

Corso WCAG 2.1

Rendiamo accessibili i nostri siti
WEB



Scopo del corso

Dopo i corsi su HTML e CSS, **aggiungiamo un tassello fondamentale:**
l'accessibilità web, spesso trascurata ma **essenziale per un design inclusivo**.

Cosa impareremo:

- ✓ Cosa sono le WCAG 2.1 e perché sono importanti
- ✓ Come implementare modifiche concrete in HTML/CSS
- ✓ Strumenti gratuiti per testare l'accessibilità
- ✓ Esempi pratici "prima/dopo" per ogni criterio

Perché questo corso?

- ⚠ Oltre il 15% della popolazione globale ha disabilità (fonte: WHO)
- ⚠ Requisito legale in molti paesi (es. Direttiva UE, ADA negli USA)
- ⚠ Migliora UX e SEO per tutti gli utenti

WCAG – strumenti

Lo strumento che useremo durante il corso è

<https://codepen.io>

CodePen is a social development environment. At its heart, it allows you to write code in the browser, and see the results of it as you build. A useful and liberating online code editor for developers of any skill, and particularly empowering for people learning to code. We focus primarily on front-end languages like HTML, CSS, JavaScript, and preprocessing syntaxes that turn into those things.

Iscrivetevi e seguite il profilo creato apposta per il corso

<https://codepen.io/matteobaccan>

WCAG – editor

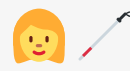
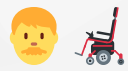


- Codepen.io
- Notepad
- Notepad++
- VisualStudio Code
- Codespace di GitHub

Va bene qualsiasi editor, non visuale, meglio se con syntax highlighter e code completion

<https://github.com/matteobacchan/CorsoWCAG>

Cosa sono?

Linee guida per rendere i contenuti web accessibili a:

 Non vedenti |  Disabili motori |  Neurodiversi |  Anziani

Chi le sviluppa?

Il **W3C** (World Wide Web Consortium) attraverso un processo:

- ◆ Collaborativo
- ◆ Basato su evidenze
- ◆ Aggiornato periodicamente (WCAG 2.2 in lavorazione)

I 4 principi fondanti (Pour)

1. Percepibile

Informazioni e componenti UI devono essere presentati in modi **percepibili** da tutti i sensi.

Esempi:

- Testo alternativo per immagini
- Sottotitoli per video
- Contrasto colore adeguato

Perché è importante?

Permette a screen reader e dispositivi assistivi di interpretare i contenuti.

I 4 principi fondanti (pOur)

2. Operabile

L'interfaccia deve essere **utilizzabile** con diverse modalità di input.

Esempi:

- Navigazione da tastiera
- Tempo sufficiente per leggere i contenuti
- Evitare contenuti lampeggianti

Perché è importante?

Garantisce accesso a utenti con disabilità motorie o epilessia.

I 4 principi fondanti (poUr)

3. Uncomprensibile 🧠

Contenuti e funzionalità devono essere **chiari e prevedibili**.

Esempi:

- Messaggi d'errore intuitivi
- Linguaggio semplice
- Flusso di navigazione coerente

Perché è importante?

Aiuta utenti con disabilità cognitive o non madrelingua.

I 4 principi fondanti (pouR)

4. Robusto

I contenuti devono essere compatibili con **tecnologie attuali e future**.

Esempi:

- HTML semantico validato
- Compatibilità con screen reader
- Utilizzo corretto di ARIA

Perché è importante?

Assicura longevità del sito e adattamento a nuovi dispositivi.

Key Takeaway: I principi POUR sono interdipendenti.
Un sito accessibile li soddisfa *tutti contemporaneamente*.

Evoluzione

Evoluzione WCAG

1999 ➡ WCAG 1.0 | 2008 ➡ WCAG 2.0 | 2018 ➡ WCAG 2.1 | 2023 ➡ WCAG 2.2

Livelli di Conformità

- **A:** Requisiti base (obbligatorio per legge in UE)
- **AA:** Standard raccomandato (soddisfa la maggior parte delle disabilità)
- **AAA:** Ottimizzazione avanzata (contesti specifici)

1. Testo Alternativo (1.1.1)

Prima

```

```

Problema: Mancanza di alt text per non vedenti.

Dopo

```

```

Strumenti:

- WAVE <https://wave.webaim.org/>
- Lighthouse <https://developer.chrome.com/docs/lighthouse>

2. Contrasto Colori (1.4.3)

Prima

```
.button { background: #888; color: #fff; } /* Contrasto 4:1 */
```

Problema: Contrasto insufficiente (< 4.5:1 per testo normale).

Dopo

```
.button { background: #0056b3; color: #fff; } /* Contrasto 7:1 */
```

Strumenti:

- Contrast Checker <https://webaim.org/resources/contrastchecker/>
- Color Contrast: <https://dequeuniversity.com/rules/axe/4.10/color-contrast>

3. HTML Semantico (4.1.2)

Prima

```
<div>Scopri di più</div>
```

Problema: Elemento non semantico per bottoni/navigazione.

Dopo

```
<button aria-label="Apri menu">Scopri di più</button>  
<nav aria-label="Menu principale">...</nav>
```

Strumenti:

- HTML Validator (validator.w3.org)
- Axe DevTools (estensione browser)
- IBM Accessibility Equal Access Toolkit: Accessibility Checker

4. Form Accessibili (3.3.2)

Prima

```
<input type="text" name="email">
```

Problema: Manca label associata.

Dopo

```
<label for="email">Email:</label>  
<input type="email" id="email" aria-describedby="email-help">  
<span id="email-help" class="sr-only">Inserisci un indirizzo valido</span>
```

CSS per screen reader:

```
.sr-only { position: absolute; left: -10000px; }
```

5. Navigazione da Tastiera (2.1.1)

Prima

```
<div class="dropdown" onclick="openMenu()">Menu</div>
```

Problema: Non navigabile via tastiera (manca `tabindex`).

Dopo

```
<div class="dropdown" tabindex="0" role="button" aria-expanded="false">Menu</div>
```

Verifica:

- Usa `Tab` e `Enter` per testare
- Strumento: Tota11y (khan.github.io/tota11y/)

6. Multimedia (1.2.2)

Prima

```
<video controls>  
  <source src="video.mp4">  
</video>
```

Problema: Mancano sottotitoli.

Dopo

```
<video controls>  
  <source src="video.mp4">  
  <track src="sottotitoli.vtt" kind="captions" label="Italiano">  
</video>
```

Strumenti:

- Amara (per creare sottotitoli) - YouTube Studio (auto-captioning)

7. Responsive Design (1.4.10)

Prima

```
<meta name="viewport" content="width=device-width">
```

CSS problematico:

```
.container { width: 1200px; }
```

Dopo

```
.container { max-width: 100%; }  
@media (max-width: 768px) { ... }
```

Strumenti:

- Browser DevTools (responsive mode) - Lighthouse (mobile accessibility)

8. ARIA per Dinamicità (4.1.3)

Prima

```
<div id="alert">Nuovo messaggio!</div>
```

Problema: Screen reader non notificano cambiamenti.

Dopo

```
<div id="alert" role="alert" aria-live="assertive">Nuovo messaggio!</div>
```

Verifica:

- NVDA Screen Reader ([nvaccess.org](https://www.nvaccess.org))
- VoiceOver (macOS)
- Simulatore Daltonismo <<https://www.toptal.com/designers/colorfilter>

9. Focus Visibility (2.4.7)

Prima

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

Problema: Focus non visibile per utenti keyboard-only.

Dopo

```
.button:focus {  
  outline: 3px solid #0056b3;  
  outline-offset: 2px;  
}
```

Strumenti:

- Tasti `Tab` + `Shift+Tab` per test manuale
- Focus Order Viewer in axe DevTools

10. Lingua della Pagina (3.1.1)

Prima

```
<html>  
<head><title>Welcome</title></head>
```

Problema: Lingua non dichiarata per screen reader.

Dopo

```
<html lang="it">  
<head><title>Benvenuto</title></head>
```

Verifica:

- Validatore HTML (<https://validator.w3.org/>)
- Strumento: axe "html-has-lang" rule

Testing Completo

Checklist:

1. Validazione HTML/CSS
2. Test contrasto colori
3. Navigazione solo da tastiera
4. Screen reader test (es. JAWS, NVDA)
5. Lighthouse Audit

Tool Gratuiti:

- axe DevTools
- WAVE Evaluation Tool
- Pa11y (pa11y.org)

- **WCAG 2.1 Full Guide:** <https://www.w3.org/TR/WCAG21/>
- **WebAIM Checklist:** <https://webaim.org/standards/wcag/checklist>
- **MDN Accessibility Guide:** <https://developer.mozilla.org/en-US/docs/Web/Accessibility>
- **Definizioni e argomenti:** <https://it.wikipedia.org>

Ogni immagine inserita riporta la fonte

Disclaimer

L'autore ha generato parte di questo testo con strumenti di AI. Dopo aver generato la bozza del documento, l'autore ha rivisto, modificato e integrato il testo a proprio piacimento e si assume la responsabilità ultima del contenuto di questa pubblicazione.

L'immagine di sfondo è stata generata con <https://app.haikei.app>

IN PREPARAZIONE

11. Tabelle Accessibili (1.3.1)

Prima

```
<div>Nome</div><div>Età</div>  
<div>Mario</div><div>30</div>
```

Problema: Struttura non semantica per dati tabellari.

Dopo

```
<table>  
  <caption>Dipendenti</caption>  
  <thead><tr><th scope="col">Nome</th><th scope="col">Età</th></tr></thead>  
  <tbody><tr><td>Mario</td><td>30</td></tr></tbody>  
</table>
```

Strumenti:

- Screen reader test con NVDA

- Table Inspector in WAVE

13. Saltare Contenuti Ripetitivi (2.4.1)

Prima

```
<header>
  <nav>...</nav>
</header>
```

Problema: Nessun modo per saltare la navigazione.

Dopo

```
<a href="#main" class="skip-link">Salta al contenuto</a>
<nav>...</nav>
<main id="main">...</main>

<style>
.skip-link {
  position: absolute;
  left: -9999px;
}
.skip-link:focus {
```

14. Animazioni e Motion (2.3.3)

Prima

```
.slider { transition: transform 0.8s; }
```

Problema: Animazioni potenzialmente dannose per utenti sensibili.

Dopo

```
@media (prefers-reduced-motion: reduce) {  
  .slider { transition: none; }  
}
```

Strumenti:

- Simulatore preferenze movimento: Chrome DevTools > Rendering
- CSS Media Queries Level 5 validator

15. Link Descrittivi (2.4.4)

Prima

```
<a href="/blog">Clicca qui</a>
```

Problema: Testo non contestuale fuori contesto.

Dopo

```
<a href="/blog">Leggi gli ultimi articoli del blog</a>
```

Verifica:

- Lista link con Web Developer Toolbar
- Strumento: Link Text Analyzer (<https://axesslab.com/link-texts/>)

16. Ruoli ARIA per Widget (4.1.2)

Prima

```
<div class="modal">...</div>
```

Problema: Ruolo non definito per componenti complessi.

Dopo

```
<div class="modal" role="dialog" aria-labelledby="modalTitle">  
  <h2 id="modalTitle">Conferma ordine</h2>  
  ...  
</div>
```

Strumenti:

- ARC Toolkit (<https://www.paciellogroup.com/toolkit/>)
- Accessible Name Calculator