

# Barrierefreiheit

---

## Lernziele

In dieser Sitzung lernst du:

- ☐ was Barrierefreiheit ist und warum sie für Webentwickler wichtig ist
  - ☐ welche Vorteile semantisches HTML bietet
  - ☐ wie man Code barrierefrei gestaltet
- 

## Barrierefreiheit

Im Web Development ist es wichtig, so vielen Menschen wie möglich den Zugang zu unseren Websites zu ermöglichen, auch Menschen mit Behinderungen (hörend, sehend, neurologisch, körperlich, etc.). Es gibt starke wirtschaftliche Gründe für die Anpassung barrierefreier Designs:

- Wir können ein breiteres Publikum erreichen (ältere Menschen, Menschen mit Behinderungen, Menschen mit langsamer Internetverbindung).
  - Barrierefreies Design ist wichtig für SEO.
  - Barrierefreies Design macht Websites benutzerfreundlicher und verbessert die allgemeine Benutzererfahrung.
  - In vielen Ländern ist es gesetzlich vorgeschrieben, Websites barrierefrei zu gestalten.
- 

## Semantisches HTML

Semantisches HTML hilft Screenreadern und anderen unterstützenden Technologien dabei, nützliche Informationen über die Struktur einer Website zu erhalten. Es ist auch wichtig für SEO und verbessert die Lesbarkeit unseres HTML.

- Verwende Elemente wie **header**, **nav**, **main**, **article**, **aside**, **footer**, um jede Seite zu strukturieren.
  - Verwende pro Seite nur eine **h1**-Überschrift.
  - Überspringe keine Überschriftenebenen (verwende **h1** einmal, dann für die nächste Überschriftenebene **h2**, für die strukturell nächste Ebene **h3**).
- 

## ARIA

ARIA steht für Accessible Rich Internet Applications. Es ist eine Sammlung von Attributen und Rollen, die dabei helfen, Websites für Menschen mit Behinderungen zugänglicher zu machen.

### ARIA-Rollen

Eine ARIA-Rolle ist ein Attribut, das einem HTML-Element hinzugefügt werden kann. Sie geben den Elementen eine semantische Bedeutung. Es gibt sechs verschiedene Kategorien von Rollen. Schauen wir uns kurz zwei Kategorien an:

- Dokumentstruktur-Rollen wie **note** oder **tooltip** beschreiben die Struktur eines Inhaltsabschnitts.
- Widget-Rollen wie **slider** oder **menu** beschreiben den Typ des Elements, das auf einer Website präsentiert wird.

💡 Eine umfassende Liste der Rollen und ihrer Kategorien findest du in den [MDN ARIA Roles docs](#).

## ARIA-Zustände und Eigenschaften

ARIA-Zustände und -Eigenschaften beziehen sich auf ähnliche Merkmale. Sie bieten spezifische Informationen zu Elementen, deren Zustand und Beziehung zu anderen Elementen. Unterstützende Technologien wie Screenreader nutzen sie, um Inhalte den Benutzern zu präsentieren.

💡 Eine Liste der ARIA-Zustände und -Eigenschaften findest du in den [MDN ARIA Attributes docs](#).

Zwei wichtige ARIA-Attribute sind:

- **aria-label**: Definiert ein Label für ein interaktives Element. Verwende es, wenn der zugängliche Name eines Elements fehlt, zum Beispiel wenn ein Button keinen Text, sondern nur ein Icon enthält:

```
<button aria-label="Schließen" onclick="...">
  <svg ...><path .../></svg>
</button>
```

- **aria-labelledby**: Identifiziert, welches Element das Element, auf das es angewendet wird, beschriftet. Einige Elemente haben eine native Möglichkeit, ein anderes Element mit seinem Label zu referenzieren (zum Beispiel Input-Elemente und Label-Elemente, auf die wir später eingehen werden). Wenn es keine native Möglichkeit gibt, ein beschriftendes Element zu referenzieren, verwende das **aria-labelledby**-Attribut:

```
<nav aria-labelledby="title">
  <h2 id="title">Produkte</h2>
  ...
</nav>
```

## Barrierefreiheit Quick Wins

- Verwende **alt**-Attribute für deine Bilder:

```

```

- Verwende Farben mit hohem Kontrast, damit deine Inhalte auf allen Geräten gut sichtbar sind.
- Stelle sicher, dass alle deine interaktiven Elemente einen zugänglichen Namen haben.
- Verwende einfache und klare Sprache.

---

## Resources

- [MDN: Accessibility](#)
- [The A11y Project: Checklist](#)
- [Web Accessibility Initiative: Introduction to Web Accessibility](#)
- [Web Accessibility for Designers](#)