European Accessibility Act (EAA)

Maaike van Putten
Software Developer & Trainer
www.brightboost.nl

{brightboost}

n

EAA

What and why?

p

WCAG

Web Content Accessibility Guidelines

c

POUR

Perceivable, Operable, Understandable, Robust

V

Must Haves

Non UX/UI

UX/UI

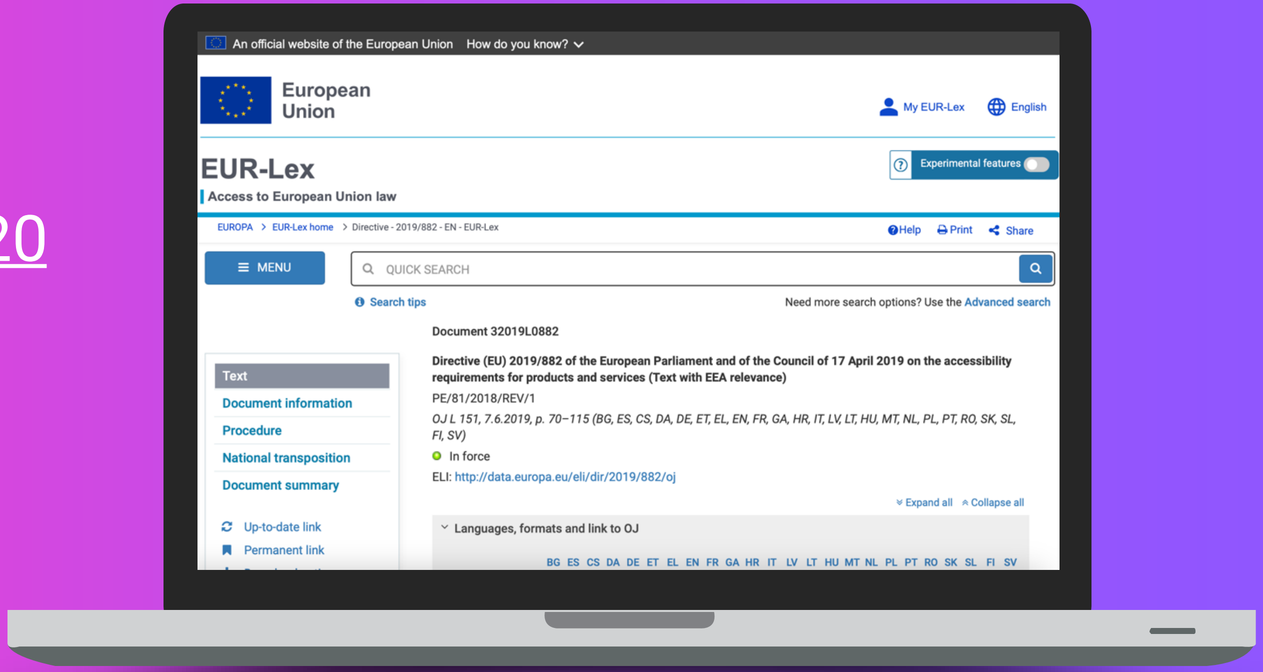
European Accessibility Act (EAA)

An EU directive that determines which products must be accessible to people with disabilities

European Accessibility Act (EAA)

Easy, just read it here...

https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32019L0882


An official website of the European Union
How do you know?
European Union
My EUR-Lex
English
EUR-Lex
Access to European Union law
Experimental features
EUROPA > EUR-Lex home > Directive - 2019/882 - EN - EUR-Lex
Help
Print
Share
MENU
QUICK SEARCH
Search tips
Need more search options? Use the Advanced search
Text
Document information
Procedure
National transposition
Document summary
Up-to-date link
Permanent link
Expand all
Collapse all
Languages, formats and link to OJ
BG ES CS DA DE ET EL EN FR GA HR IT LV LT HU MT NL PL PT RO SK SL FI SV

Document 32019L0882

Directive (EU) 2019/882 of the European Parliament and of the Council of 17 April 2019 on the accessibility requirements for products and services (Text with EEA relevance)

PE/81/2018/REV/1

OJ L 151, 7.6.2019, p. 70–115 (BG, ES, CS, DA, DE, ET, EL, EN, FR, GA, HR, IT, LV, LT, HU, MT, NL, PL, PT, RO, SK, SL, FI, SV)

● In force

ELI: http://data.europa.eu/eli/dir/2019/882/oj

{brightboost}

Directive (richtlijn)

A legislative act that sets out a goal that EU countries must achieve. It is up to the individual countries to devise their own laws to implement the act.

European Accessibility Act

- V Standardization
- I Broad scope
- b Legal compliance
- R Focus on usability
- N Enforcement

{brightboost}

Who does EAA benefit?

Q People with disabilities

V Businesses

H Society

P Developers and designers

{brightboost}

Video European Commission

https://ec.europa.eu/social/main.jsp?catId=1202

Internal market

The internal market of the European Union (EU) is a single market in which the free movement of goods, services, capital and persons is assured, and in which citizens are free to live, work, study and do business.

The EAA will help at least 87 million people with disabilities, including elderly.

Compliance deadline

- New products/services covered by the act released after **June 28, 2025**, must comply with the accessibility requirements.
- Article 32 (about services): “transitional period ending on 28 June 2030 during which service providers may continue to provide their services using products which were lawfully used by them to provide similar services before that date.”
- Exception for certain terminals that are already in use, those can be used for their lifetime with a maximum of 20 years (June 28, 2045).

Key compliance areas

f

Accessibility statements

- Mandatory for websites, apps, and other digital services
- Must clearly state how the product/service meets accessibility requirements

e

Monitoring & Reporting

- National authorities will monitor compliance and report to the European Commission
- Businesses may be required to provide regular updates on compliance status

N

Penalties

- Non-compliance can lead to legal actions

What industries, products, and services does the EAA apply to?

- Phone services
- Banking services
- E-commerce
- Websites, mobile services, electronic tickets and all sources of information for air, bus, rail and waterborne transport services
- E-books
- Access to Audio-visual media services (AVMS)
- Calls to the European emergency number 112

(source: https://ec.europa.eu/social/main.jsp)

Who should be compliant?

- Any business that trades in the EU
- Companies based outside the EU must comply with the EAA if they sell covered goods or services within the EU.
- Exempt:
  - Chapter 2, article 4.5
  - Chapter 5, article 14.1

How to be compliant?

I

The EEA does not impose detailed technical restrictions to make products and services accessible.

f

This will still depend on the exact implementation of the directive, likely to be largely WCAG 2.2 compliant.

U

Function requirement: Users must be able to consult a website’s content, structure and navigate through the pages (also with assistive devices or apps).

Exercise: screen reader

screen-reader.md

WCAG

Web Content Accessibility Guidelines

What is WCAG?

- Web Content Accessibility Guidelines
- Internationally recognized standards
- Created by the World Wide Web Consortium (W3C)
- Aims to make the internet more inclusive for people with disabilities
- Applies to all types of web content
    - Text
    - Images
    - Audio
    - Video
    - Interactive elements

What is WCAG?

L

Global standard

- Most widely recognized standard for web accessibility
- Guides legislation and policy in over 40 countries

Y

Regular updates

- Continuously updated to address new technologies and user needs

b

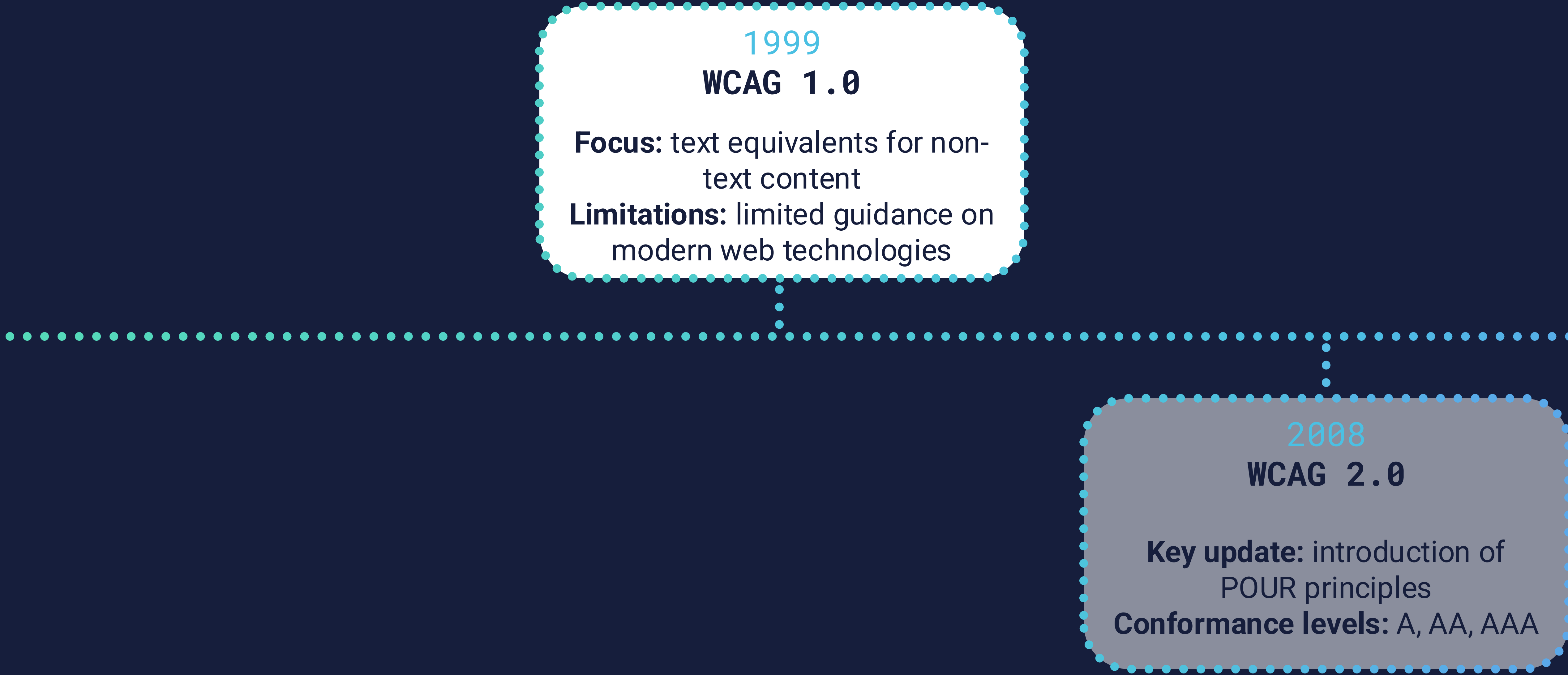
Legal compliance

- Adhering to WCAG can help organizations avoid legal actions

Who is WCAG for?

- Q People with disabilities: over 1 billion diverse individuals
- P Web developers & designers
- j Content writers
- V Businesses and organizations
- b Government & public sector
- l Everyone!

Evolution of WCAG versions



1999

WCAG 1.0

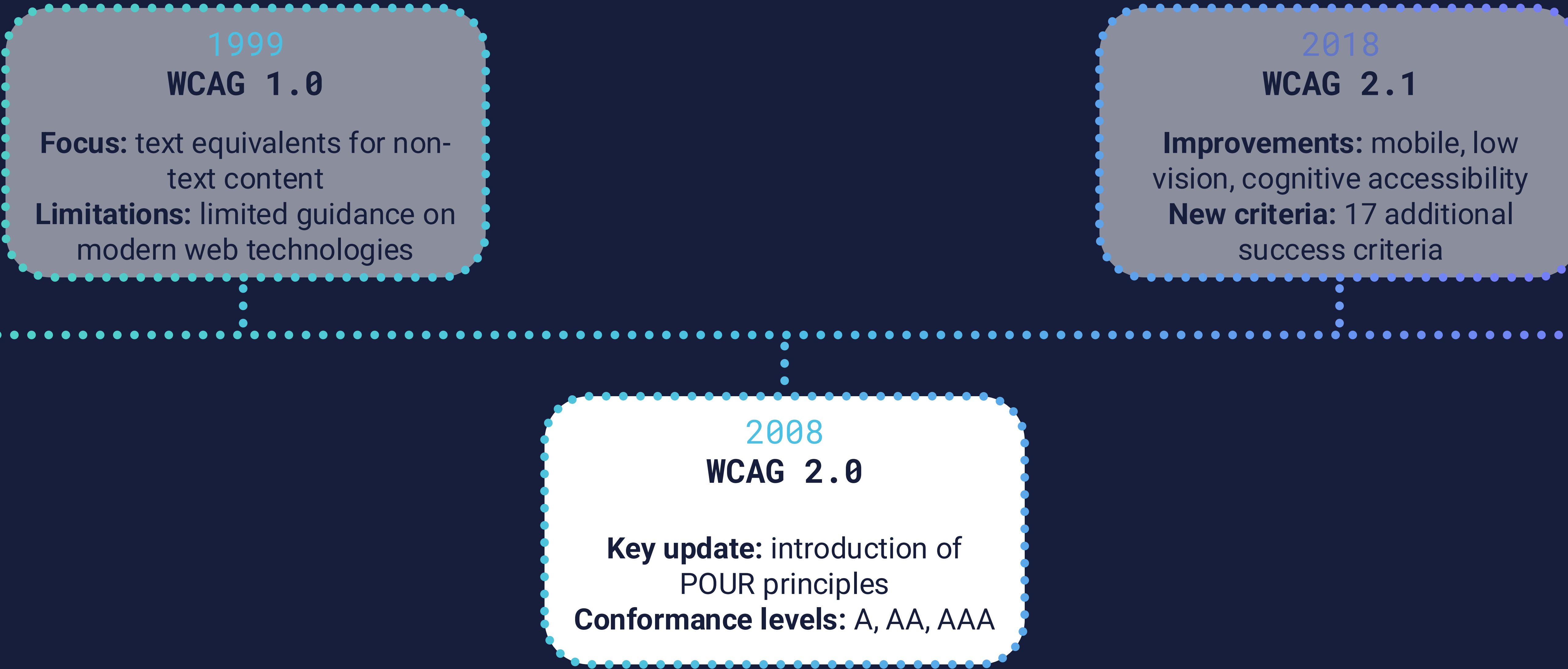
Focus: text equivalents for non-text content
Limitations: limited guidance on modern web technologies

2008

WCAG 2.0

Key update: introduction of POUR principles
Conformance levels: A, AA, AAA

Evolution of WCAG versions



1999
WCAG 1.0

Focus: text equivalents for non-text content
Limitations: limited guidance on modern web technologies

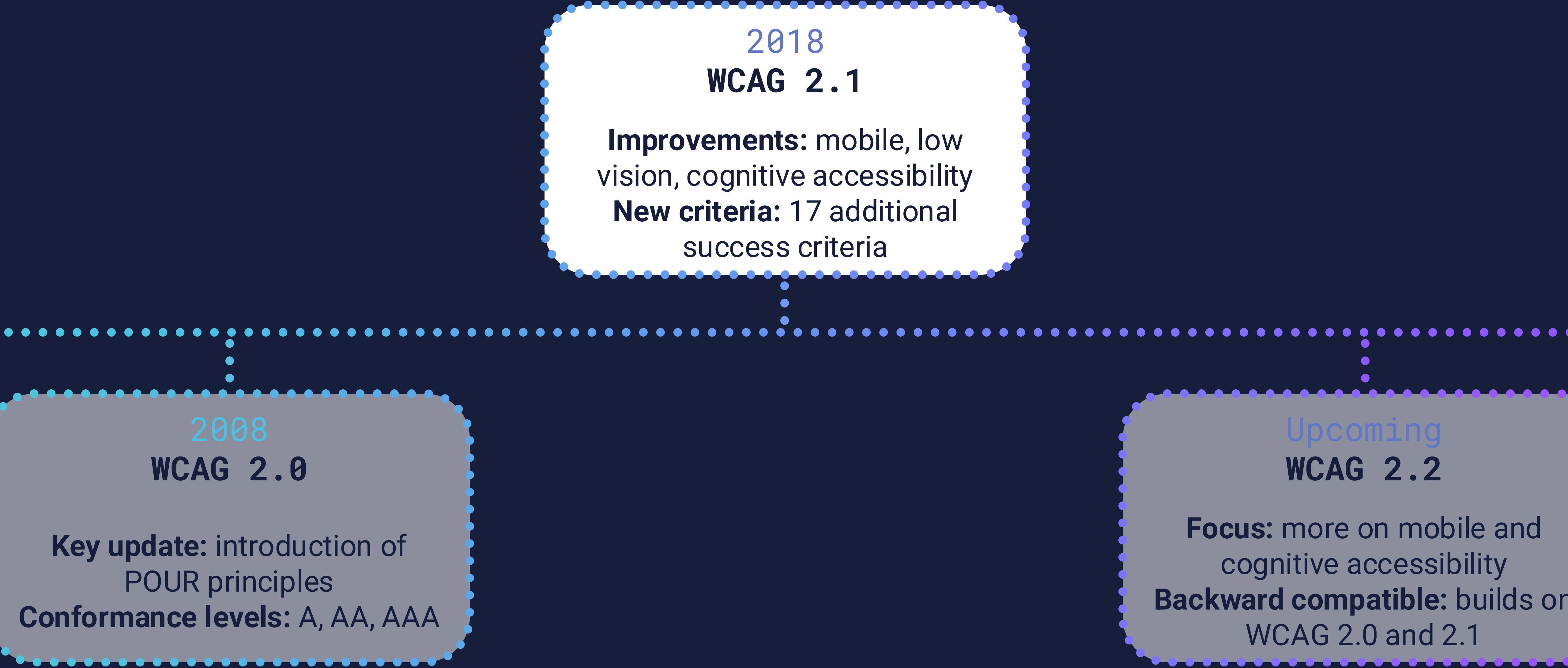
2008
WCAG 2.0

Key update: introduction of POUR principles
Conformance levels: A, AA, AAA

2018
WCAG 2.1

Improvements: mobile, low vision, cognitive accessibility
New criteria: 17 additional success criteria

Evolution of WCAG versions



2008
WCAG 2.0

Key update: introduction of POUR principles
Conformance levels: A, AA, AAA

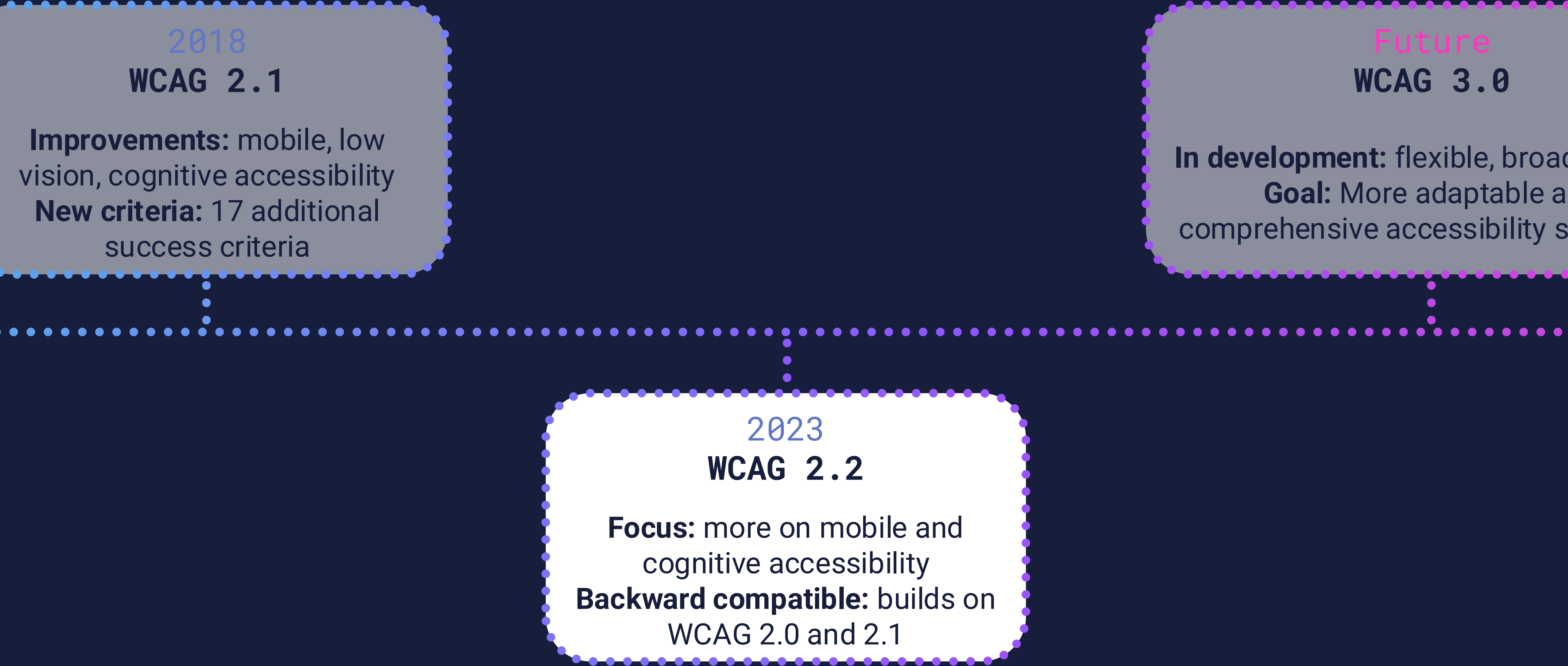
2018
WCAG 2.1

Improvements: mobile, low vision, cognitive accessibility
New criteria: 17 additional success criteria

Upcoming
WCAG 2.2

Focus: more on mobile and cognitive accessibility
Backward compatible: builds on WCAG 2.0 and 2.1

Evolution of WCAG versions



2018

WCAG 2.1

Improvements: mobile, low vision, cognitive accessibility
New criteria: 17 additional success criteria

2023

WCAG 2.2

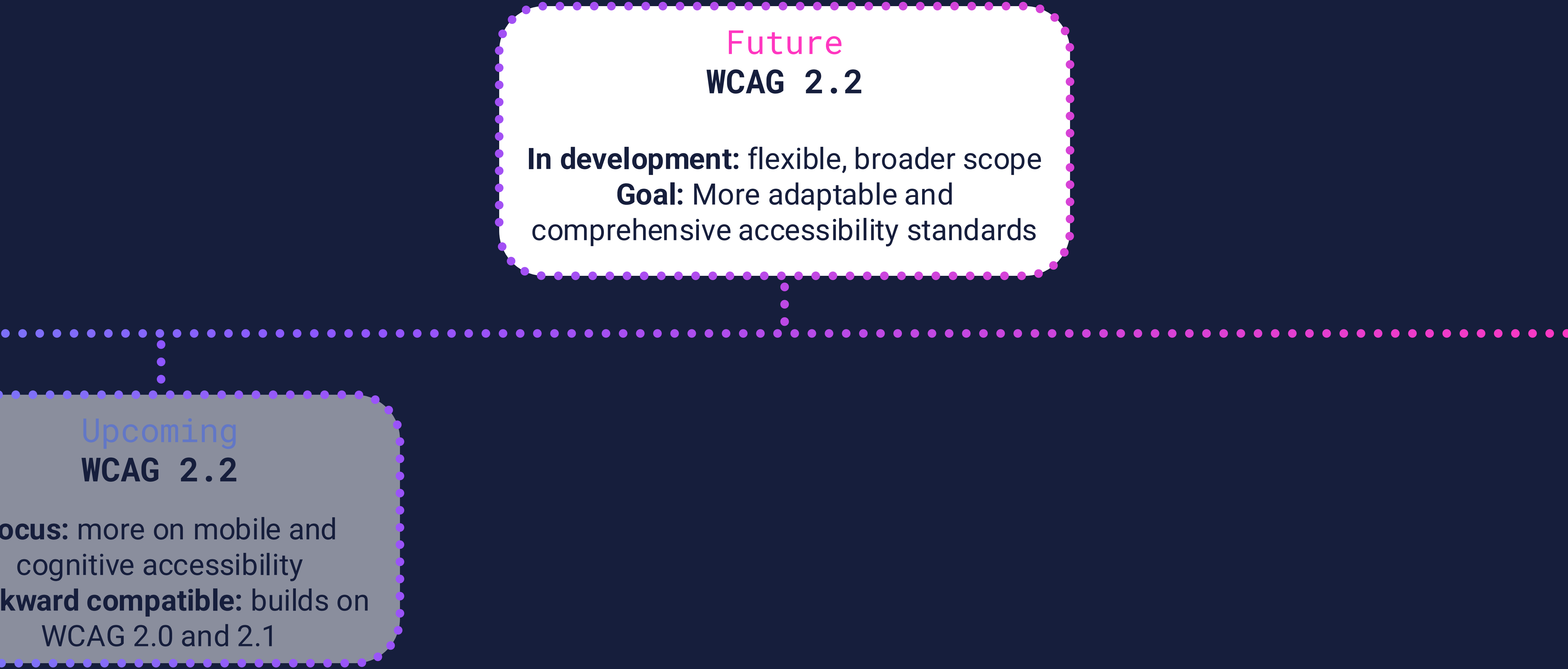
Focus: more on mobile and cognitive accessibility
Backward compatible: builds on WCAG 2.0 and 2.1

Future

WCAG 3.0

In development: flexible, broa
Goal: More adaptable a
comprehensive accessibility s

Evolution of WCAG versions



Future

WCAG 2.2

In development: flexible, broader scope
Goal: More adaptable and comprehensive accessibility standards

Upcoming

WCAG 2.2

ocus: more on mobile and cognitive accessibility
kward compatible: builds on WCAG 2.0 and 2.1

WCAG: 3 levels of conformance

Level A

- Minimum accessibility
- Basic web accessibility features
- Failure to meet this level makes it impossible for some users to access content

Level AA

- Mid-range accessibility
- Deals with the biggest and most common barriers for disabled users
- Most websites aim to meet this level

Level AAA

- Highest accessibility
- Most complex level
- Not all content can conform
- Where possible, this accessibility level should be the goal

{brightboost}

Key WCAG success criteria for developers

g

Text alternatives

For non-text content

K

Navigable content

Logical and consistent navigation

M

Keyboard accessibility

For users who cannot use a mouse

j

Error Identification

Clearly describe errors, helping users understand and correct mistakes

Key WCAG success criteria for designers

q

Contrast and color

Contrast between text and background for readability

A

Consistent layouts

To help users navigate and understand content more easily

B

Responsive design

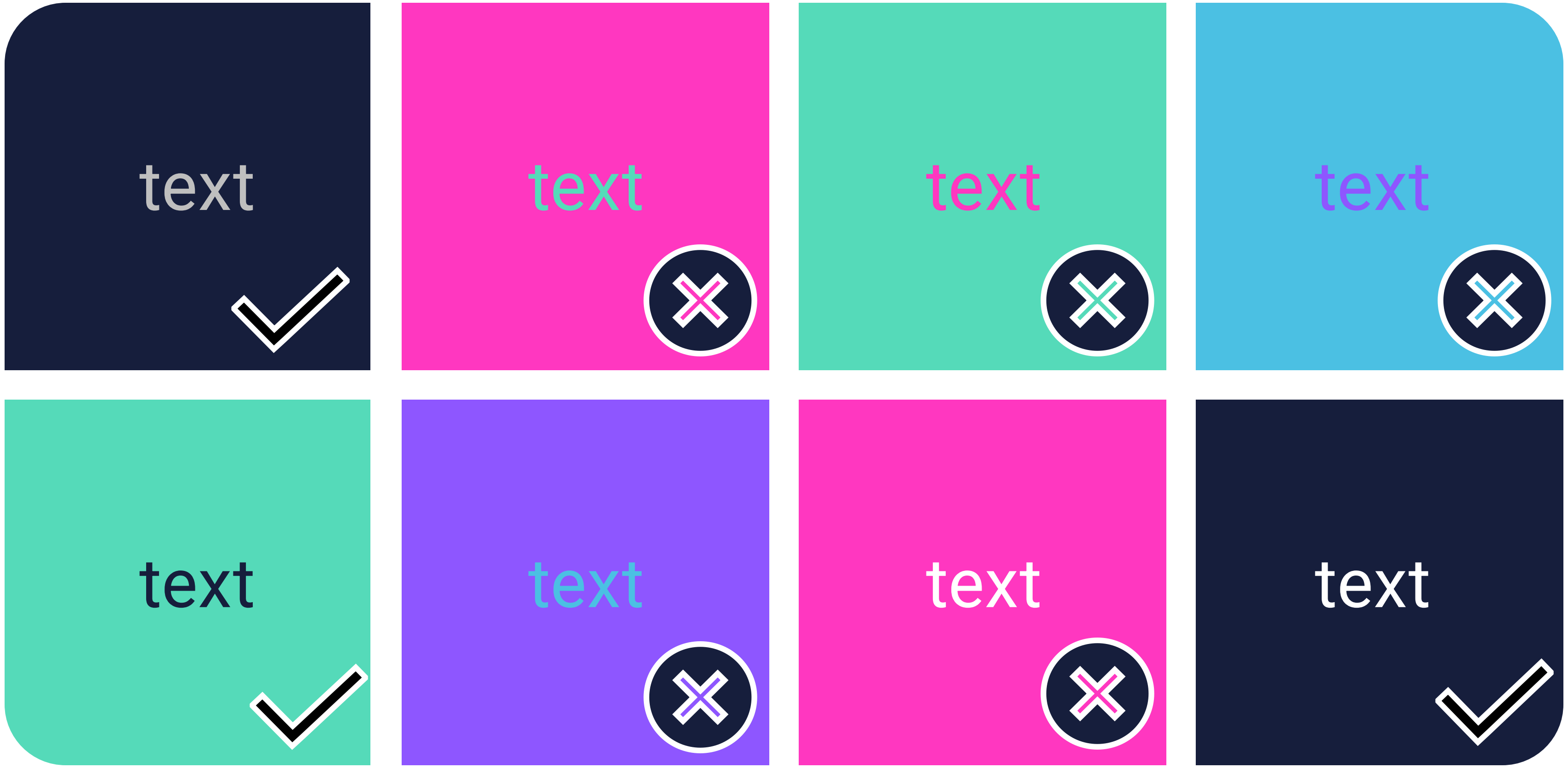
Design for accessibility across devices

d

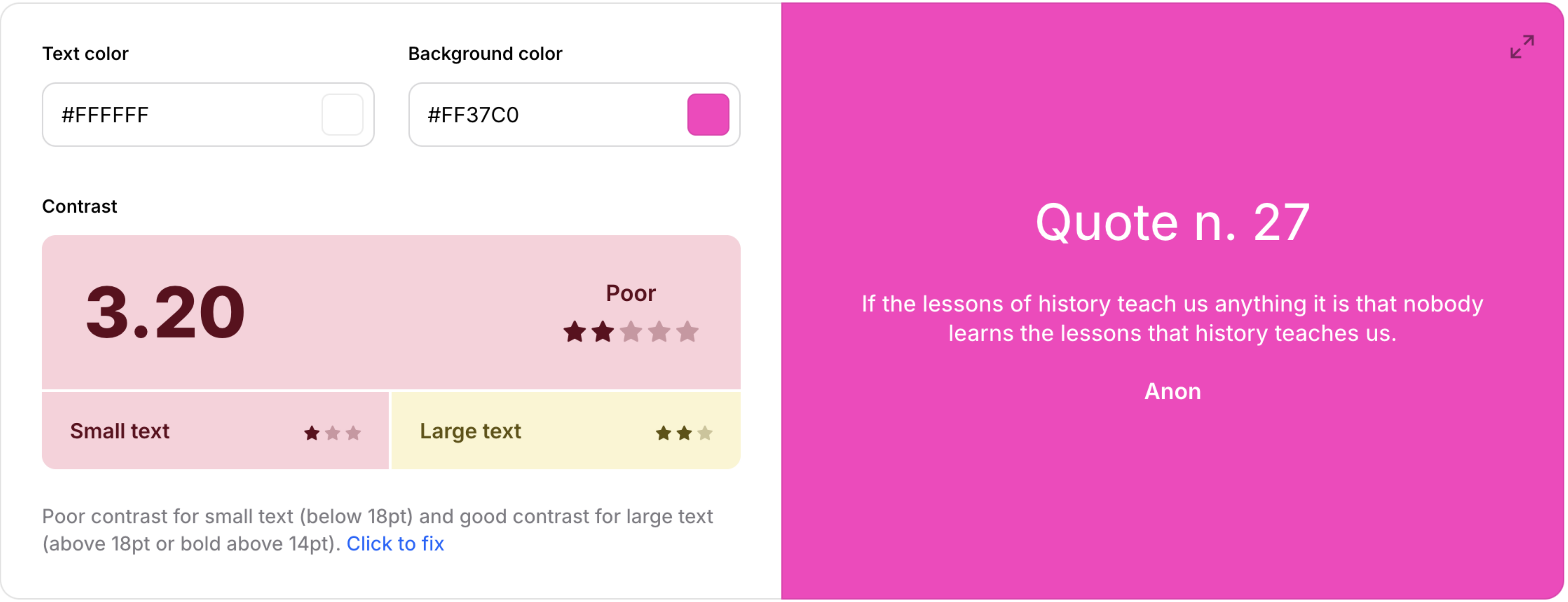
Scalable text

To support users with low vision

Sufficient contrast?



Sufficient contrast?



Text color

#FFFFFF □

Background color

#FF37C0 ■

Contrast

3.20

Poor ★★☆☆☆

Small text ★☆☆

Large text ★★☆

Poor contrast for small text (below 18pt) and good contrast for large text (above 18pt or bold above 14pt). Click to fix

Quote n. 27

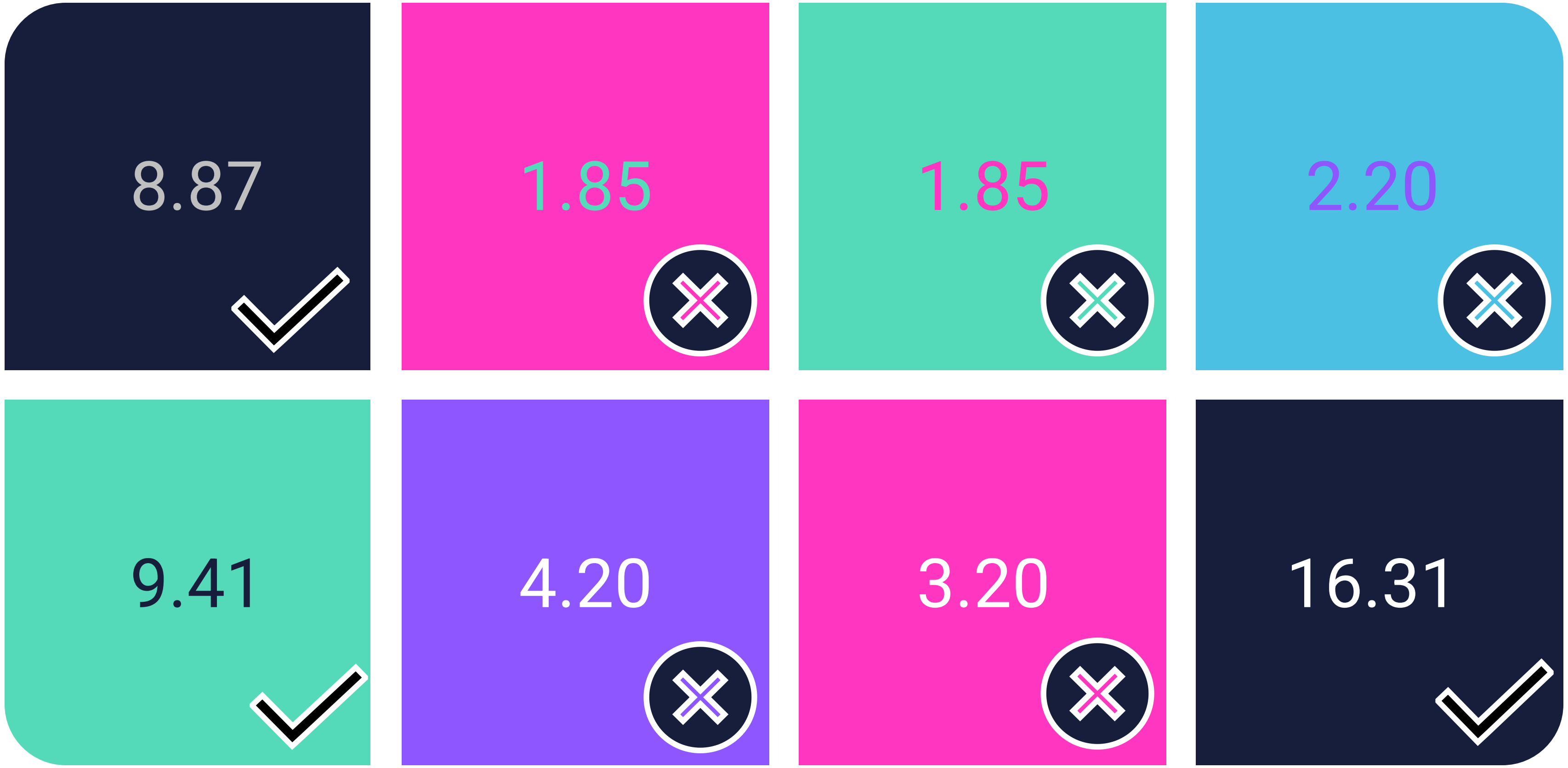
If the lessons of history teach us anything it is that nobody learns the lessons that history teaches us.

Anon

https://coolors.co/contrast-checker/

{brightboost}

Sufficient contrast?



8.87 ✓	1.85 ✗	1.85 ✗	2.20 ✗
9.41 ✓	4.20 ✗	3.20 ✗	16.31 ✓

Current compliance statistics

2023 Analysis:

- 96.3% of homepages fail WCAG 2.0 level AA criteria
- Slight improvement from 97.8% in 2019
- Source: https://webaim.org/projects/million

EU study:

- 85% of websites pass simplified tests
- 93.23% fail in-depth manual evaluations

Market Growth

Digital accessibility software market:

- Projected to reach $893.7 million by 2031
- Reflects increasing demand
- Source: https://www.businessresearchinsights.com/market-reports/website-accessibility-software-market-101575

WCAG additional resources

- For developers: www.wcag.com/developers
- For designers: www.wcag.com/designers
- For content writers: www.wcag.com/authors
- WebAIM’s Million Report: https://webaim.org/projects/million/

POUR principles

Perceivable, Operable, Understandable, Robust

{brightboost}

POUR principles

w

Perceivable

Content must be seen or heard by all users

e

Operable

All functions must be accessible via keyboard and accessible navigation

o

Understandable

Information and user interface should be clear and predictable

k

Robust

Content must work with current and future technologies, including assistive technologies

Perceivable (P)

Digital content must be able to be understood by multiple senses.

Key success criteria:

- Provide **text descriptions** for non-text content like images and videos
- Add **captions** to videos for users with hearing impairments
- Apply sufficient **color contrast** between text and background

Not Perceivable – example

Issue:

- A website uses images without alt text
- Screen readers can't describe the content

Example:

- An infographic that solely relies on visual elements with no text description provided

Impact:

- Visually impaired users miss out on critical information

{brightboost}

Improved Perceivable – fixing the issue

Solution:

- Add descriptive alt text to all images and infographics

Example:

- “A graph showing a 20% increase in sales over the last quarter”

Impact:

- Screen readers can now convey the content to visually impaired users

{brightboost}

Tools and techniques for perceivability

j **Alt text generators:** Microsoft's Image Analysis for alt text

W **Color contrast checkers:** WebAIM Contrast Checker, Stark Plugin for Sketch/Figma

W **Screen reader simulators:** NVDA or VoiceOver to simulate user experience

Operable (O)

Web functionality must be able to be accessed and used by all users regardless of ability.

Key success criteria:

- All functionalities must be available using only a **keyboard**
- Users should have **enough time** to read and interact with content

Not Operable - example

Issue:

- A website requires a mouse for navigation
- No keyboard access

Example:

- A drop-down menu that cannot be accessed with the keyboard

Impact:

- Users relying on keyboard navigation cannot interact with the content

Improved Operable – fixing the issue

Solution:

- Enable keyboard navigation
- Make all interactive elements keyboard accessible

- Users can now navigate menus using the tab key and select items with the enter key

Impact:

- The website becomes accessible to users who rely on keyboards or other assistive technologies for navigation

{brightboost}

Tools and techniques for operability

Keyboard navigation testing: Microsoft’s Image Analysis for alt text

W

Screen reader testing: JAWS, VO or NVDA for comprehensive operability checks

e

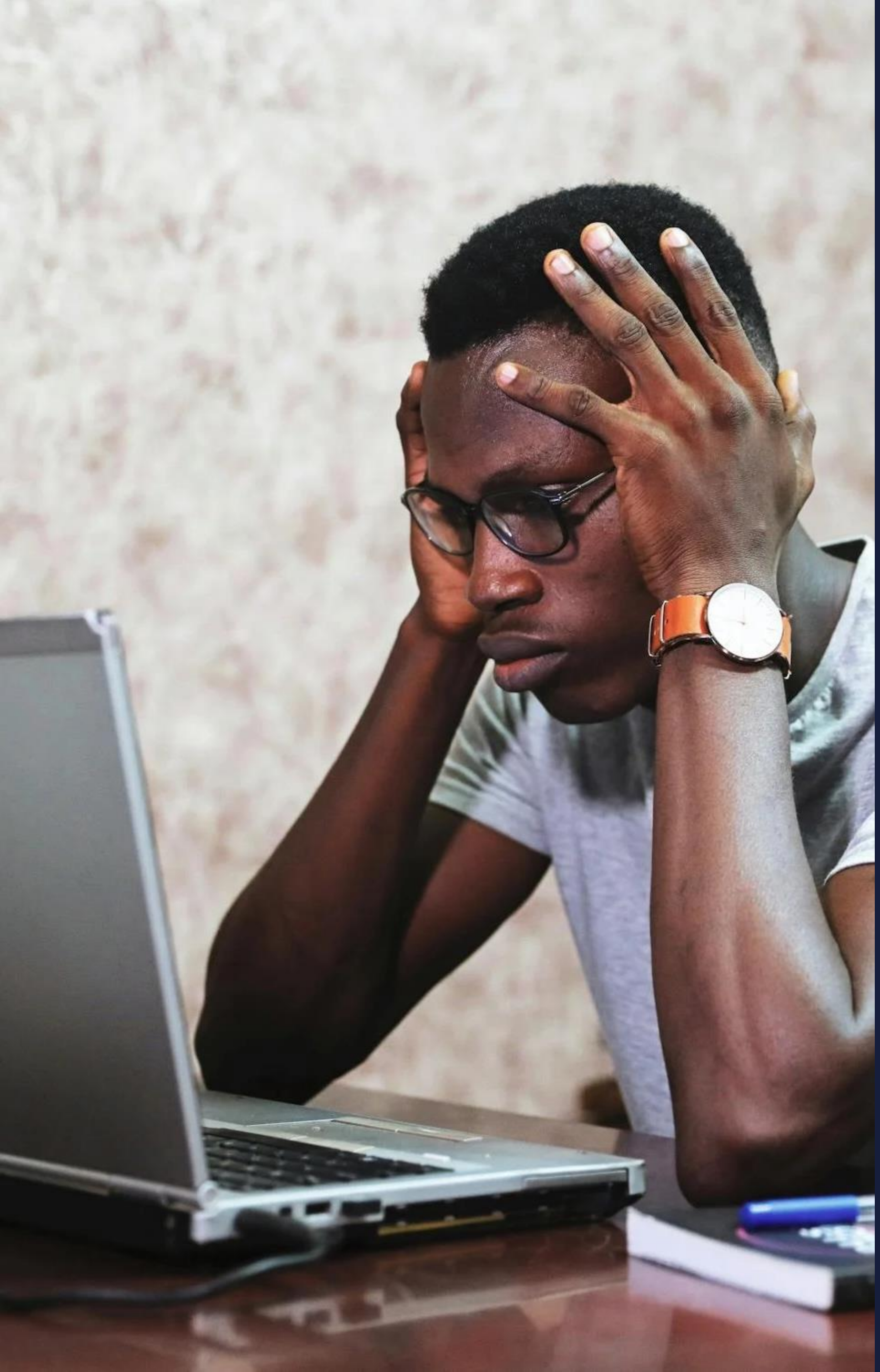
Focus indicators: Use the “focus-visible” CSS property for keyboard navigation visibility

Understandable (U)

All content must be able to be understood by users. The operation of user interfaces needs to be understandable as well.

Key success criteria:

- Use **simple** and straightforward **language**
- Webpages must appear and operate in **predictable** ways



Not Understandable - example

Issue:

- A website uses complex jargon
- And inconsistent navigation

Example:

- Different sections of a site have varying menu structures

Impact:

- Users struggle to understand content and predict how to navigate the site

Improved Understandable - example

Solution:

- Simplify language
- And inconsistent navigation

Example:

- Consistent header menus across all pages: “Home”, “About”, “Contact”, etc.

Impact:

- Users can Easily understand the content and navigate the site

{brightboost}

Tools and techniques for understandability

D

Readability analysers: Hemingway App, Grammarly, or Readable.io to verify content is clear and simple

I

User testing platforms: UsabilityHub or UserTesting to gather feedback on site navigation and content clarity

Form validators: Use tools like Parsley.js to provide clear error messages and guidance in forms

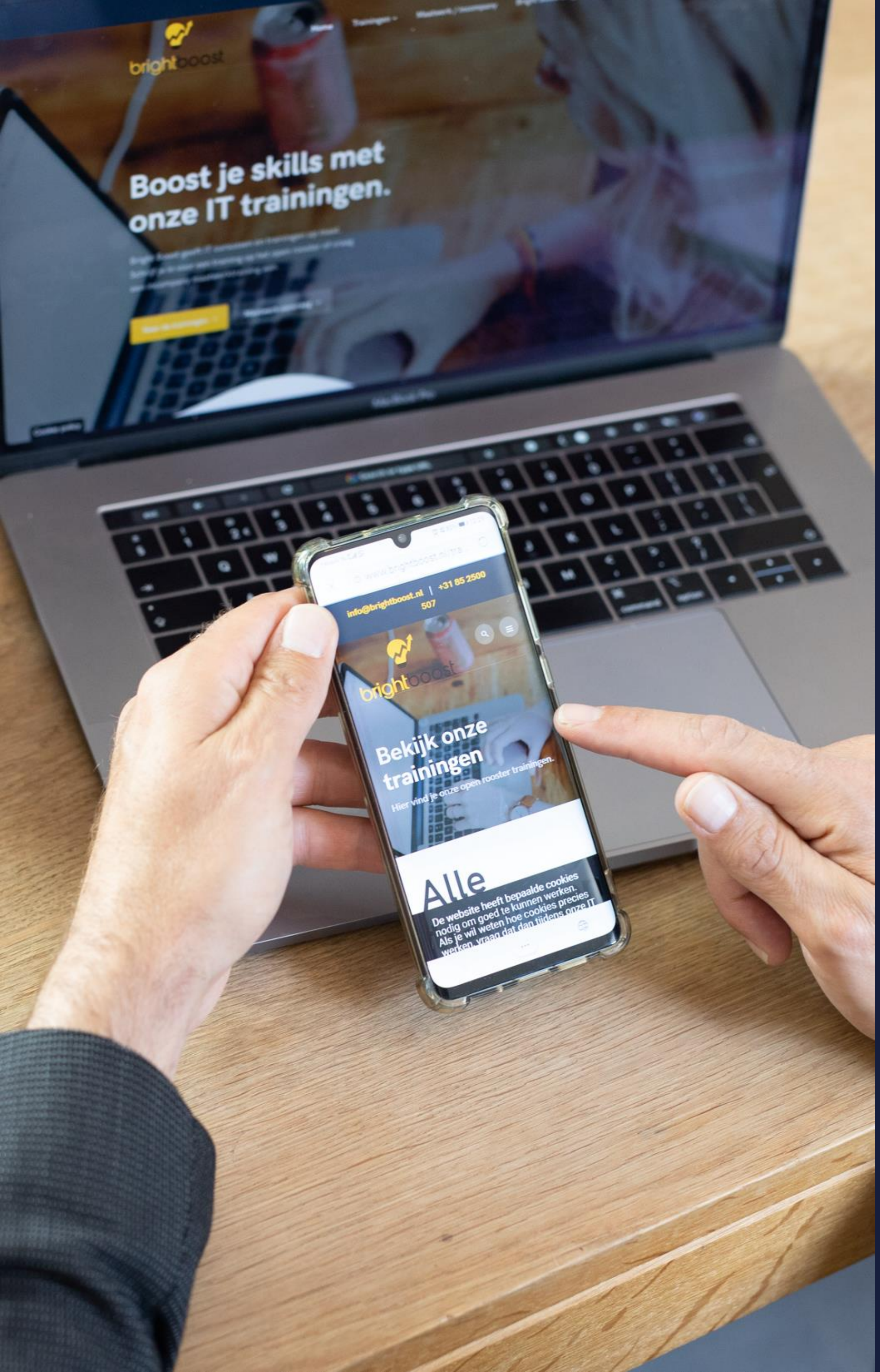
{brightboost}

Robust (R)

Digital content must be able to be interpreted by assistive technologies and other user agents.

Key success criteria:

- Use valid HTML/CSS to ensure **compatibility** with various browsers and assistive technologies
- Implement **ARIA** (Accessible Rich Internet Applications) attributes to improve accessibility


brightboost
Boost je skills met onze IT trainingen.
info@brightboost.nl | +31 85 2500 507
brightboost
Bekijk onze trainingen
Hier vind je onze open rooster trainingen.
Alle
De website heeft bepaalde cookies nodig om goed te kunnen werken.

Not Robust - example

Issue:

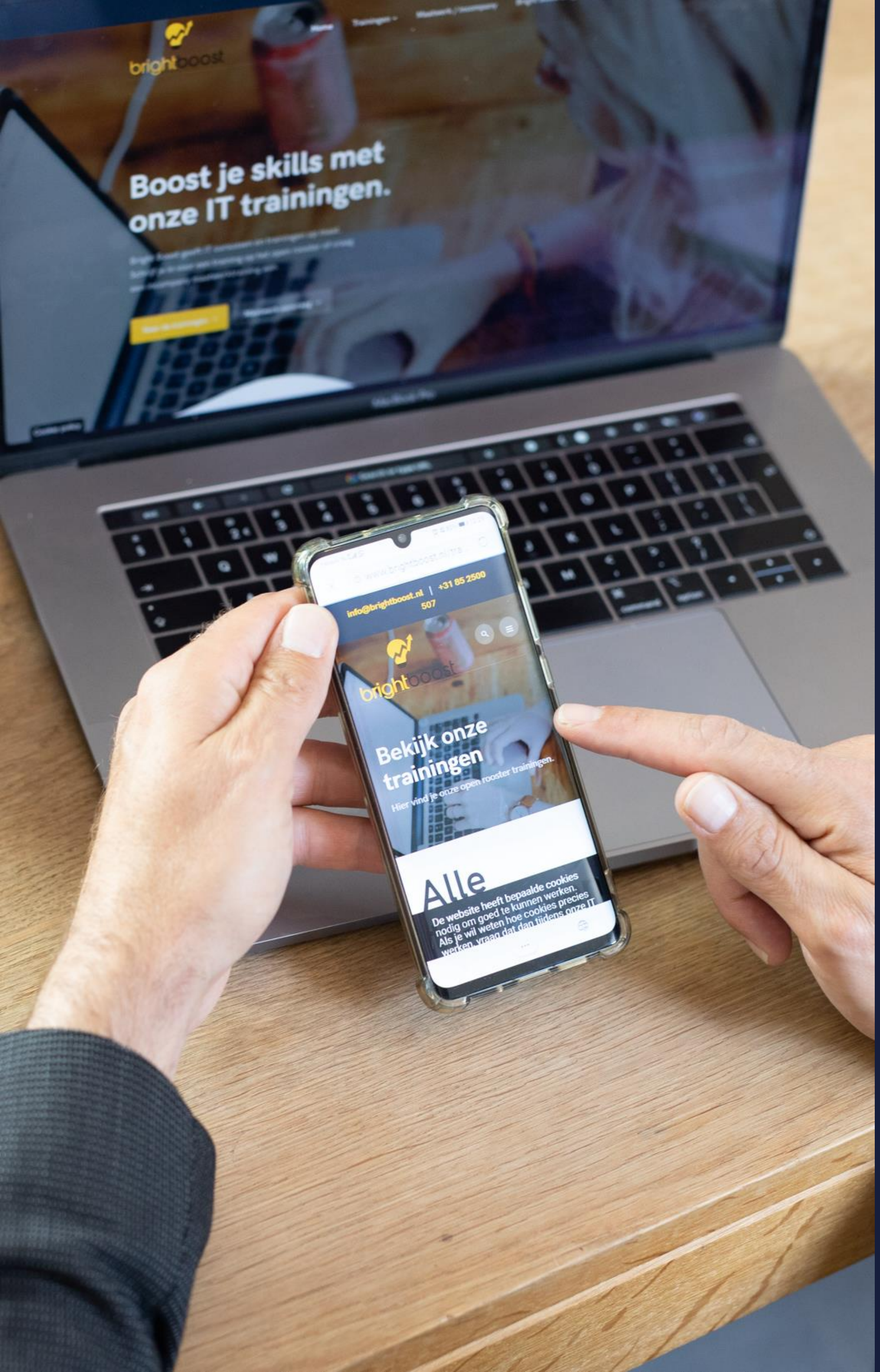
- A website uses outdated or invalid HTML
- Causing issues with assistive technologies

Example:

- A site that fails to render correctly on certain browsers or isn’t accessible via screen readers due to missing ARIA roles.

Impact:

- Users relying on assistive technologies cannot fully interact with or access the content.



Improved Robust – fixing the issue

Solution:

- Update the code to valid HTML/CSS
- Include ARIA attributes where necessary

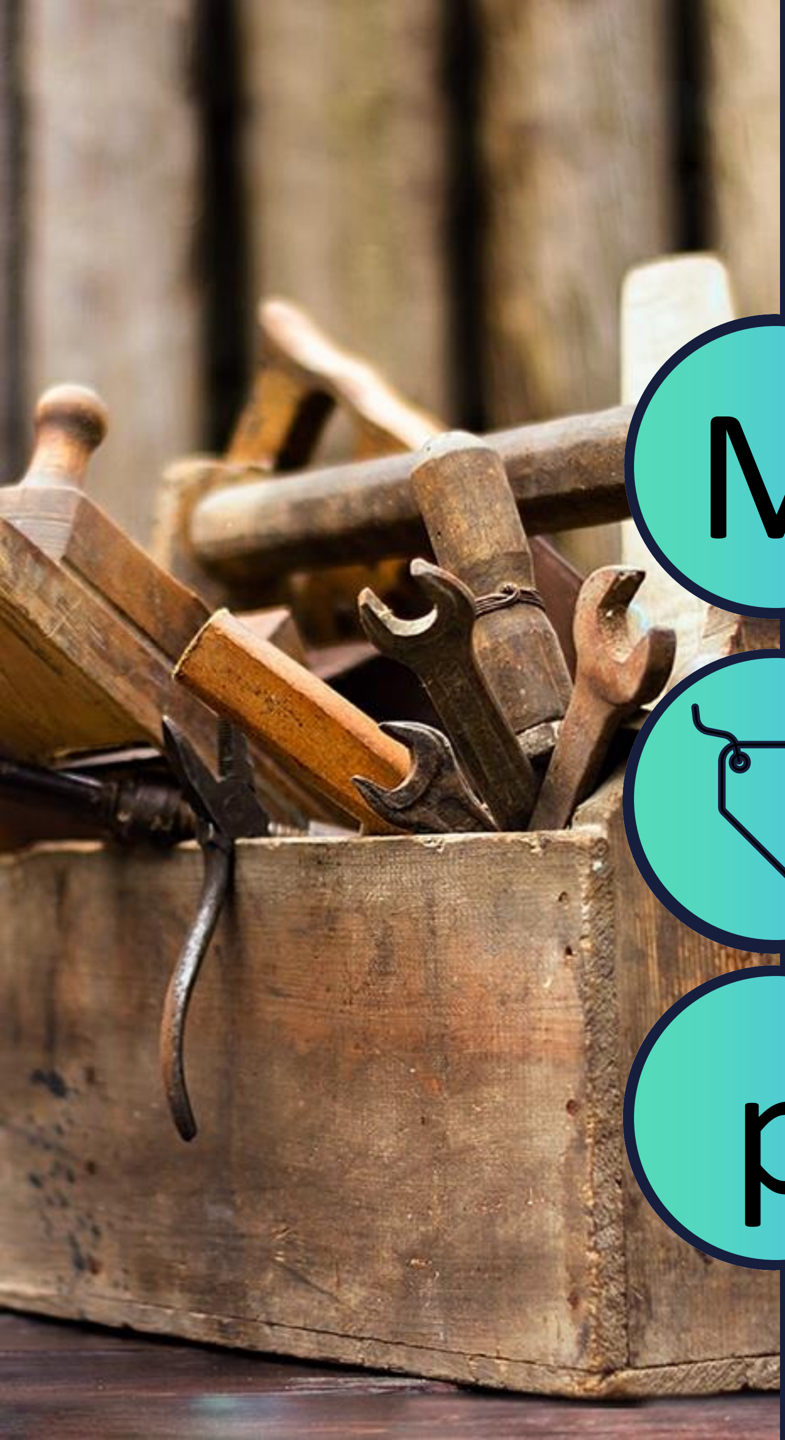
Example:

- Adding ARIA roles like "navigation" or "main" to key sections of the site

Impact:

- The website is accessible across various browsers and works well with assistive technologies

{brightboost}



Tools and techniques for robustness

M **W3C HTML validator:** Checks whether the site’s code is valid and robust

ARIA validator: To check for correct use of ARIA roles and attributes

p **Cross-browser testing:** Use tools like BrowserStack or CrossBrowserTesting to check compatibility across multiple browsers and devices

{brightboost}

Exercise: find examples

search-examples.md

Must haves – Non-UI

To follow WCAG guidelines

S

J

T

U

Contact methods

- Accessible contact methods for support or information
- Easy to find and use
- Accommodate different disabilities
- Multiple channels
  - Phone
  - Email
  - Live chat
  - Etc.

{brightboost}

Accessibility Statement

Public declaration of website’s commitment to accessibility, containing:

- Steps taken for accessibility
- Level of WCAG compliance
- How to report accessibility issues

Demonstrates transparency and accountability

Obligated for governmental organization in the NL based on ”Wdo”)

Transcripts/Subtitles for videos

Transcript

- Text record of all spoken content in a video

Subtitles

- Timed text that appears on the video, synchronized with audio

Both make content accessible for:

- users with hearing impairments
- those who prefer reading over listening

{brightboost}

Clear and understandable text

- Content should be written in plain language
- Avoid complex jargon
- Keep sentences concise

- Helps make information accessible to users of literacy levels and cognitive abilities

NOS

JOURNAAL IN MAKKELIJKE TAAL

Screen readers

- Convert text and elements on a screen to
  - Speech
  - Braille
- Essential for visual impaired users
- Must support screen reader navigation

{brightboost}

Must haves - UI

To follow WCAG guidelines

Alt text for images

- Descriptive text for images
- Crucial for conveying visual information
- Should be concise but descriptive

{brightboost}

Semantic HTML

- Use HTML elements that convey meaning
  - `<header>`
  - `<article>`
  - etc.
- Helps structure content for accessibility tools
- Enhances readability and organization of web content

Semantic HTML

```
<header>
  <h1>Welcome to my website</h1>
</header>
<nav>
  <ul>
    <li><a href=“#home”>Home</a></li>
    <li><a href=“#about”>About</a></li>
  </ul>
</nav>
<article>
  <h2>Latest news</h2>
  <p>...</p>
</article>
```

- Semantic elements convey meaning and structure of content
- Improves keyboard navigation
- Clarifies content roles
- SEO

```
<div class=“header”>
  <h1>Welcome to my website</h1>
</div>
<div class=“nav”>
  <ul>
    <li><a href="#home">Home</a></li>
    <li><a href="#about">About</a></li>
  </ul>
</div>
<div class=“content”>
  <h2>Latest news</h2>
  <p>...</p>
</div>
```

Non-semantic HTML

- Decreased accessibility
- Poor content structure
- Lack of meaning
- No clear roles

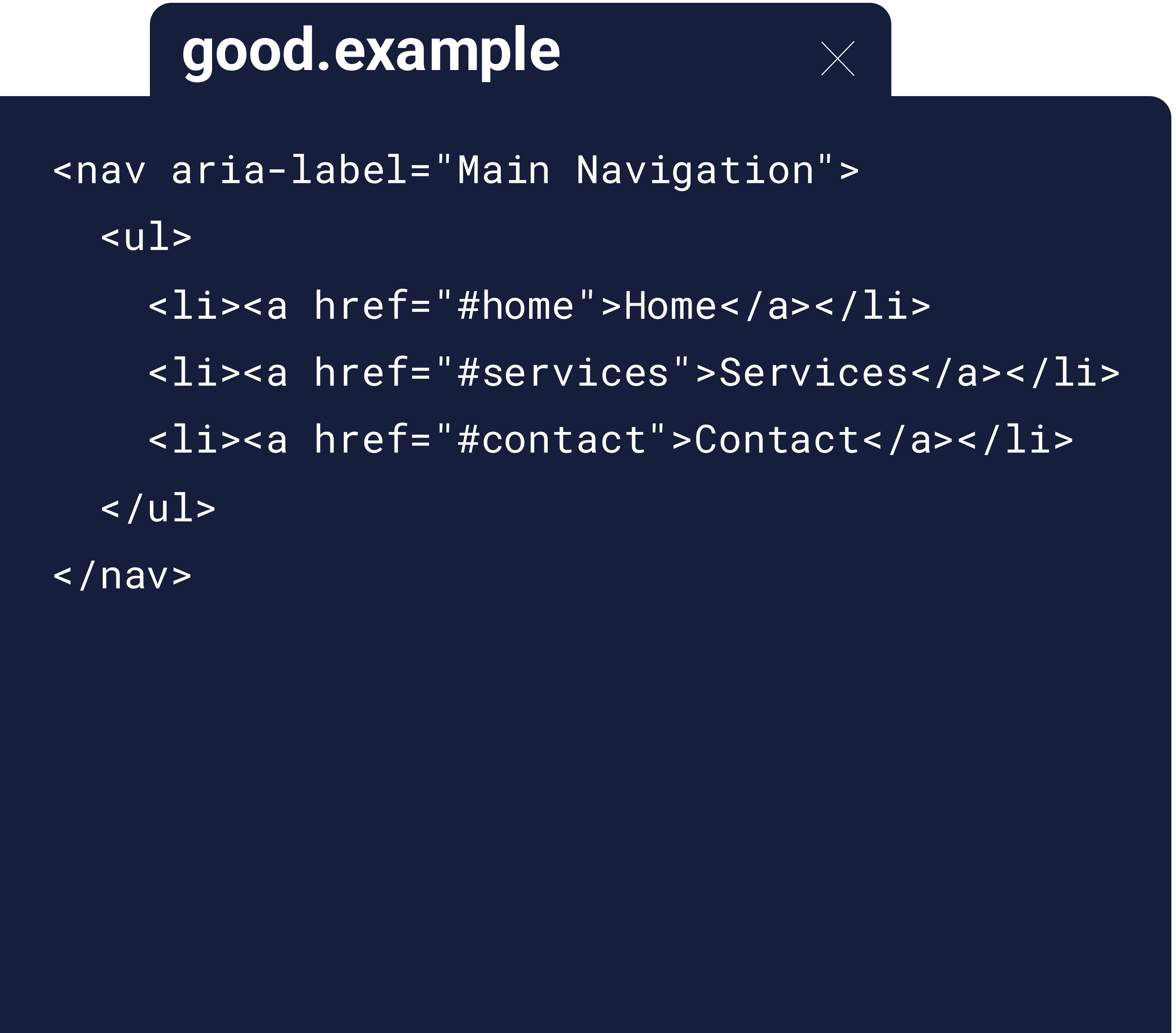
Use of colors

- Sufficient contrast between text and background
- Color coding
  - Don’t rely on color alone to convey information
  - Use additional indicators like text labels or icons

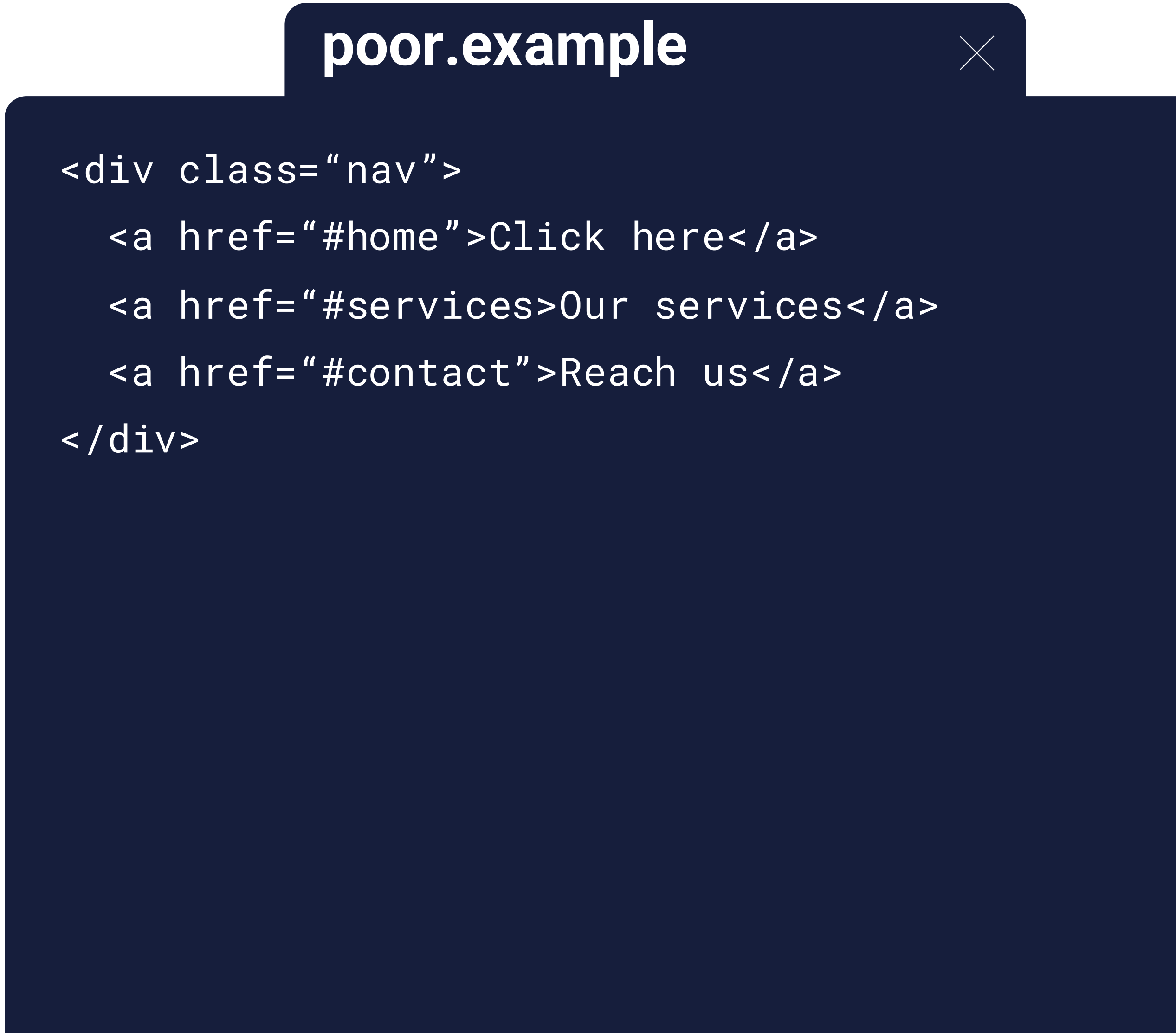
Accessible navigation

- Navigation must be clear and intuitive
- Properly label links and use ARIA landmarks
- Avoid hidden or overly complex menus

Accessible navigation



```
<nav aria-label="Main Navigation">
  <ul>
    <li><a href="#home">Home</a></li>
    <li><a href="#services">Services</a></li>
    <li><a href="#contact">Contact</a></li>
  </ul>
</nav>
```



```
<div class=“nav”>
  <a href=“#home”>Click here</a>
  <a href=“#services>Our services</a>
  <a href=“#contact”>Reach us</a>
</div>
```

Keyboard-friendly forms and navigation

- All interactive elements must be accessible using keyboard
- Use the correct tab index for form fields
- Apply visible focus indicators

Keyboard-friendly forms and navigation

good.example

```
<input type="text" name="firstName"
tabindex="1" aria-label="First Name”>


<input type="text" name="lastName"
tabindex="2" aria-label="Last Name”>


<button type="submit"
tabindex="3">Submit</button>
```

poor.example

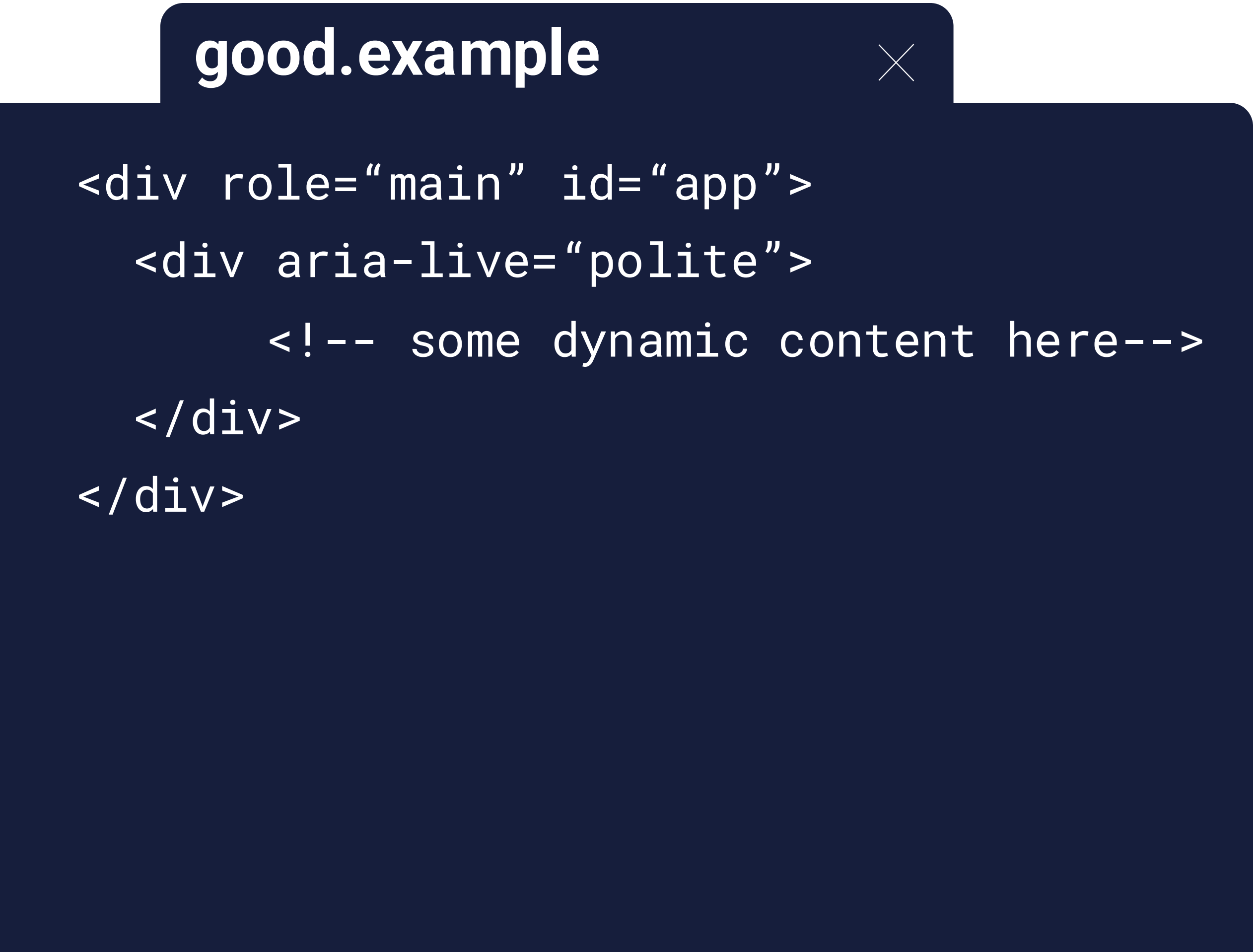
```
<input type="text" name=“firstName">


<input type="text" name=“lastName">


<button type="submit">Submit</button>
```

SPAs and accessibility

- Single Page Applications
- SPAs dynamically load content on a single page
- Use ARIA live regions to announce change
- Ensure focus management for new content

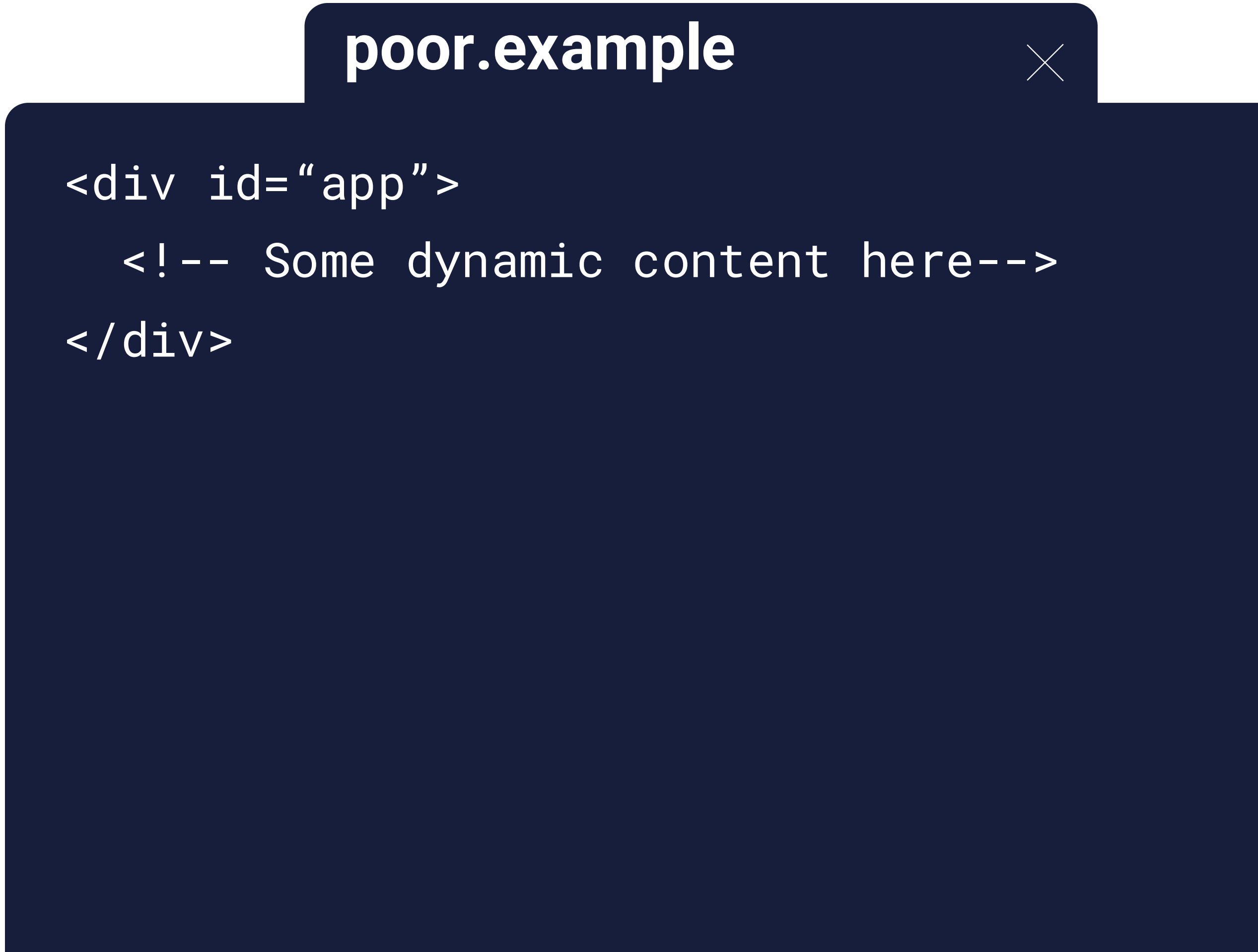
SPA accessibility

SPA with and without ARIA live regions



good.example

```
<div role=“main” id=“app”>
  <div aria-live=“polite”>
       <!-- some dynamic content here-->
  </div>
</div>
```



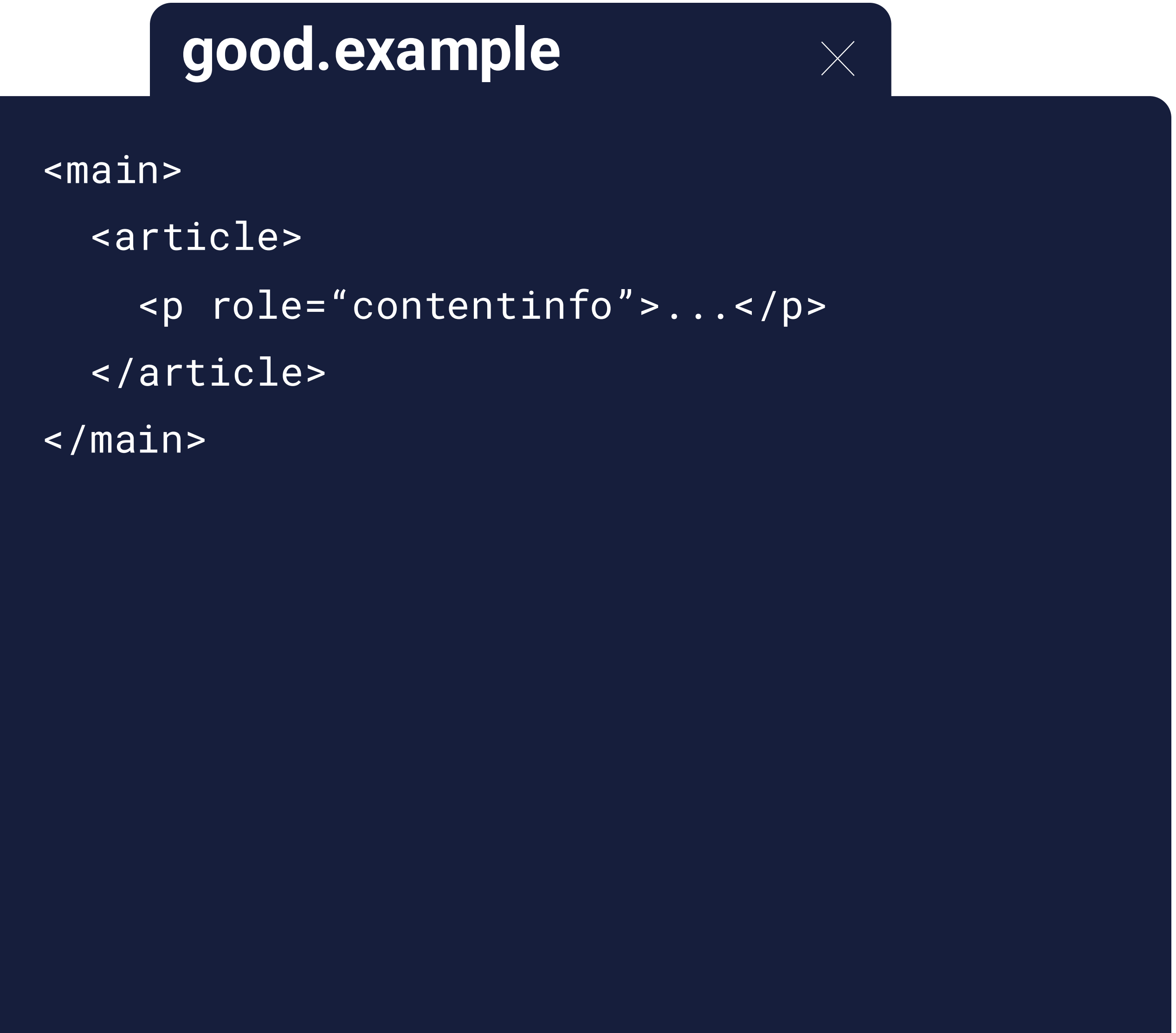
poor.example

```
<div id=“app”>
  <!-- Some dynamic content here-->
</div>
```

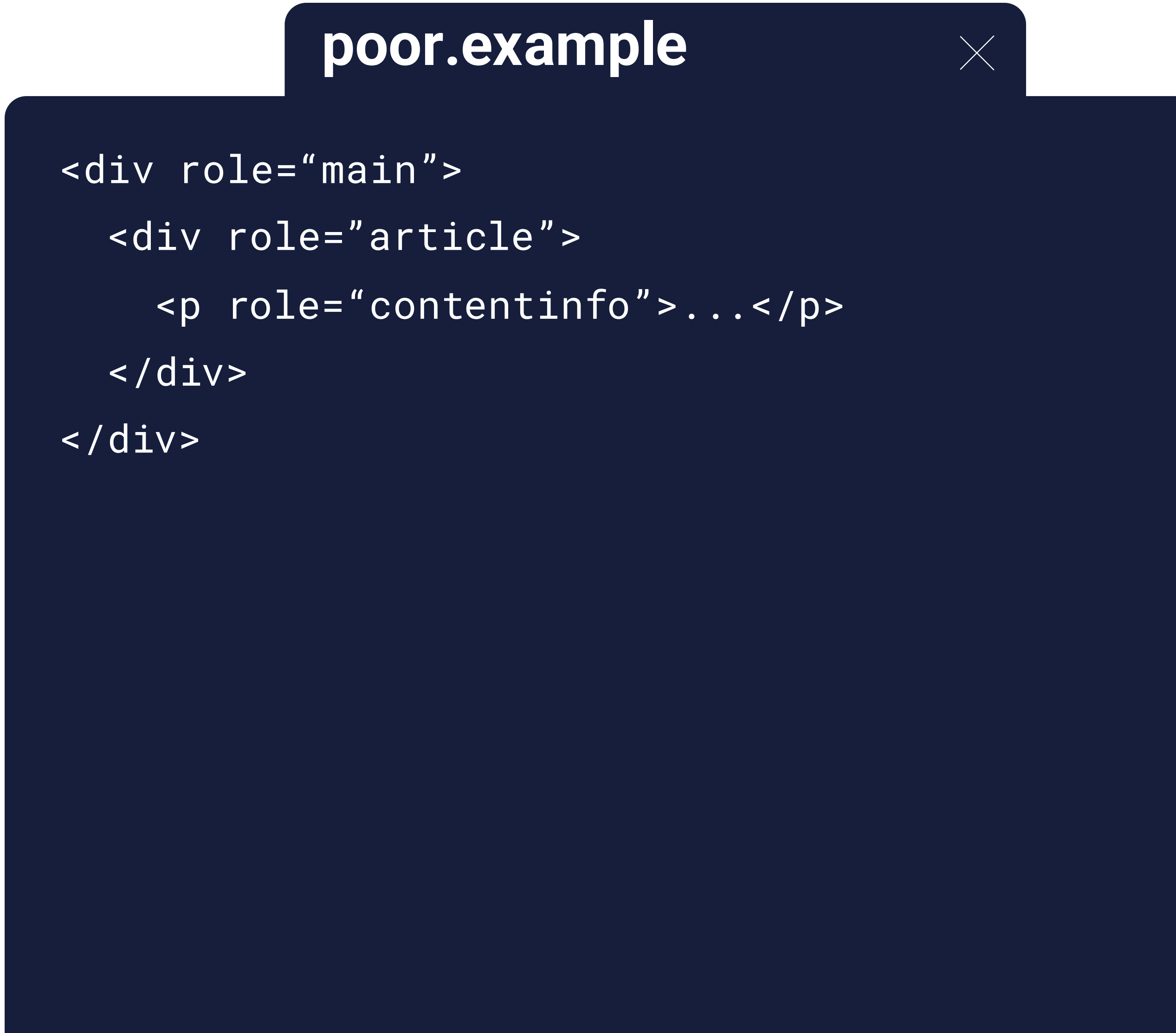
WAI-ARIA

- Accessible Rich Internet Applications
- Use ARIA roles, states, and properties to support assistive technologies
- Dynamic content
- Avoid overusing ARIA where native HTML can suffice

Proper ARIA role usage



```
<main>
  <article>
    <p role=“contentinfo”>...</p>
  </article>
</main>
```



```
<div role=“main”>
  <div role=”article”>
    <p role=“contentinfo”>...</p>
  </div>
</div>
```

Exercise: aria

aria.md

Responsive design and font flexibility

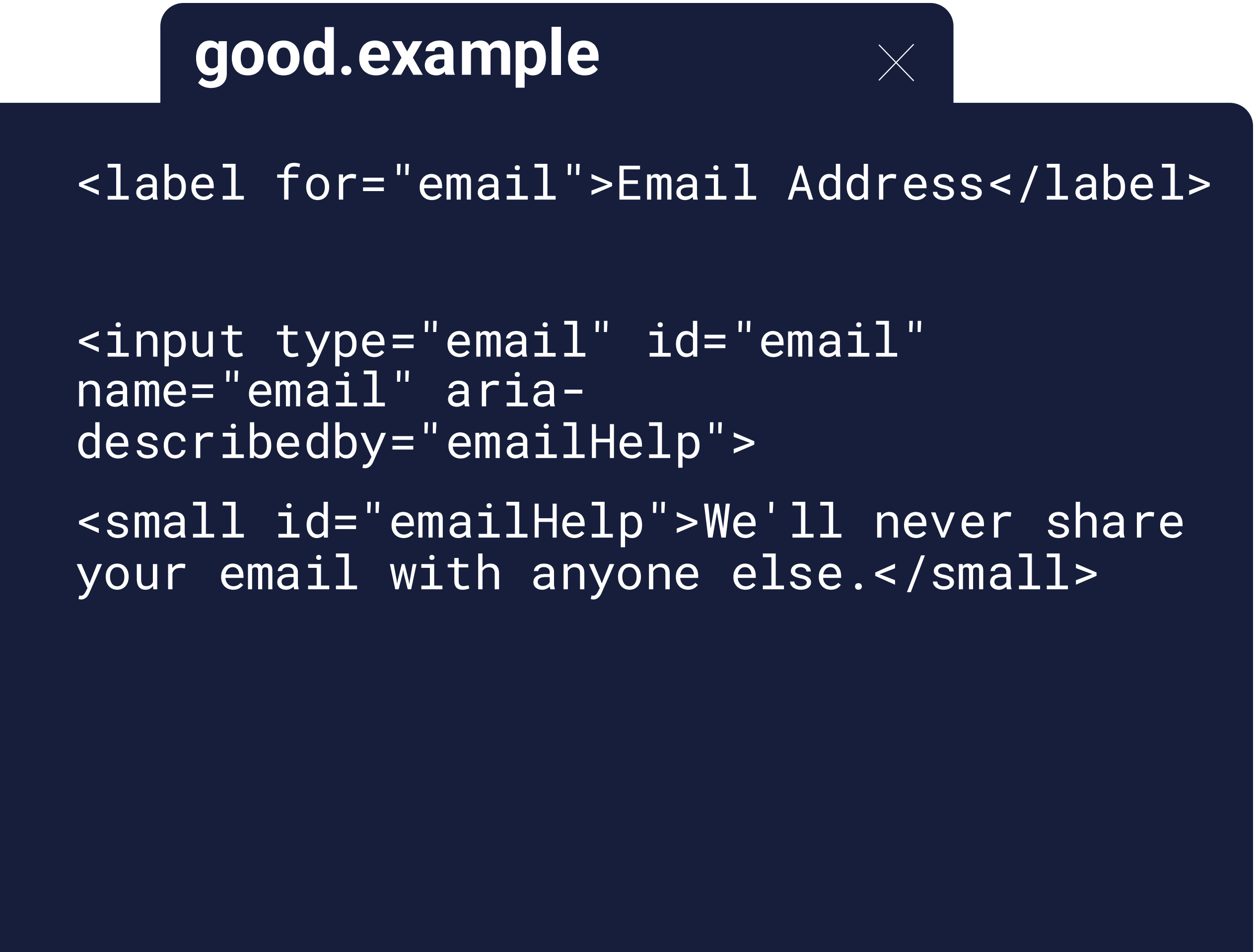
- Scalability across devices
- Use relative font sizes: em, rem instead of fixed sizes
- Enable users to adjust settings

Accessible forms

- Easy to navigate
- Clear labels and instructions
- Provide errors that screen readers can handle
- Make sure all fields are keyboard accessible

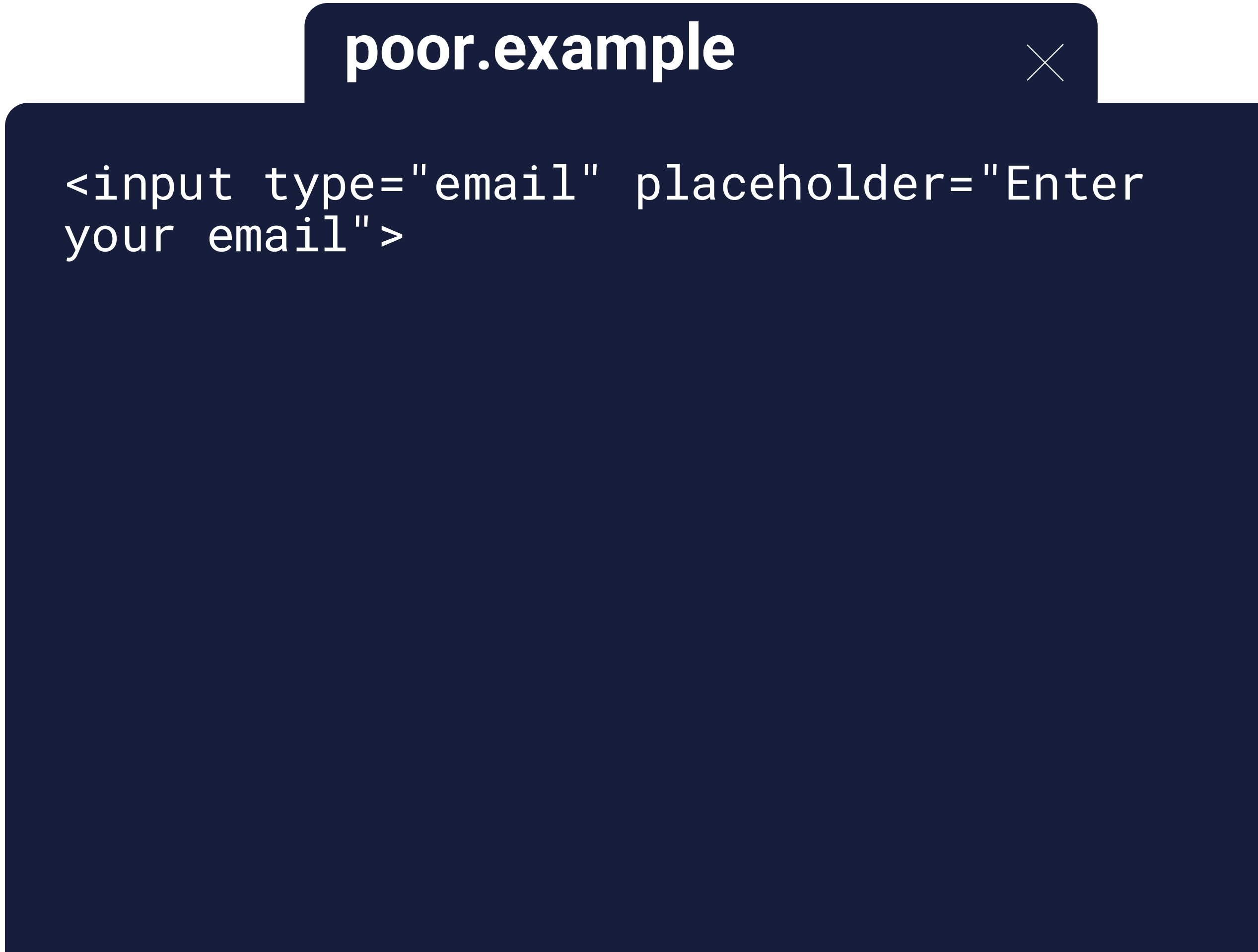
Labeled form fields vs unlabeled form fields

Proper labels and descriptions make the form accessible to screen readers and assistive technologies.



good.example

```
<label for="email">Email Address</label>


<input type="email" id="email"
name="email" aria-
describedby="emailHelp">

<small id="emailHelp">We'll never share
your email with anyone else.</small>
```



poor.example

```
<input type="email" placeholder="Enter
your email">
```

Time limit accessibility

- Allow users to extend or remove time limits
- Provide warnings before time expires
- Give users enough time to complete tasks

{brightboost}

Use of focus indicators

- Visual cues that show which element is selected
- Essential for users that navigate by keyboard
- Must be clearly visible and consistent

Use of focus indicators

A clear and visible focus indicator makes sure that users with sight navigating by keyboard can easily see which element is currently selected.

Good-example.css

```
button:focus {
    outline: 2px solid #0056b3;
    outline-offset: 2px;
}
```

Poor-example.css

```
button:focus {
    outline: none;
}
```

Avoiding automatic actions

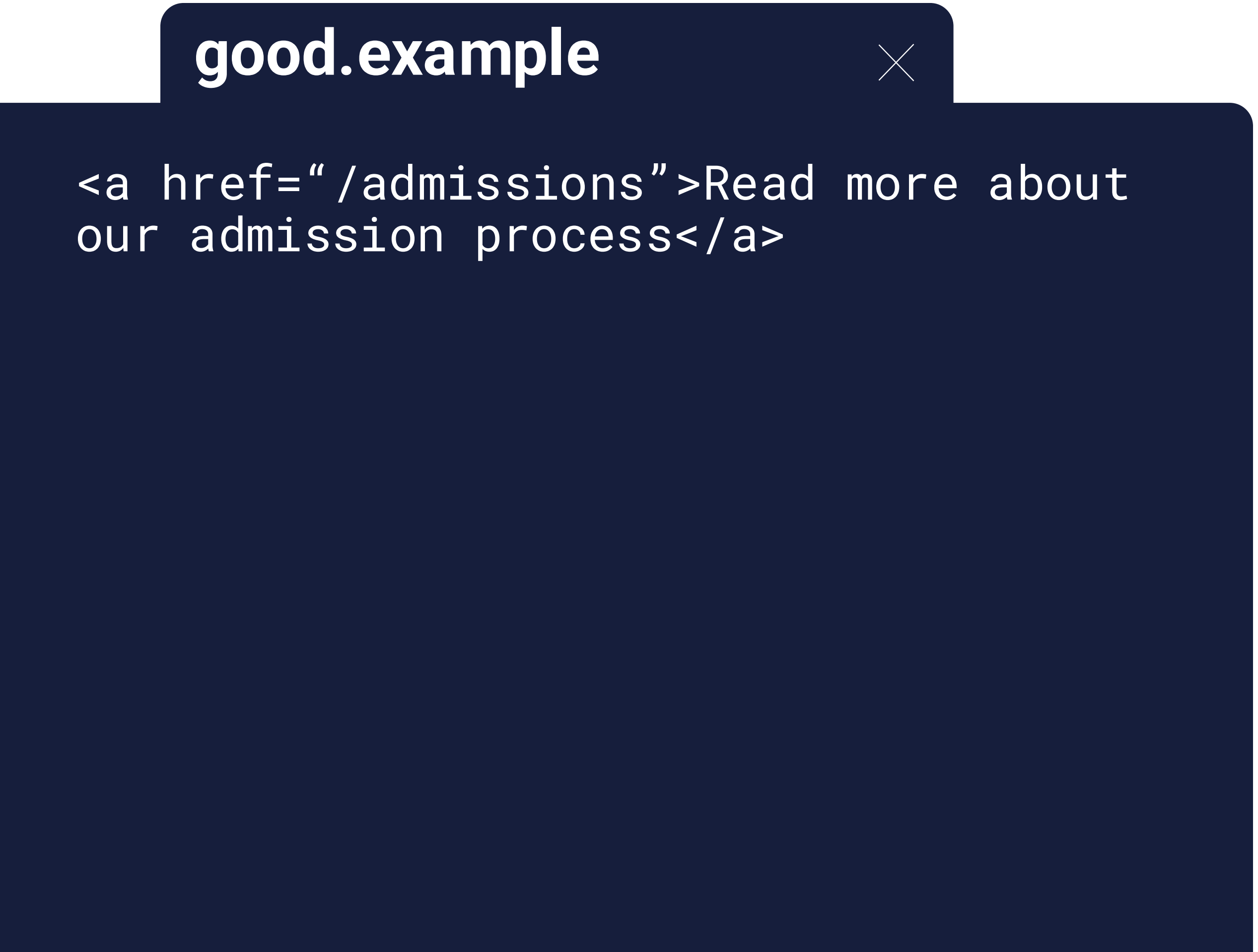
- Avoid actions that occur without user input
  - Auto-rotating carousels
- Provide controls to pause or stop automatic actions
- Prevents disorientation
- Improves user control

Descriptive link text

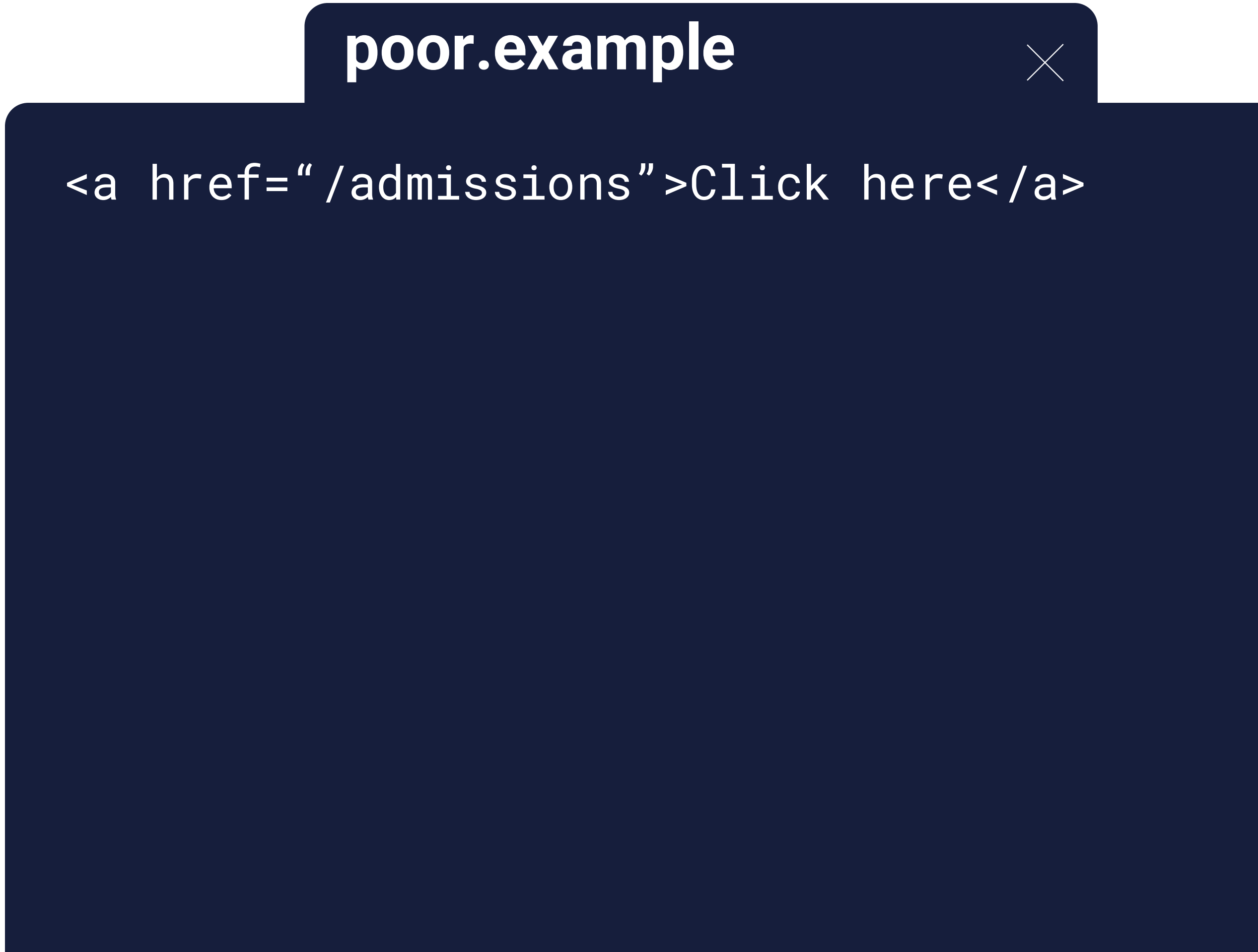
- Links should clearly describe their destination or action
- Avoid generic text like “Click here!”
- Improves context for screen reader users

Descriptive link texts

Descriptive link texts vs generic link texts.



```
<a href=“/admissions”>Read more about
our admission process</a>
```

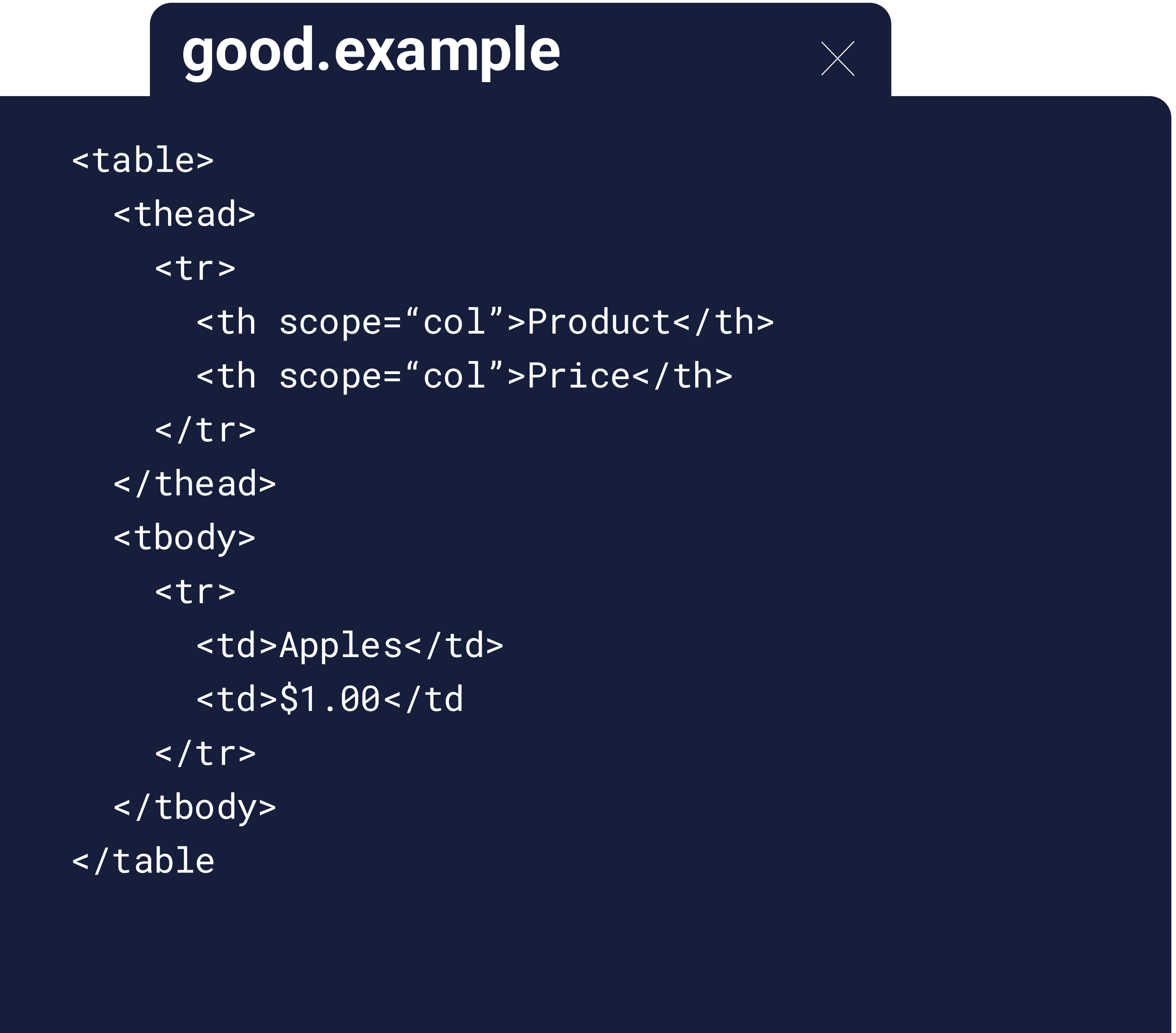


```
<a href=“/admissions”>Click here</a>
```

Table accessibility

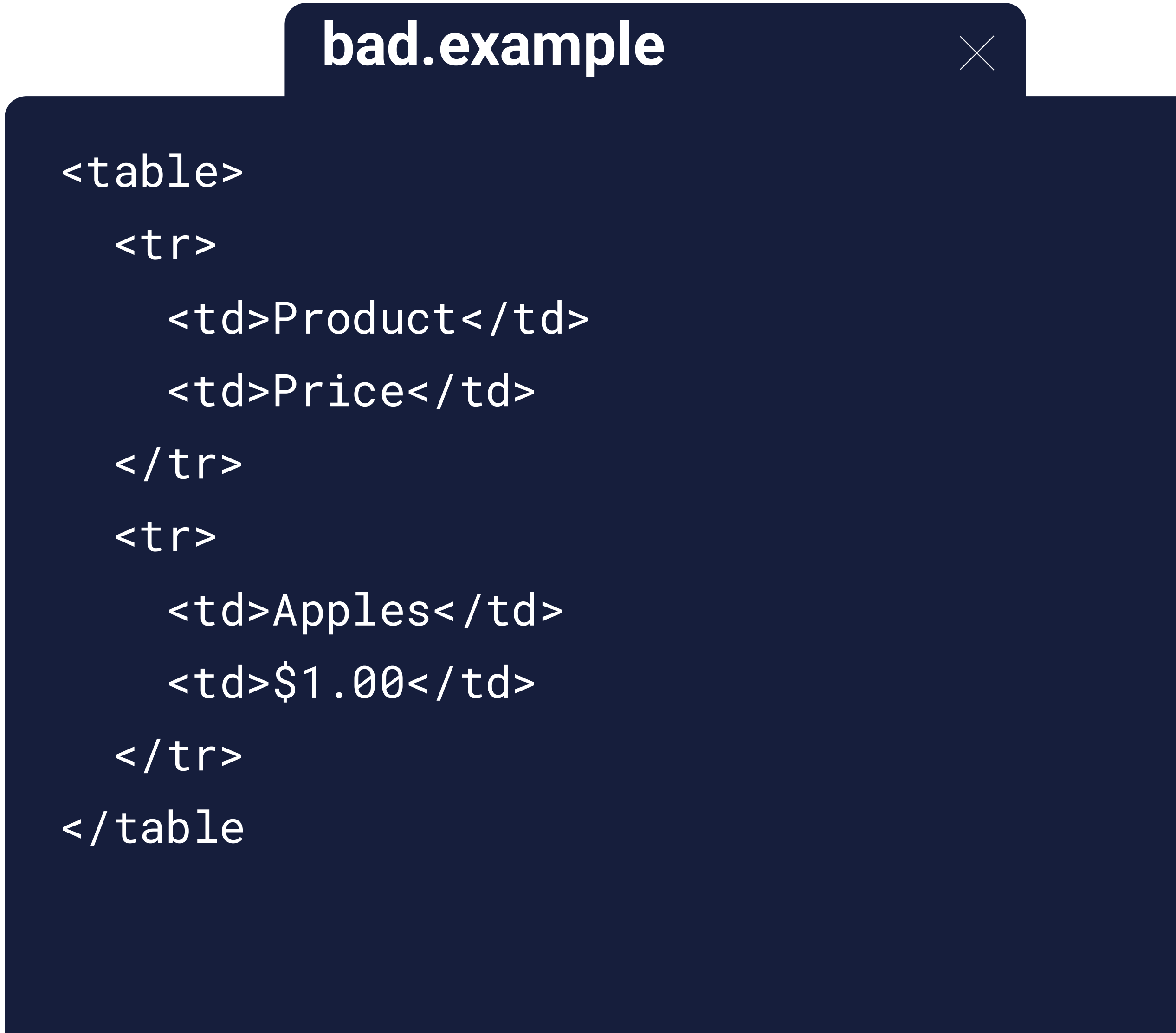
- Structure tables with proper headers and associations
- Use `<th>` tags and scope attributes
- Create tables that are readable by screen readers

Table accessibility



good.example

```
<table>
  <thead>
    <tr>
      <th scope=“col”>Product</th>
      <th scope=“col”>Price</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Apples</td>
      <td>$1.00</td
    </tr>
  </tbody>
</table
```



bad.example

```
<table>
  <tr>
    <td>Product</td>
    <td>Price</td>
  </tr>
  <tr>
    <td>Apples</td>
    <td>$1.00</td>
  </tr>
</table
```

Accessibility testing tools

- Tools analyze websites for accessibility issues
- Identify common problems and suggest improvements
- Essential for maintaining compliance
- Example: Google Lighthouse

{brightboost}

Google Lighthouse

- Accessibility testing tool
- Integrated into Chrome DevTools
- Accessiblity score from 0 to 100
- WCAG
- Run accessibility test before launching

{brightboost}

Without accessibility testing

Potential issues are left undetected

Non-compliance to accessibility standards can lead to lawsuits and fines

Loss of customers, reduced reach

{brightboost}

Final exercise

final-exercise.md

Quiz time!

Question 1

To meet WCAG guidelines, what should you avoid when designing a form?

{A} Using clear labels for all fields

{B} Using color alone to indicate required fields

{C} Grouping related fields with fieldsets

{D} Providing error messages next to each form field

Question 2

Which of the following features ensures a website is "Operable" under the POUR principles?

{A} High-contrast text

{B} Keyboard navigation

{C} Clear error messages

{D} Relying on the mouse for usage

Question 3

Which of the following is required by the EAA for compliance?

{A} Implementation of AI-based customer service

{B} Multi-language support for all digital content

{C} Accessible product / service

{D} Integration with social media platforms

Question 4

What does WCAG 2.1 introduce that was not covered in WCAG 2.0?

{A} Guidelines for mobile accessibility

{B} Standards for print accessibility

{C} Regulations for broadcasting

{D} None of the above

Question 5

Which feature would improve the accessibility of a contact form?

{A} Auto-submission after a time limit

{B} Providing a “reset form” button

{C} Allowing users to submit the form without completing every field

{D} Adding ARIA roles to enhance screen reader support

Question 6

Which of the following is NOT a recommended practice for writing an accessibility statement?

{A} Being honest about known accessibility issues

{B} Using technical jargon to describe accessibility features

{C} Providing contact information for accessibility support

{D} Updating the statement regularly

Question 7

Which of the following practices can improve the experience of screen reader users?

{A} Using ARIA roles to provide additional context

{B} Hiding content from screen readers that is only decorative

{C} Ensuring that navigation menus are announced correctly

{D} All of the above

Question 8

Which of the following scenarios would require detailed alt text?

{A} A decorative pattern used as background

{B} A chart showing quarterly sales figures

{C} A purely aesthetic image with no informational value

{D} A button with text displayed on it

Question 9

Which of the following is a good practice to make a website's navigation accessible?

{A} Hiding navigation menus on mobile devices

{B} Removing focus indicators from links

{C} Providing keyboard access to all interactive elements

{D} Using hover effects that do not work on touchscreens

Question 10

For which type of image is an empty alt attribute (alt="") appropriate?

{A} A decorative border image

{B} A product image on an e-commerce site

{C} An infographic

{D} A team photo on a company page

Question 11

Which ARIA role is most appropriate for a modal dialogue?

{A} role=“alert”

{B} role=“dialog”

{C} role=“presentation”

{D} role=“button”

Question 12

Why is it important to use relative units (e.g., em, rem) instead of absolute units (e.g., px) in responsive design?

{A} To make the website load faster

{B} To ensure consistent font sizes across devices

{C} To improve the website’s lighthouse report

{D} To increase the complexity of the code

Question 13

Which of the following HTML code snippets correctly uses semantic elements instead of generic `<div>` tags?

```
<div class="main-content"><p>This is the main content</p></div>
```

{A} `<section><p>This is the main content</p></section>`

{B} `<article><p>This is the main content</p></article>`

{C} `<header><p>This is the main content</p></header>`

{D} `<main><p>This is the main content</p></main>`

Question 14

Lighthouse is used to test a website’s:

{A} Performance

{B} SEO

{C} Accessibility

{D} All of the above

Questions?

maaike.vanputten@brightboost.nl

{brightboost}

{brightboost}

www.brightboost.nl