

Laboratorio di Accessibilità

Paolo Castagna castagna@cefriel.it
Basilio Pugliese pugliese@cefriel.it

“The power of the Web is in its universality.
Access by everyone regardless of disability
is an essential aspect.”
– *Tim Berners-Lee*

Progetto Scuola - U.S.R. Lombardia



Le giornate di laboratorio



- Ciascuna giornata di laboratorio sarà suddivisa in:
 - ▶ 1 ora di spiegazione, spunti, discussione assieme
 - ▶ 6 ore di lavoro a gruppi
 - ▶ 1 oretta, anche meno, di esposizione da parte di ciascun gruppo sul lavoro svolto, le decisioni prese, le difficoltà incontrate
- Per ogni giornata di laboratorio:
 - ▶ Obiettivi precisi
 - ▶ Riferimenti e lavoro da svolgere a casa



Obiettivi di oggi



- Progettazione del sito
- Realizzazione della struttura su filesystem rappresentante sezioni e sottosezioni del sito
- Realizzazione dei template delle pagine accessibili



Sommario



- **Progettazione di un sito accessibile**
- WCAG 1.0: le 14 linee guida
- Cascading Style Sheet (CSS)
- Esempi da cui prendere spunto





Progettare un sito



- Contesto: avere chiaro fin dall'inizio l'obiettivo del sito
 - ▶ In che ambito si colloca il sito?
 - ▶ Quali sono gli obiettivi del sito?
 - ▶ Perché realizzare un sito del genere?
- Contenuti: massima cura per i contenuti
 - ▶ Quali contenuti?
 - ▶ Come scrivere per Internet?
- Utenti: sono loro a decretare il successo di un sito
 - ▶ Chi sono gli utenti del sito?
 - ▶ Cosa cercano?
 - ▶ Quali sono i loro bisogni, i loro modelli mentali?
- Nel documento di progetto descrivere: contesto, contenuti e utenti cercando di rispondere alle domande sopra riportate. Non più di 2 pagine scritte, non più di 5 minuti di resoconto orale.



Accessibilità sin dall'inizio



- E' opportuno occuparsi degli aspetti di accessibilità sin dalle primissime fasi di progettazione del sito
- Realizzare un sito accessibile non è più costoso di realizzare un sito non accessibile
- Realizzare siti accessibili porta verso siti meglio strutturati, markup più puliti, layout e presentazioni più gestibili
- Puntare sempre al massimo: Triple-A





Controllo di qualità



- Validazione del markup (HTML/XHTML):
 - ▶ <http://validator.w3.org/>
- Validazione dei fogli di style CSS:
 - ▶ <http://jigsaw.w3.org/css-validator/>
- Controllo dell'accessibilità:
 - ▶ Bobby - <http://bobby.watchfire.com/>
 - ▶ Torquemada - <http://www.webxtutti.it/testa.htm>
 - ▶ Cynthia - <http://www.contentquality.com/>
 - ▶ Dreamweaver MX 2004
- Validare le proprie pagine: si consiglia di procedere passo a passo validando e correggendo gli errori di volta in volta



Sommario



- Progettazione di un sito accessibile
- **WCAG 1.0: le 14 linee guida**
- Cascading Style Sheet (CSS)
- Esempi da cui prendere spunto





WCAG: le 14 linee guida



W3C®

1. Provide equivalent alternatives to auditory and visual content
2. Don't rely on color alone
3. Use markup and style sheets and do so properly
4. Clarify natural language usage
5. Create tables that transform gracefully
6. Ensure that pages featuring new technologies transform gracefully
7. Ensure user control of time-sensitive content changes
8. Ensure direct accessibility of embedded user interfaces
9. Design for device-independence
10. Use interim solutions
11. Use W3C technologies and guidelines
12. Provide context and orientation information
13. Provide clear navigation mechanisms
14. Ensure that documents are clear and simple



1. Provide equivalent alternatives to auditory and visual content



- Fornisci alternative equivalenti al contenuto acustico e visuale (immagini, grafici, applet, frame, suoni, video, ecc.)

```

```





2. Don't rely on color alone



- Non fare affidamento unicamente sul colore
 - ▶ Cerca di scegliere colori ad elevato contrasto e non associare una semantica a elementi distinti unicamente attraverso il colore
 - ▶ Il testo nero sullo sfondo bianco è l'abbinamento che assicura il miglior contrasto e la miglior leggibilità
 - ▶ La pagina dovrebbe essere comprensibile anche in bianco e nero, senza colori



3. Use markup and style sheets and do so properly



- Utilizza markup e fogli di stile e fallo in maniera appropriata
 - ▶ Ritornare alle origini e utilizzare correttamente i tag strutturali con la corretta semantica: `<h1>`, ... , `<h6>`, `<p>`, `<blockquote>` `<cite>`, `<q>`, `<abbr>`, `<acronym>`, ecc.
 - ▶ Utilizzare i fogli di stile (CSS) per la presentazione
 - ▶ Perché `` e `` invece di `<i>` e ``?





4. Clarify natural language usage



- Chiarisci il linguaggio utilizzato

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="it">

<abbr title="Versus">vs.</abbr>
<acronym title="HyperText Markup Language">HTML</acronym>

<acronym
  title="World Health Organization"
  lang="en">WHO</acronym>

<acronym
  title="Organizzazione mondiale della sanità"
  lang="it">OMS</acronym>

<p>And with a certain <span lang="fr">je ne sais quoi</span>,
she entered both the room, and his life, forever. <q>My name
is Natasha,</q> she said. <q lang="it">Piacere,</q> he
replied in impeccable Italian, locking the door.</p>
```



5. Create tables that transform gracefully



- Crea tabelle che si trasformino elegantemente

```
<table border="1" summary="Questa tabella indica..." />
<caption>Orari di partenza...</caption>
<tr>
  <th id="header1">Città</th>
  <th id="header2">Orario</th>
</tr>
<tr>
  <td headers="header1">Milano</td>
  <td headers="header2">7:30</td>
</tr>
<tr>
  <td headers="header1">Roma</td>
  <td headers="header2">9:30</td>
</tr>
<tr>
  <td headers="header1">Napoli</td>
  <td headers="header2">11:00</td>
</tr>
</table>
```

- E' meglio non utilizzare tabelle per il layout della pagina





6. Ensure that pages featuring new technologies transform gracefully



- Assicurati che le pagine che le pagine che utilizzano nuove tecnologie si trasformino elegantemente
 - ▶ I documenti dovrebbero poter essere letti anche se è completamente assente il supporto per i fogli di style
 - ▶ Script, applet... sono veramente necessari?



7. Ensure user control of time-sensitive content changes



- Assicura all'utente il controllo degli elementi che cambiano nel tempo
 - ▶ Meglio non creare contenuti che si muovono
 - ▶ Non usare <blink> and <marquee>
 - ▶ Consentire agli utenti di mettere in pausa o riavvolgere filmati e contributi audio





8. Ensure direct accessibility of embedded user interfaces



- Assicura l'accessibilità diretta di interfacce utente inserite nell'HTML
 - ▶ Attenzione all'accessibilità di ActiveX, Applet, Flash, ecc.
 - ▶ Sono veramente necessari?



9. Design for device-independence



- Progetta indipendentemente dal dispositivo
 - ▶ Dispositivi: PC, WAP, Palm, PDA, Smart Phone, Web-TV, ecc. Non è possibile fare versioni differenti per ciascun dispositivo
 - ▶ Dispositivi di input: tastiere, mouse, voce, barre Braille, ecc.)
 - ▶ Se non ho il mouse come utilizzo un menù a tendina basato sull'evento JavaScript `onmouseover`?





10. Use interim solutions



- Usa soluzioni temporanee per sopperire alle lacune tecnologiche dei device o delle tecnologie assistive attuali
 - ▶ Spesso occorre trovare vie alternative per sopperire alla mancanza di strumenti che implementino correttamente gli standard attuali: fai questo nel modo più semplice possibile



11. Use W3C technologies and guidelines



- Usa le tecnologie e le linee guida del W3C
 - ▶ Rispetto degli standard
 - ▶ Evitare funzionalità dichiarate deprecate
 - ▶ Content negotiation





12. Provide context and orientation information



- Fornisci informazioni di contesto e orientamento
 - ▶ Label significativi per ciascun elemento di controllo nelle interfacce
 - ▶ Suddivisione dei contenuti della pagina in blocchi omogenei
- Come valutare il rispetto di questa linea guida in automatico?



13. Provide clear navigation mechanisms



- Fornisci chiari meccanismi di navigazione
 - ▶ Strumenti di navigazione efficaci
 - Navigazione globale, locale, contestuale
 - Ricerca, mappe, indici, tour, help, ecc.
 - Navigare per tipologia d'utente, per compito, ecc.
 - ▶ Fai capire sempre all'utente dove si trova e dove può andare
 - [Home](#) > [Computer](#) > [Software](#)
 - ▶ Usa un *labelling* significativo, omogeneo e coerente con i titoli delle pagine e le voci del menù di navigazione
- Come valutare il rispetto di questa linea guida in automatico?





14. Ensure that documents are clear and simple



- Garantisci che i documenti siano chiari e semplici
 - ▶ Parole semplici
 - ▶ Un concetto per ogni paragrafo
 - ▶ Paragrafi modulari
 - ▶ Piramide invertita
 - ▶ Dai del tu al lettore
 - ▶ Microcontent
- Come valutare il rispetto di questa linea guida in automatico?



Sommario



- Progettazione di un sito accessibile
- WCAG 1.0: le 14 linee guida
- **Cascading Style Sheet (CSS)**
- Esempi da cui prendere spunto





Tag e semantica



- E' importante utilizzare esclusivamente i tag semantici del linguaggio HTML e affidare la presentazione ai fogli di stile (CSS)
- La struttura delle pagine HTML si semplifica
- E' opportuno ridurre al minimo l'uso di marcatori strutturali <div> e con gli attributi "class" e "id"
- Separazione tra contenuto, presentazione e funzionalità:
 - ▶ HTML/XHTML: contenuto
 - ▶ CSS: presentazione
 - ▶ DOM, JavaScript: funzionalità



Esempio



- Il tag <blockquote> è stato usato spesso per indentare invece che per riportare una citazione

```
<blockquote cite="http://www.rcs.it/rcslibri/bompiani/_minisiti/coelho/undiciminuti/" lang="it">Fino a quando il principe azzurro non fosse apparso, lei non avrebbe potuto far altro che sognare. S'innamorò per la prima volta a undici anni, mentre si recava a piedi da casa fino alla scuola.</blockquote>
```

```
<q cite="http://www.usabile.it/archivio/feb2003a.htm#stanca">Il problema è evitare un'accessibilità vuota, che arricchisca solo le tasche dei valutatori o di organismi di verifica senza soddisfare nella realtà i bisogni dell'utenza.</q>
```





Cascading Style Sheet (CSS)



- Sintassi:
 - ▶ `selettore { proprietà: valore; }`
- ```
body {
 font-family: Georgia;
 font-size: 0.9em;
}
```
- ```
h1, h2, h3, h4, h5, h6 {  
  color: #900;  
}
```



CSS e classi



```
p.destra {  
  text-align: right;  
}
```

```
p.sinistra {  
  text-align: left;  
}
```

```
<p class="destra">Testo del paragrafo.</p>  
<p class="sinistra">Testo del paragrafo.</p>
```





CSS e id



```
h1#news1 {  
    ...  
}  
  
h1#news2 {  
    ...  
}  
  
<h1 id="news1">News 1</h1>  
<h2 id="news2">News 2</h2>
```



Class e id



- Class serve per identificare un gruppo di tag
- Id serve per identificare univocamente un singolo tag
- E' meglio utilizzare class per gli elementi che potrebbero ripetersi e id per gli elementi di cui si è certi che ce n'è uno ed uno solo nella pagina
- Solitamente intestazione, piè di pagina, colonne laterali sono racchiuse da un tad <div> con un id:
 - ▶ <div id="header">...</div>
- Si possono avere anche selettori che utilizzino solo class o id:
 - ▶ #header
 - ▶ .secondario





Altri tipi di selettori



- Seleziona ogni elemento contenuto in un elemento

```
h1 a {  
  color: #000;  
}
```

- Seleziona tutti i diretti discendenti di un elemento

```
code > a {  
  color: #000;  
}
```

- Seleziona i discendenti diretti di un elemento

```
div + a {  
  color: #000;  
}
```



Pseudoclassi



```
a, a:link {  
  color: #009;  
  background-color: transparent;  
  text-decoration: underline;  
}  
a:visited {  
  color: #006;  
  text-decoration: underline;  
}  
a:hover {  
  text-decoration: none;  
}  
a:active {  
  color: #000;  
  background-color: #eee;  
  text-decoration: none;  
}  
a:focus {  
  background-color: #eee;  
  text-decoration: none;  
}
```





CSS e HTML



- Sili in linea
 - ▶ `<p style="text-align: center;">...</p>`
- Stili inclusi nella pagina
 - ▶ `<style type="text/css">
 p {text-align: left;}
 </style>`
- Stili linkati
 - ▶ `<head>
 <link rel="stylesheet"
 type="text/css"
 href="file.css"
 media="screen" />
 </head>`
- Stili importati
 - ▶ `<style type="text/css">
 @import "/css/nomefile.css";
 </style>`



CSS: riferimenti



- Cascading Style Sheets, level 1
W3C Recommendation 17 Dec 1996, revised 11 Jan 1999
<http://www.w3.org/TR/REC-CSS1>
- Cascading Style Sheets, level 2
W3C Recommendation 12-May-1998
<http://www.w3.org/TR/REC-CSS2/>
- W3Schools CSS Tutorial
<http://www.w3schools.com/css/>
- A List Apart: CSS
<http://www.alistapart.com/topics/css/>





Sommario



- Progettazione di un sito accessibile
- WCAG 1.0: le 14 linee guida
- Cascading Style Sheet (CSS)
- **Esempi da cui prendere spunto**



Esempi da cui prendere spunto



- <http://www.stopdesign.com/>
- <http://www.usabile.it/>
- <http://www.webaccessibile.org/>
- <http://www.lucarosati.it/>
- <http://www.porteapertesulweb.it>





Buon lavoro!

