="id-ID" dir="ltr" style="isolation: isolate; color-scheme: light;">
        <div class="Mw91UW_flex" style="flex-direction: column;"> flex
          <header>
            <nav class="container-fluid"> flex
              <ul> ... </ul> flex
              <ul> flex
                <li> ... </li>
                <li> ... </li>
                <li> ... </li>
                <li>
                  ::before
                  <button class="Dniwja_spectrum-ActionButton Dniwja_spectrum-BaseButton Dniwja_i18nFontFamily Dniwja_spectrum-FocusRing Dniwja_spectrum-FocusRing-ring Dniwja_spectrum-ActionButton--emphasized" type="button" aria-label="Dark Mode" aria-pressed="false" style="cursor: pointer;"> ... </button> flex == $0
```

▼ Accessibility Tree

Enable full-page accessibility tree

▼ RootWebArea "ReactTodos"

- ▼ generic
  - ▼ generic
    - ▼ generic
      - ▼ banner
        - ▼ navigation
          - ▼ list
            - ▼ listitem
              - button "Dark Mode"

▼ ARIA Attributes

aria-label: Dark Mode  
aria-pressed: false

▼ Computed Properties

▼ Name: "Dark Mode"

aria-labelledby: Not specified  
aria-label: "Dark Mode"  
From label: Not specified  
Contents: Not specified  
title: Not specified  
Role: button  
Invalid user entry: false  
Focusable: true  
Pressed: false



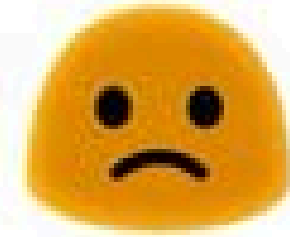
Receive promotional offers

```
<input type="checkbox" checked name="jLetter">
```

Receive promotional offers?



Receive promotional offers



```
<input type="checkbox" checked name="jLetter">
```

Receive promotional offers?

“

*checked, checkbox*”



## Receive promotional offers

```
<label>  
  <input type="checkbox" checked name="jLetter">  
  Receive promotional offers?  
</label>
```

“Receive promotional offers, checked, checkbox”



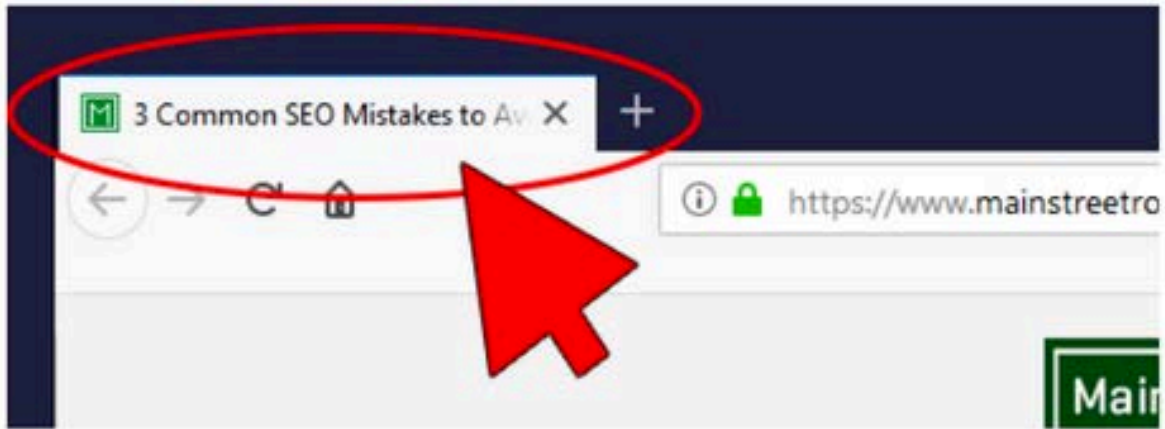
Receive promotional offers

```
<input type="checkbox" checked name="jLetter" id="letter">  
<label for="letter">Receive promotional offers?</label>
```

“Receive promotional offers, checked, checkbox”

# SPA and get titles

To get this...



...use this!

<https://github.com/nfl/react-helmet>

```
<Helmet><title>My Page Title</title></Helmet>
```

**WAI - ARIA**

**Web**

**Accessibility**

**Initiative**

**Accessible**

**Rich**

**Internet**

**Applications**





# Add semantics



Receive promotional offers

```
<div class="checkbox checked">  
  Receive promotional offers  
</div>
```

*text*

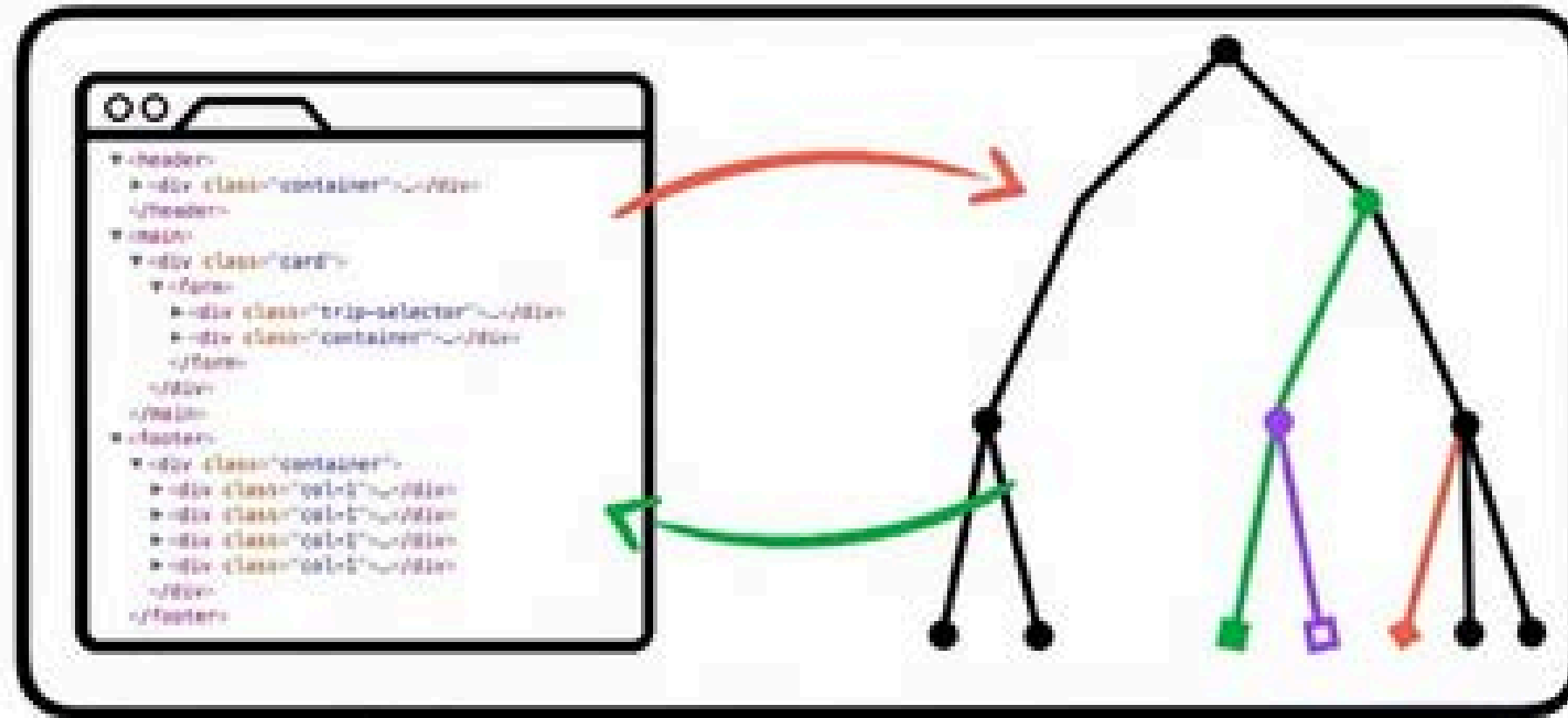
*value: "Receieve promotional offers"*

```
<div role="checkbox"  
  aria-checked="true">  
  Receive promotional offers  
</div>
```

*checkbox*

*name: "Receieve promotional offers"*

*state: checked*

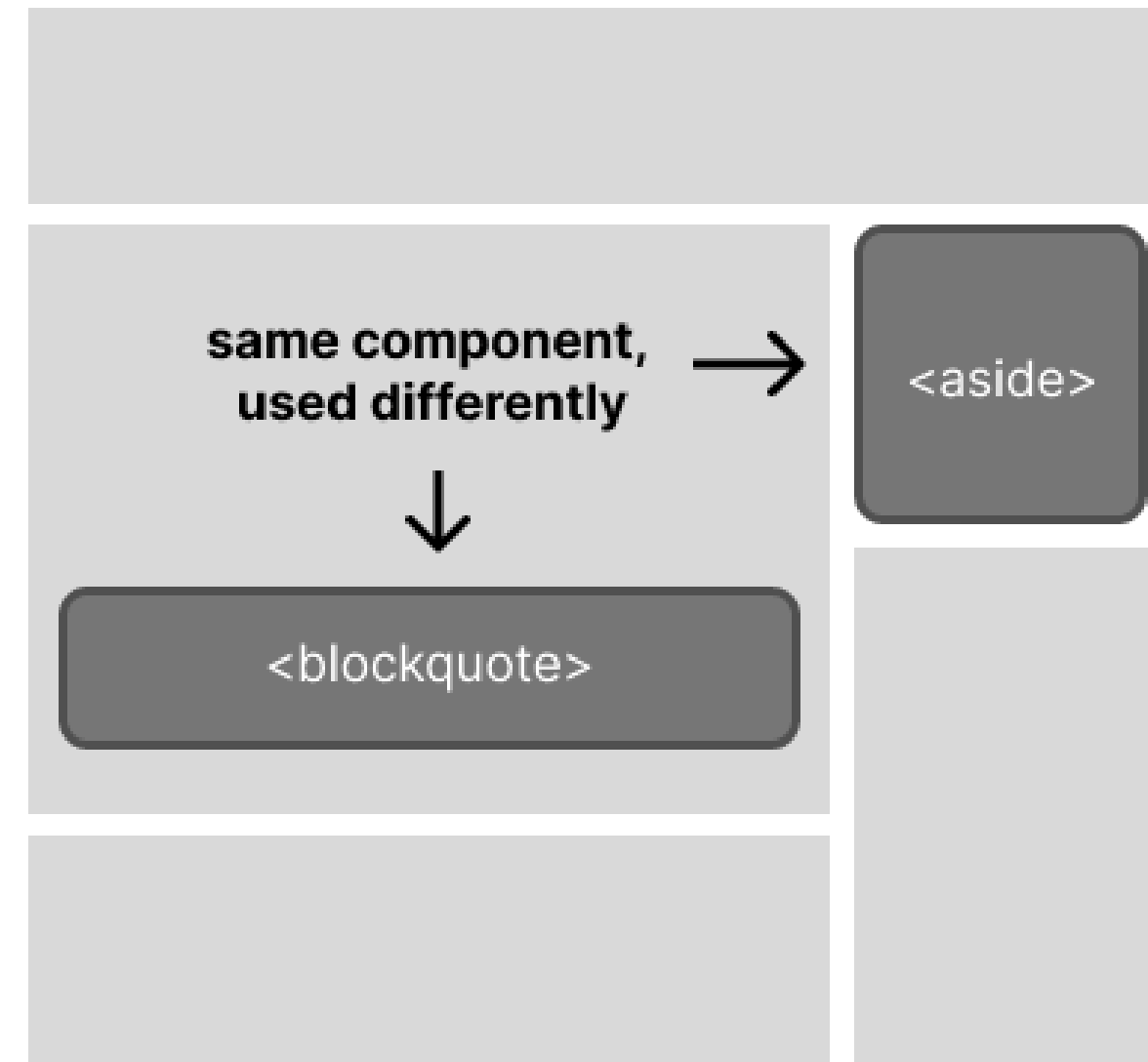


DOM  
+  
ARIA

accessibility  
tree

# Abstracting Components

Keep in mind the various roles or descriptions that your component might need, based on context.



# Abstracting Components using the “as” prop pattern



```
const Dynamic = ({ as = 'h1', children, ...props }) =>
  const As = as;
  return <As {...props}>{children}</As>;
};

export default Dynamic;
```

# Abstracting Components using the “as” prop pattern

in jsx



```
<Dynamic style={{textAlign: 'center'}} as="p">Welcome to react-todos</Dynamic>
```

compiles browser

```
<header> == $0
  <nav data-testid="nav" class="container-fluid" aria-label="Navigation">⋮</nav>
    <p style="text-align: center;">Welcome to react-todos</p>
</header>
```

# aria-labelledby

```
<div id="men-lbl">  
  Men's T-Shirts  
</div>
```

```
<button id="men-btn"  
  aria-labelledby="men-lbl men-btn">
```

```
  Shop Now  
</button>
```

Men's T-Shirts


SHOP NOW

*button*

name: "Men's T-Shirts  
Shop Now"

# aria-describedby

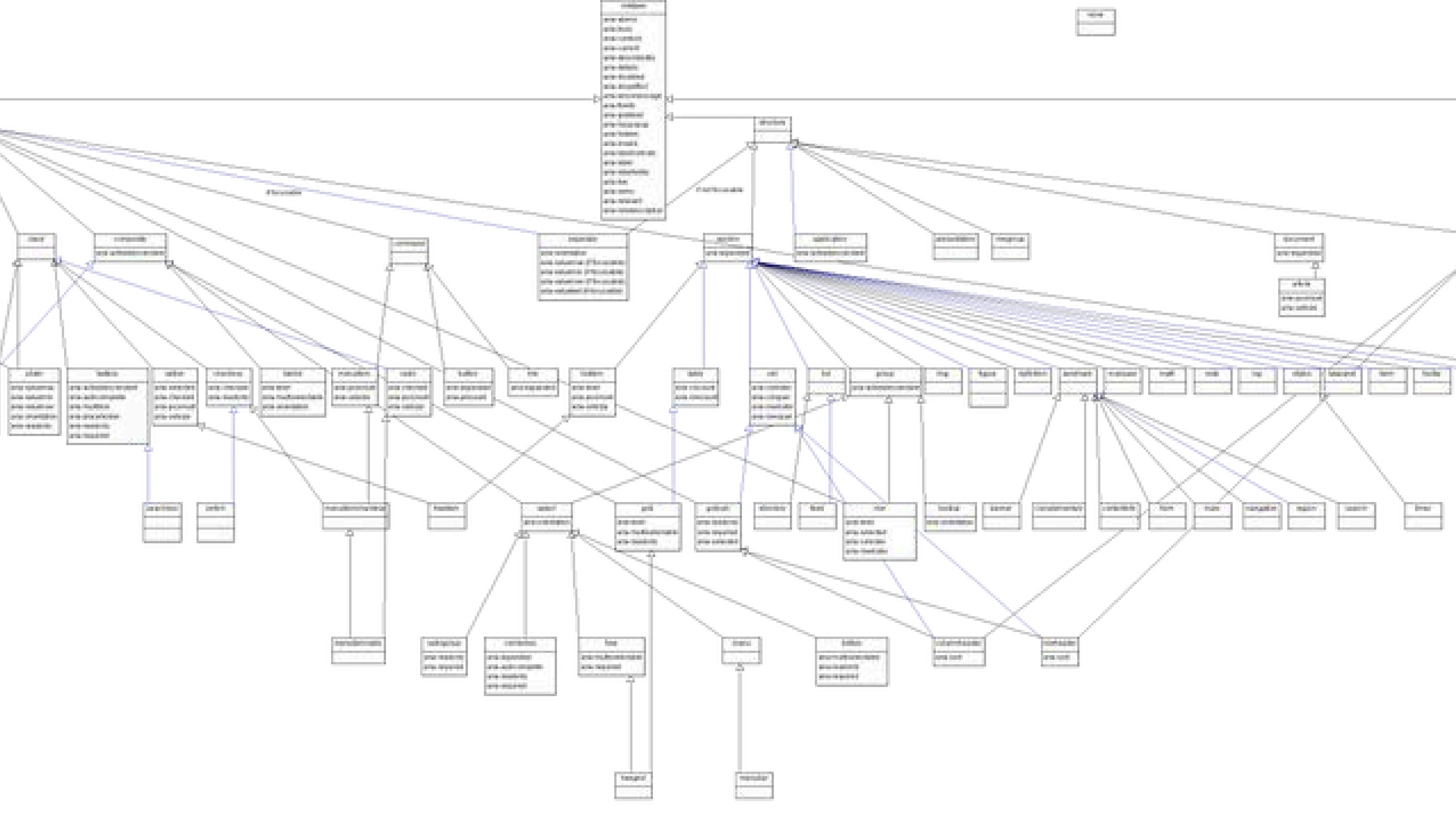
```
<label for="pw">Password:</label>  
<input type="password" id="pw"  
      aria-describedby="pw-help">  
<div id="pw-help">  
  Password must be at least 12 characters  
</div>
```



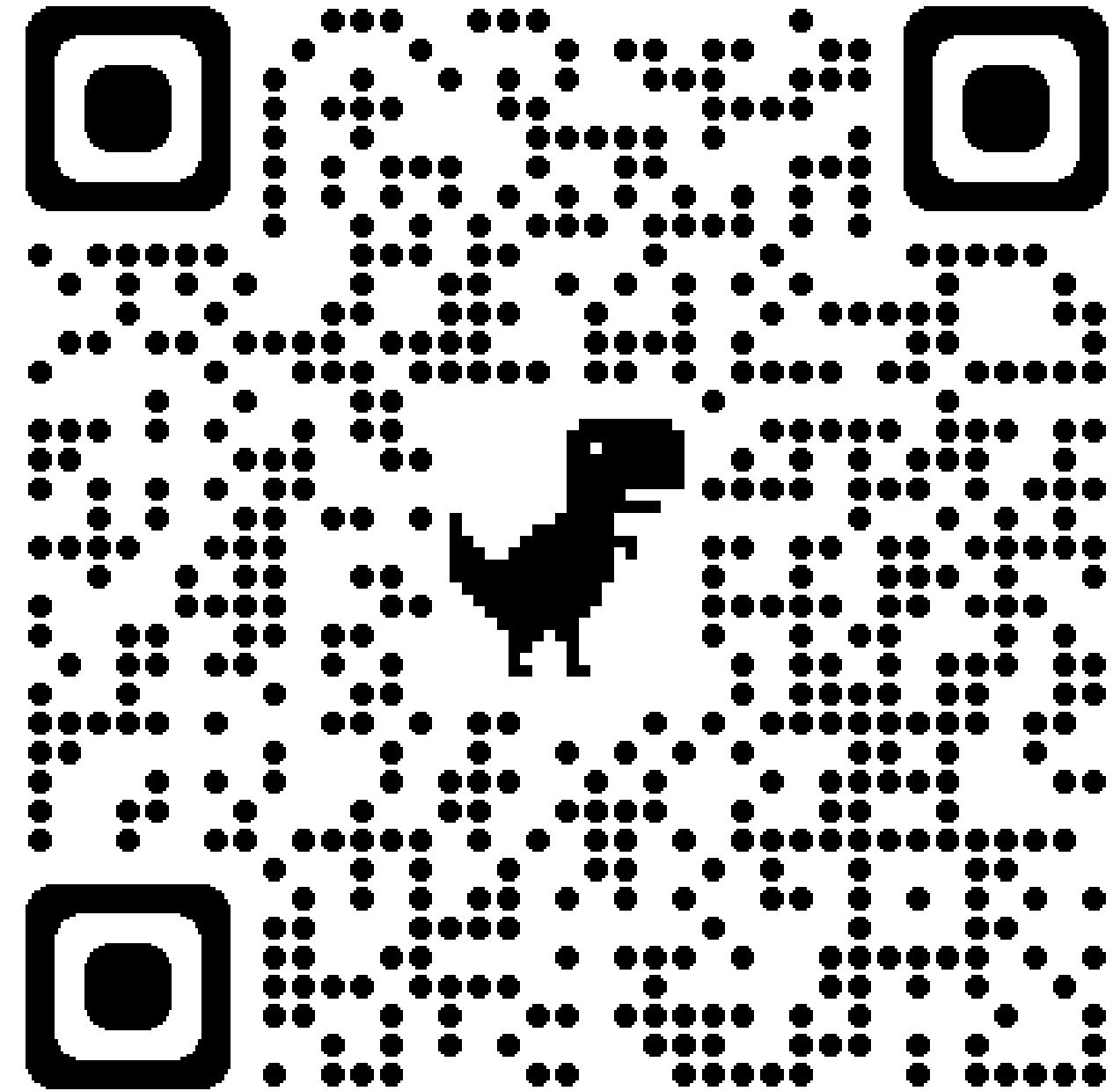
Password:

*Password must be at least 12 characters*

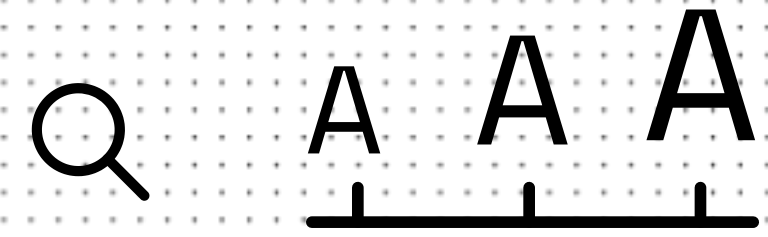




WAI-ARIA



*[bit.ly/aria-guide](https://bit.ly/aria-guide)*



# Audit Accessibility



# How to audit

1. Manual Check

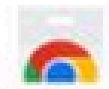
2. Tools

- **lighthouse**
- **Wave**



# Screen Reader

*A web-powered screen reader made by Google.*



chrome web store

---

[Home](#) > [Extensions](#) > Screen Reader



## Screen Reader

 [chrome.google.com](https://chrome.google.com)


★★★★★ 1,036 ⓘ | [Developer Tools](#) | 200,000+ users

 By Google

---

# Accessibility Insights for Web


*Accessibility Insights for Web helps developers quickly find and fix accessibility issues.*




chrome web store

---

[Home](#) > [Extensions](#) > Accessibility Insights for Web



## Accessibility Insights for Web

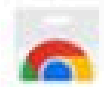
 **Featured**

★★★★★ 35 ⓘ | [Developer Tools](#) | 100,000+ users

---

# axe DevTools - Web A11Y Testing

*Accessibility Checker for Developers, Testers, and Designers in Chrome.*



chrome web store

---

[Home](#) > [Extensions](#) > [axe DevTools - Web Accessibility Testing](#)



## axe DevTools - Web Accessibility Testing



[www.deque.com](http://www.deque.com)



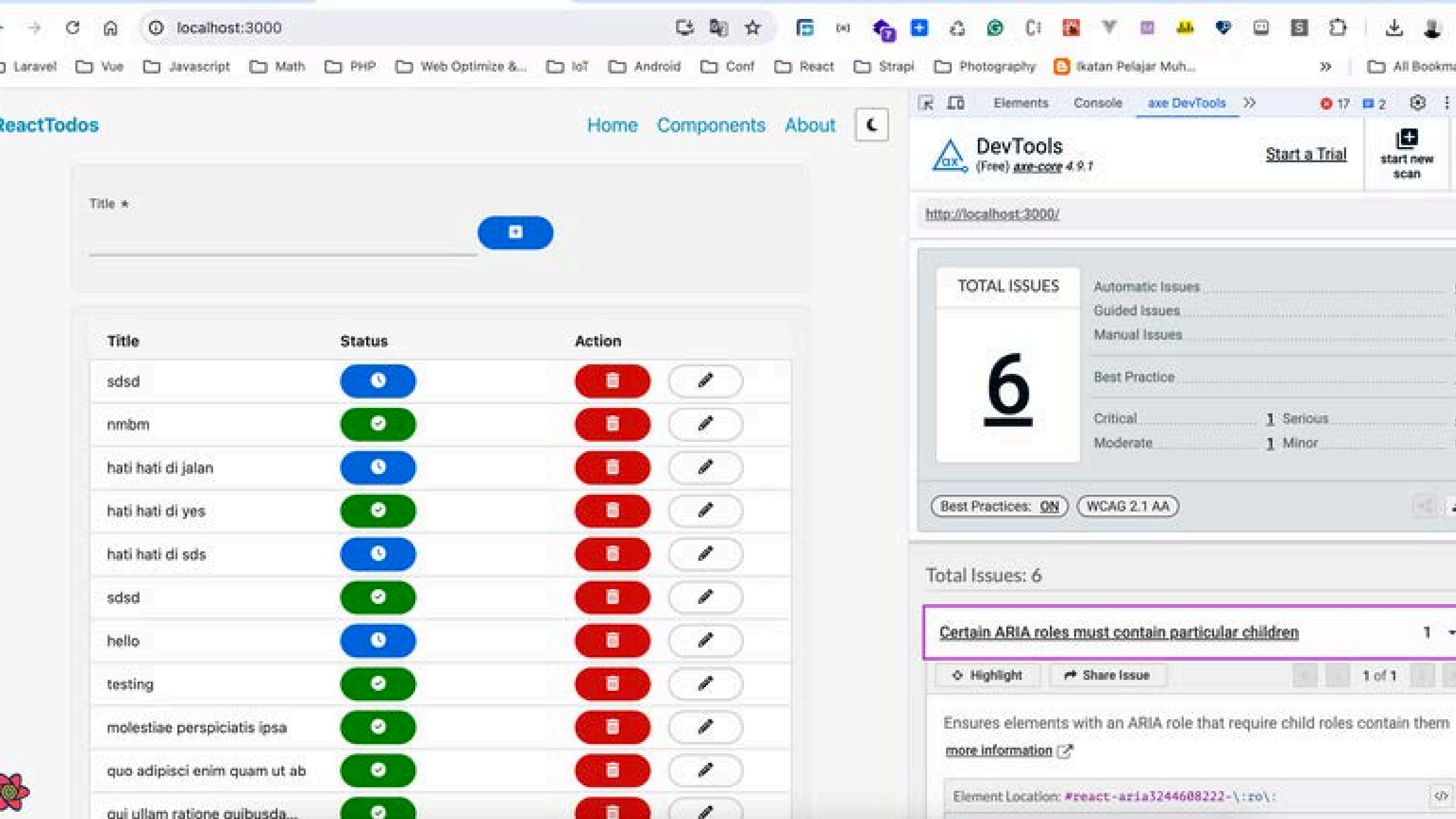
Featured

★★★★★ 97 ⓘ

[Accessibility](#)

| 300,000+ users

---



Title \*



Title	Status	Action
sdsd		
nmbm		
hati hati di jalan		
hati hati di yes		
hati hati di sds		
sdsd		
hello		
testing		
molestiae perspiciatis ipsa		
quo adipisci enim quam ut ab		
qui ullam ratione quibusda...		

Elements Console axe DevTools

DevTools (Free) axe-core 4.9.1

Start a Trial

start new scan

http://localhost:3000/

TOTAL ISSUES

6

Automatic Issues  
Guided Issues  
Manual Issues  
Best Practice

Critical 1 Serious  
Moderate 1 Minor

Best Practices: ON WCAG 2.1 AA

Total Issues: 6

Certain ARIA roles must contain particular children 1

Highlight Share Issue 1 of 1

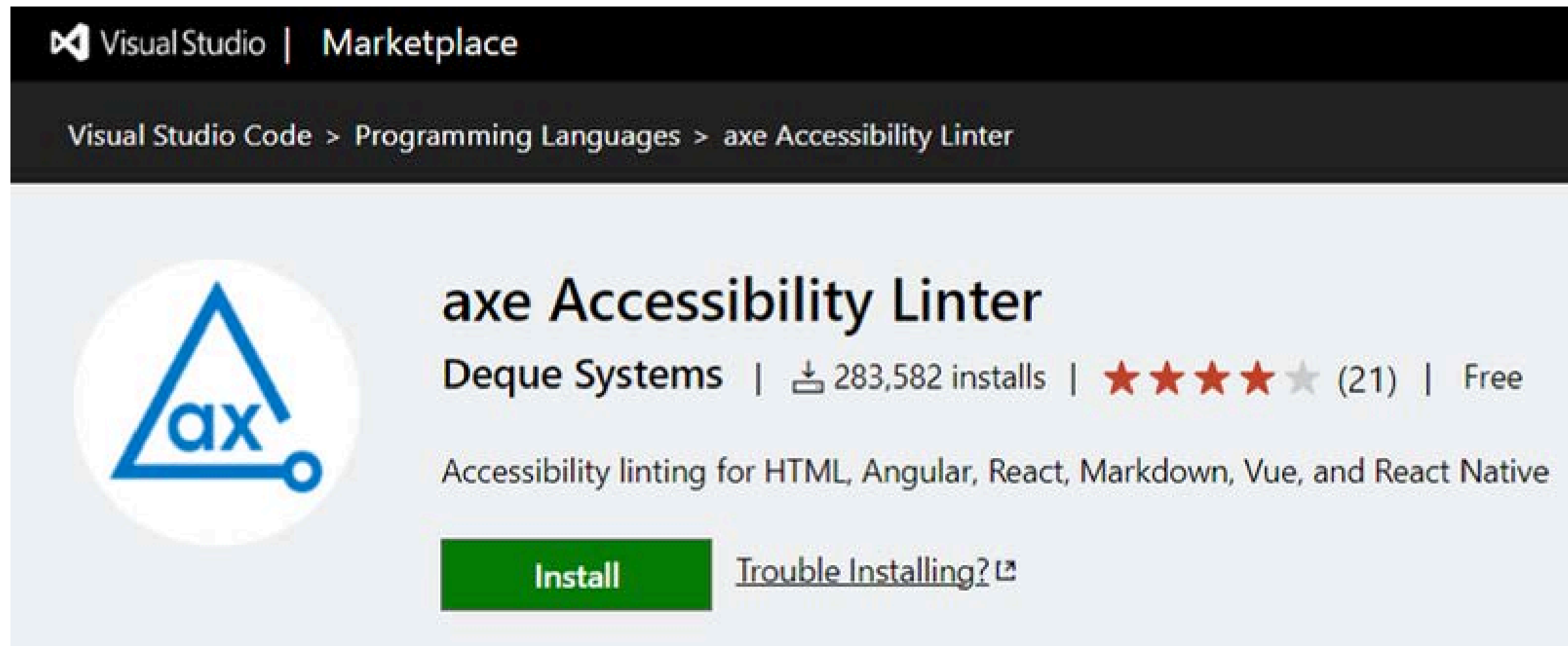
Ensures elements with an ARIA role that require child roles contain them more information

Element Location: #react-aria3244608222-\:ro\:




## axe Accessibility Linter

*Check code files for common accessibility defects. Checks React (JSX), React Native, Angular, Vue, HTML, and Markdown.*





Visual Studio | Marketplace

Visual Studio Code > Programming Languages > axe Accessibility Linter



### axe Accessibility Linter

Deque Systems |  283,582 installs |  (21) | Free

Accessibility linting for HTML, Angular, React, Markdown, Vue, and React Native

[Install](#) [Trouble Installing?](#)

## Accessibility UI Component Library

*UI component libraries can get you up and running on a new project pretty fast.*

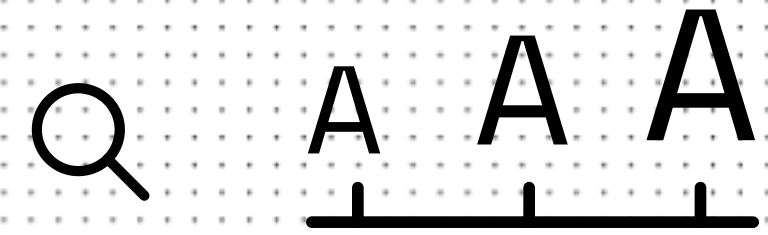


Title \*

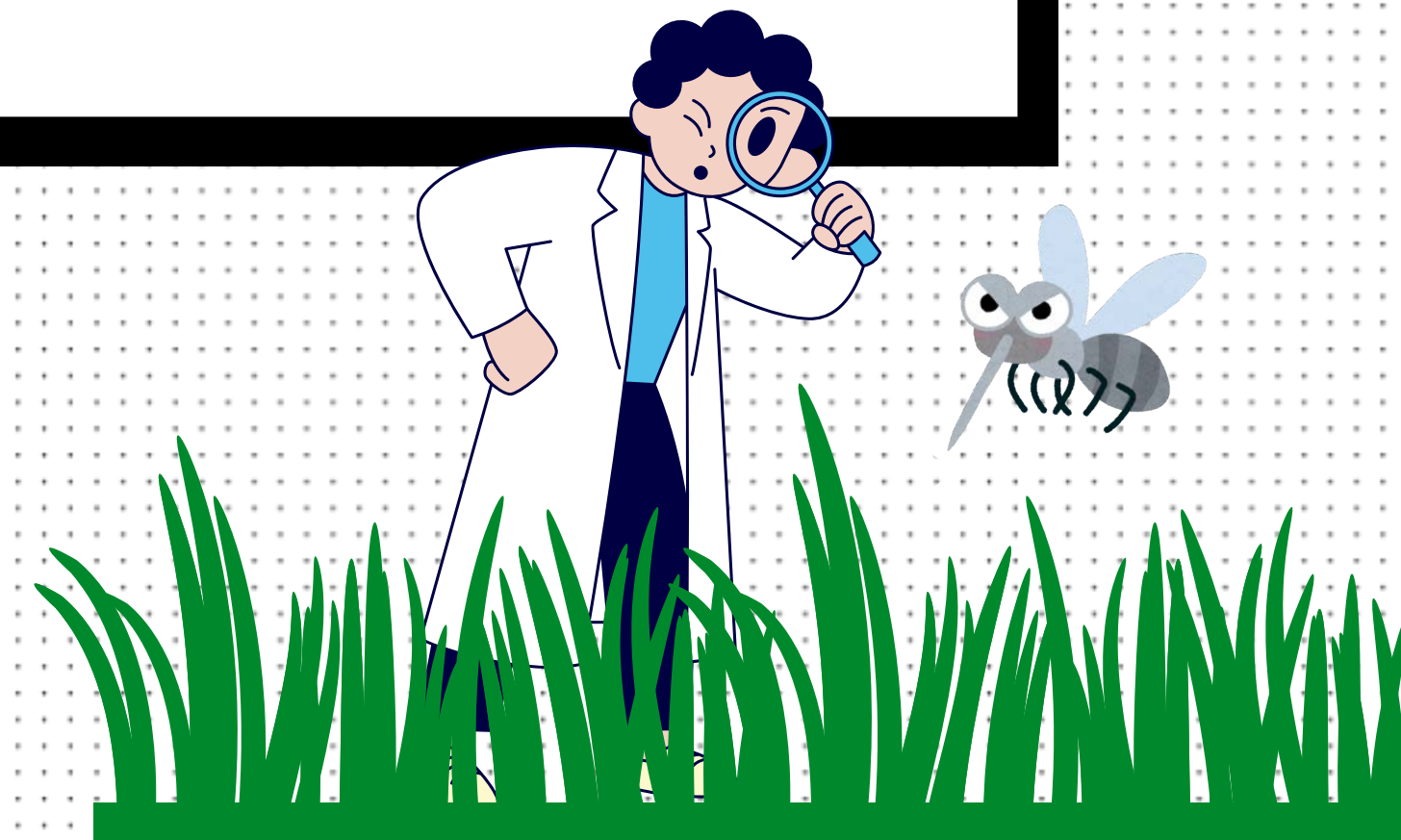
  

Title	Status	Action
sdsd	<input type="checkbox"/>	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
nmbm	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
hati hati di jalan	<input type="checkbox"/>	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
hati hati di yes	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
hati hati di sds	<input type="checkbox"/>	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
sdsd	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
hello	<input type="checkbox"/>	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
testing	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
molestiae perspiciatis ipsa	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
quo adipisci enim quam ut ab	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/> <input type="button" value="Edit"/>

Link List About Link List



# Unit Testing



# Testing a button

```
describe('Nav tests', () => {  
  it('should contains button', () => {  
    const handleDarkMock = jest.fn();  
    render(<Nav handleDark={handleDarkMock} />);  
    const button = screen.getByRole('button', { name: 'Dark Mode' });  
    fireEvent.click(button);  
    expect(button).toHaveAttribute('aria-pressed', 'true');  
  });  
});
```

---

}); Raihan Nismara, 4 days ago • Unit testing for ally: button role. pressed

# Testing buttons

```
const MyComponent = ({ onClick }) => {  
  return (  
    <div>  
      <button onClick={onClick}>Button 1</button>  
      <button onClick={onClick}>Button 2</button>  
      <button onClick={onClick}>Button 3</button>  
    </div>  
  );  
};
```

# Testing buttons

```
...
test('should call onClick handler for each button click', () => {
  const handleClick = jest.fn();
  render(<MyComponent onClick={handleClick} />);

  const buttons = screen.getAllByRole('button');

  buttons.forEach(button => {
    fireEvent.click(button);
  });

  expect(handleClick).toHaveBeenCalledTimes(buttons.length);
});
```

@testing-library

# Testing text/heading

```
getByRole('heading', {level: 1});  
// <h1>Heading Level One</h1>
```

```
getAllByRole('heading', {level: 2});  
// [  
//   <h2>First Heading Level Two</h2>,  
//   <div role="heading" aria-level="2">Second Heading Level Two</div>  
// ]
```



@axe-linter

# Combining with linter

```
return (  
  <nav data-testid="nav" className="..." aria-label="Navigation"> "testid": Unknown wor  
    <ul>  
      <li>  
        <Link href="/">  
          </img> axe-linter (image-alt): Ensures <img> element  
        </Link>  
      </li>  
    </ul> You, 9 months ago • Adding crud, react-spectrum,etc  
    <ul>  
      <li><Link href="/">Home</Link></li>
```

@jsx-a11y

# eslint-plugin-jsx-a11y

## eslint-plugin-jsx-a11y

Static AST checker for accessibility rules on JSX elements.

*Read this in other languages.*

Mexican Spanish 

### Why?

This plugin does a static evaluation of the JSX to spot accessibility issues in React apps. Because it only catches errors in static code, use it in combination with [@axe-core/react](#) to test the accessibility of the rendered DOM. Consider these tools just as one step of a larger a11y testing process and always test your apps with

```
src/app/schedule/index.js
```

```
Line 122:15: A form label must be associated with a control. jsx-a11y/label-has-associated-control
```

@cypress-axe

@axe-core

# Testing with Cypress

```
import "cypress-axe";
```

```
describe('sign up accessibility suite', () => {  
  | cy.contains('header nav a', 'Home');  
});
```

@cypress-axe

@axe-core

# Testing with Cypress

```
describe('sign up accessibility suite', () => {  
  before(() => {  
    cy.visit("http://localhost:3000/");  
    cy.injectAxe();  
  });  
  
  it('check entire page for a11y issue', () => {  
    cy.checkA11y();  
  });  
});
```

# Testing with Cypress

@cypress-axe  
@axe-core

The screenshot shows the Cypress test runner interface. The browser window displays a 'Create an account' form with fields for 'Email address', 'Password', and 'Confirm Password', along with a checkbox for 'I agree to the terms and conditions'. The test runner console on the left shows a failed test with the following error message:

```
4 -ally error! region on 8 Nodes 8
5 -assert: 4 accessibility violations were detected: expected 4 to equal **0**
! AssertionError
4 accessibility violations were detected: expected 4 to equal 0
node_modules/cypress-axe/dist/index.js:8
3:1
81 |         .then(function (violations) {
82 |           if (!skipFailures) {
> 83 |             assert.equal(violations.length, 0);
    |             ^
84 |           }
85 |           else if (violations.length) {
86 |             Cypress.log({
```

At the bottom of the console, there are links to 'View stack trace' and 'Print to console'.

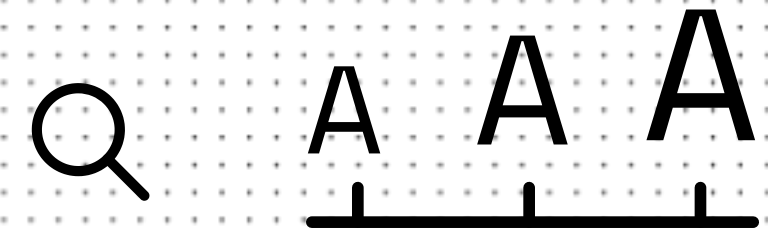
Q A A A

What benefits



# Benefits

1. Increases the audience by making it more inclusive
2. A better user experience
3. Improve search engine optimization(SEO)
4. It is morally the right thing to do
5. Avoid Potential Discrimination Lawsuits



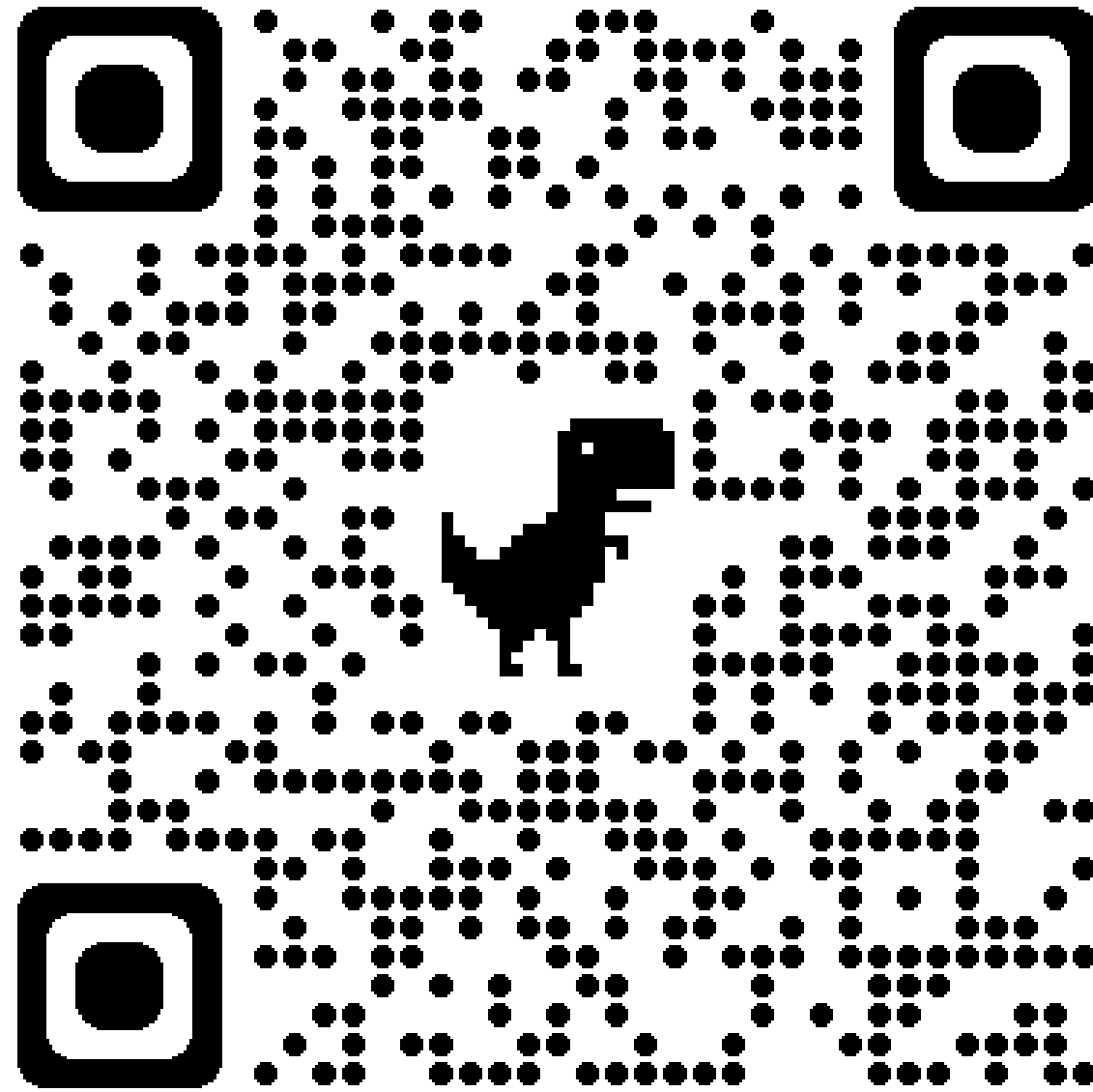
Resource



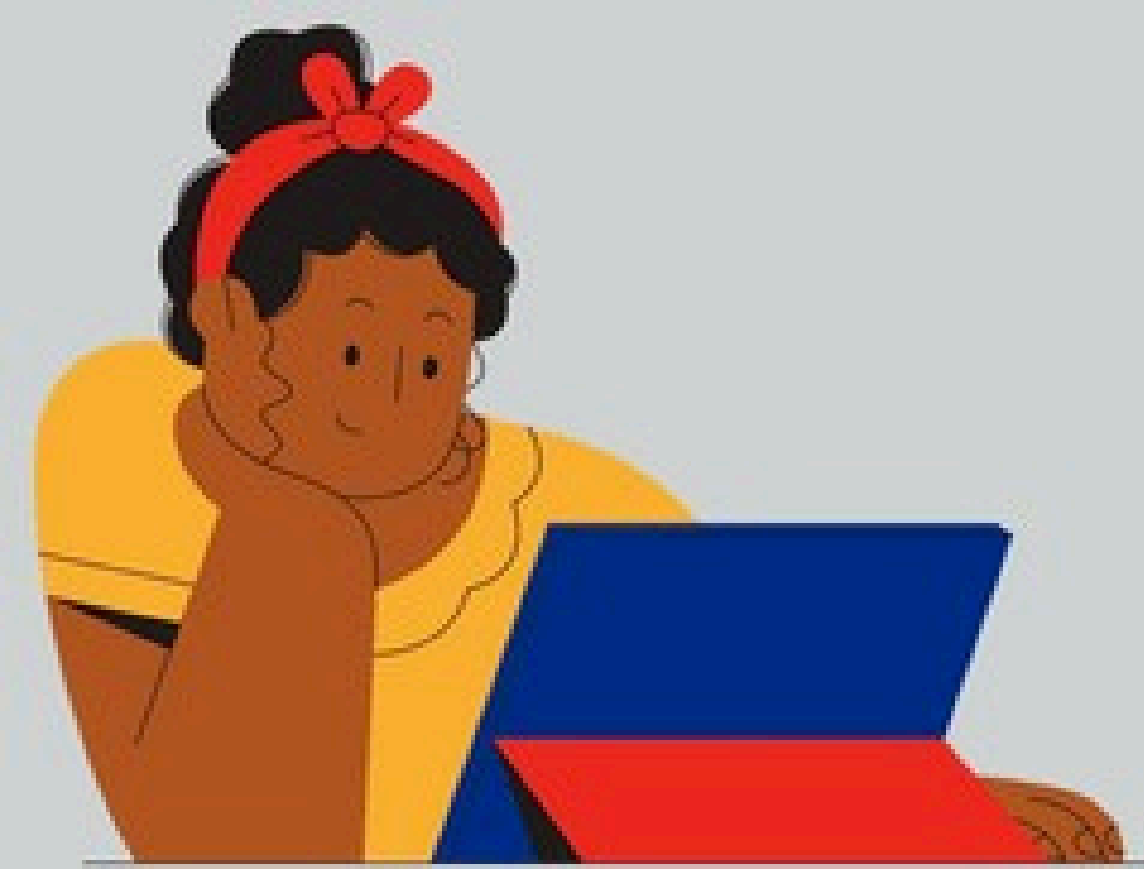


## Resource to Learn

1. W3C Accessibility Fundamental :  
<https://www.w3.org/WAI/fundamentals/>
2. A11Y development : <https://a11y-101.com/development>
3. Google Chrome Youtube Playlist :  
<https://www.youtube.com/playlist?list=PLNYkx0F6rcICWx0C9LVWWVqvH1YJyqw7g>
4. React & Accessibility 101 :  
<https://www.youtube.com/watch?v=kz9e81oRs3Y>

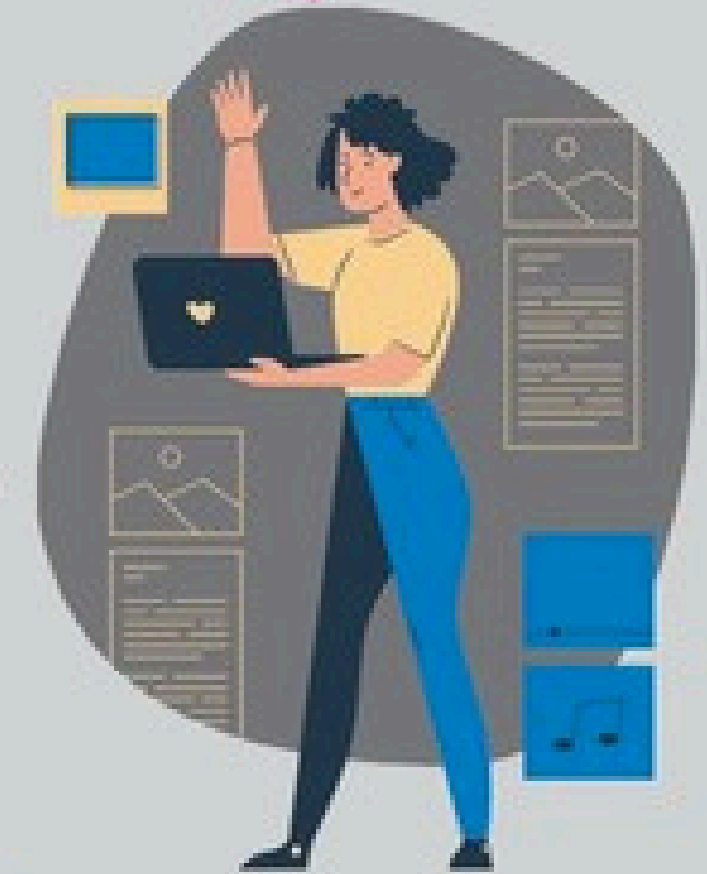


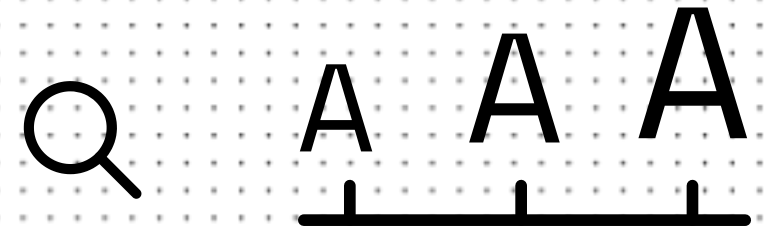
<https://github.com/raihan71/web-a11y-guide>



# WEB ACCESSIBILITY

benefits everyone





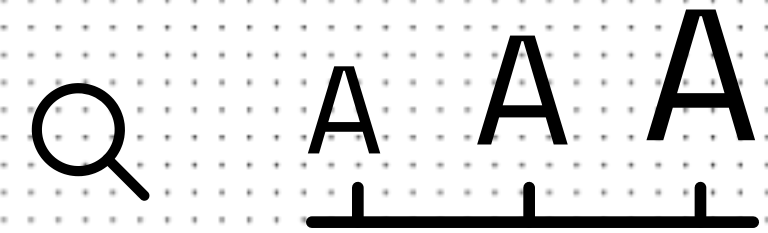
**Thank You**

 [raihan.my.id](https://raihan.my.id)

 [raihan.nismara](https://www.instagram.com/raihan.nismara)

 [codevcast](https://www.youtube.com/codevcast)

 [Raihan Nismara](https://www.linkedin.com/in/Raihan-Nismara)



**[bit.ly/yudi-web-accessible](http://bit.ly/yudi-web-accessible)**