



WEB ACCESSIBILITY BASICS

TechRacine Academy

<https://techracine.com>

<https://academy.techracine.com>

Frontend Developer Advanced Path

Table of Contents

1. Introduction to Web Accessibility	3
2. Why Accessibility Matters	3
3. Understanding WCAG Standards	3
4. Semantic HTML and Accessibility	3
5. Keyboard Navigation	4
6. Screen Readers	4
7. Accessible Forms	4
8. Alt Text for Images.....	4
9. Color Contrast.....	5
10. Accessible Typography	5
11. ARIA Attributes	5
12. Accessible Navigation	5
13. Responsive Accessibility	5
14. Focus Management	6
15. Error Messages and Validation.....	6
16. Accessibility Testing Tools	6
17. Accessibility in Modern Frameworks.....	6
18. Accessibility and SEO	6
19. Accessibility Best Practices	7
20. Common Accessibility Mistakes	7
21. Legal and Business Importance	7
22. Real World Usage	7
23. Summary.....	8
24. Exercises	8

1. Introduction to Web Accessibility

Web accessibility ensures websites and applications can be used by all users, including people with disabilities.

Accessible applications improve usability, inclusiveness, and compliance.

Modern frontend engineering strongly emphasizes accessibility.

Accessibility supports users with visual, hearing, motor, and cognitive impairments.

Professional software products should always prioritize accessibility.

2. Why Accessibility Matters

Accessibility improves usability for all users.

It increases audience reach and improves user experience.

Accessible applications are often legally required in many industries.

3. Understanding WCAG Standards

WCAG stands for Web Content Accessibility Guidelines.

WCAG defines accessibility standards for websites and applications.

The major principles are perceivable, operable, understandable, and robust.

4. Semantic HTML and Accessibility

Semantic HTML improves accessibility significantly.

Screen readers understand semantic elements better.

```
<header>
```

```
</header>
```

```
<nav>
```

```
</nav>
```

```
<main>
```

```
</main>
```

5. Keyboard Navigation

Applications should be fully usable with keyboards.

Users should navigate forms and menus without mouse dependency.

6. Screen Readers

Screen readers convert webpage content into speech.

Accessible HTML improves screen reader compatibility.

7. Accessible Forms

Forms should include labels and validation messages.

Accessible forms improve usability.

```
<label for="email">
```

```
    Email Address
```

```
</label>
```

8. Alt Text for Images

Alt text describes images for screen readers.

It improves accessibility and SEO.

```

```

9. Color Contrast

Proper color contrast improves readability.

Low contrast text becomes difficult to read.

Accessibility standards define minimum contrast ratios.

10. Accessible Typography

Readable typography improves accessibility.

Proper font sizes and spacing improve usability.

11. ARIA Attributes

ARIA stands for Accessible Rich Internet Applications.

ARIA improves accessibility for dynamic interfaces.

```
<button aria-label="Close">
```

```
</button>
```

12. Accessible Navigation

Navigation should remain simple and predictable.

Users should easily understand page structures.

13. Responsive Accessibility

Accessibility must work across devices and screen sizes.

Responsive layouts should remain keyboard and screen-reader friendly.

14. Focus Management

Visible focus indicators improve keyboard navigation.

Focus should move logically between elements.

```
button:focus {  
  
    outline: 2px solid blue;  
  
}
```

15. Error Messages and Validation

Validation messages should clearly explain problems.

Accessible forms provide understandable feedback.

16. Accessibility Testing Tools

Tools such as Lighthouse, axe DevTools, and WAVE help test accessibility.

Regular testing improves compliance and usability.

17. Accessibility in Modern Frameworks

React, Angular, Vue, and modern frameworks support accessibility features.

Frontend developers should follow accessibility practices regardless of framework.

18. Accessibility and SEO

Accessible websites often perform better in search engines.

Semantic HTML and proper headings improve SEO.

19. Accessibility Best Practices

Use semantic HTML.

Provide keyboard navigation.

Maintain proper contrast ratios.

Use accessible labels and headings.

Test applications with accessibility tools.

20. Common Accessibility Mistakes

Avoid missing alt text.

Do not rely only on colors for meaning.

Avoid inaccessible forms.

Do not remove focus outlines.

21. Legal and Business Importance

Many countries require accessible digital products.

Accessibility improves brand reputation and inclusiveness.

22. Real World Usage

Accessibility is important in SaaS applications, education platforms, government systems, banking applications, AI dashboards, and enterprise portals.

Modern frontend engineering strongly emphasizes inclusive design.

23. Summary

Web accessibility is an essential part of professional frontend development.

Understanding semantic HTML, keyboard navigation, ARIA, forms, and accessibility testing improves application quality.

Accessible applications improve usability for everyone.

24. Exercises

1. Add semantic HTML to webpages.
2. Create accessible forms.
3. Practice keyboard navigation.
4. Add alt text to images.
5. Improve color contrast.
6. Use ARIA attributes.
7. Test accessibility using Lighthouse.
8. Create accessible navigation menus.
9. Build responsive accessible layouts.
10. Analyze accessibility of popular websites.