

# Appunti di Tecnologie Web

A.A. 2023/2024

Rosso Carlo

# Indice

<b>1</b>	<b>css</b>	<b>3</b>
1.1	pseudoclassi . . . . .	4
1.2	pseudo-elementi . . . . .	5
1.3	unità di misura . . . . .	5
1.4	colori . . . . .	5
1.5	url . . . . .	5
1.6	stile del testo . . . . .	6
1.7	stile dello sfondo . . . . .	6
<b>2</b>	<b>Domande e risposte</b>	<b>7</b>
2.1	HTML . . . . .	7
	Le tabelle devono sempre essere evitate? . . . . .	7
	Le tabelle sono un ostacolo alla facile comprensione per le perosne con disabilità, quindi andrebbero evitate dove possibile. . . . .	7
	Elencare gli accorgimenti che si devono adottare per rendere una tabella accessibile. . . . .	7
	Esempio di tabella accessibile . . . . .	8
2.2	CSS . . . . .	9
	Differenze fondamentali tra id e class . . . . .	9
	Ordine di applicazione degli stili in CSS . . . . .	10
	Di che colore è il testo all'interno del tag? . . . . .	10
2.3	Web Design . . . . .	10
	Le convenzioni interne non devono essere rotte, per non disorientare l'utente . . . . .	10
	La struttura organizzativa deve essere poco ampio e molto profonda . . . . .	11
	Una buona regola è segnalare i link che portano a pagine in lingua diversa da quella del sito utilizzando le bandiere nazionali. . . . .	11
	Il numero massimo di voci in un menù è 6. . . . .	11
	Una struttura organizzativa gerarchica deve essere ampia e poco profonda. . . . .	11
	Il layout a schede può andare incontro a problemi di manutenibilità nel tempo. . . . .	11
<b>3</b>	<b>Riassunto</b>	<b>12</b>
3.1	HTML . . . . .	12

Tabelle . . . . .	12
-------------------	----

# 1 CSS

Procedimento a cascata:

1. impostazioni personali dell'utente
2. dichiarazioni definite "!important"
3. impostazioni di stile inline definite dall'autore della pagina
4. fogli di stile *embedded* definiti dall'autore
5. fogli di stile esterni definiti dall'autore
6. impostazioni di stile predefinite dal browser (le impostazioni di default cambiano da browser a browser)

Selettore di attributo:

```
1  element_name[attribute=valore] {
2  /// base
3  /* regole */
4  }
5
6  element_name[attribute] {
7  /// attributo presente
8  /* regole */
9  }
10
11 element_name[attribute="valore"] {
12 /// attributo uguale alla stringa "valore"
13 /* regole */
14 }
15
16 element_name[attribute~="valore"] {
17 /// attributo contenente la parola "valore"
18 /* regole */
19 }
20
```

**Def. 1.1 (ereditarietà)**    1. *flag attributo=valore !important*

2. *num id*

3. *num attributi*

4. *num tag html*

## 1.1 pseudoclassi

pseudoclasse	descrizione
:link	link non visitato
:visited	link visitato
:active	link attivo
:hover	mouse sopra
:focus	elemento attivo (tab)
:first	prima pag per media paginati
:left	pagine di sinistra
:right	pagine di destra
:first-child	prima occorrenza di un elemento
:lang	seleziona una lingua

Tabella 1: pseudoclassi

Esempio:

```
1 a:link:hover {  
2     font-size: 2em;  
3 }  
4  
5 // generale  
6 selettore::pseudoclasse {  
7     /* regole */  
8 }
```

## 1.2 pseudo-elementi

pseudo-elemento	descrizione
:first-letter	prima lettera di un blocco
:first-line	prima riga di un blocco
:before	testo da aggiungere prima
:after	o dopo un elemento

Tabella 2: pseudo-elementi

## 1.3 unità di misura

unità	descrizione	esempio
em	altezza media del font, relativo	h1 { margin:0.5em }
px	utilizzato, assoluto	p { font-size:12px }
in	numero pixel nello schermo, assoluto	p { font-size:0.5in }
cm	inch, pollici (1 in = 2,54 cm), assoluto	p { font-size:1cm }
mm	millimetri, assoluto	p { font-size:1mm }
pt	punti (1 pt = 1/72 in), assoluto	p { font-size:12pt }
pc	pica (1 pc = 12 pt), assoluto	p { font-size:1pc }
%	percentuale, relativo all'elemento principale	p { font-size:120% }

Tabella 3: unità di misura

## 1.4 colori

Espressi in formato rgb esadecimale funzionano su tutti i browser. Esempio:

```
1 p {  
2     /// color: #RRGGBB  
3     color: #ff0000;  
4 }
```

## 1.5 url

Esempio:

```

1 p {
2     /// url: url(protocollo://dominio/percorso)
3     background-image: url(percorso/immagine.jpg);
4     background-repeat: repeat;
5 }

```

## 1.6 stile del testo

- dimensione: `font-size: value;`
- interlinea: `line-height`
- sovrapposizione: `z-index`
- corsivo: `font-style`
- grassetto: `font-weight // bold, normal, bolder, lighter`
- variante maiuscoletto: `font-variant // small-caps`
- maiuscolo o minuscolo: `text-transform // uppercase, lowercase, capitalize`
- decorazione: `text-decoration // underline, overline, line-through`
- colori: `color, background-color`
- distanza tra le lettere: `letter-spacing`
- distanza tra le parole: `word-spacing`
- indentazione: `text-indent`
- allineamento orizzontale: `text-align // left, right, center, justify`
- allineamento verticale: `vertical-align`

## 1.7 stile dello sfondo

- `background-attachment`: stabilisce se l'immagine segue il contenuto nello scroll oppure no
- `background-repeat`

- `background-position`
- `background-image: url(/* path */)`

## 2 Domande e risposte

### 2.1 HTML

**Le tabelle devono sempre essere evitate?**

**Falso**, le tabelle non devono essere usate per presentare e organizzare i contenuti. Le tabelle sono pesanti per il browser da renderizzare. Questo a parte, se bisogna visualizzare dei dati in una struttura tabellare allora è bene usare le tabelle, rendendole accessibili, ovvero usando `thead`, `tbody`, `tfoot`, `th`, `td`, è consigliato usare anche `aria-describedby`, `scope`, `lang`, `abbr` per permettere agli *screen reader* una lettura più agevole della tabella per gli utenti che li usano.

**Le tabelle sono un ostacolo alla facile comprensione per le persone con disabilità, quindi andrebbero evitate dove possibile.**

**Vero**, le tabelle non devono essere usate per presentare e organizzare i contenuti. Inoltre sono complesse da renderizzare per i browser. Sarebbe meglio evitarle, a meno che i dati da visualizzare non si prestino alla forma tabellare. In ogni caso, se si intende adottarle è importante strutturarle rigorosamente in modo accessibile.

**Elencare gli accorgimenti che si devono adottare per rendere una tabella accessibile.**

Di seguito sono riportati i passi da seguire per rendere accessibile una tabella:

1. Riportare una breve descrizione della tabella. Il tag della descrizione deve avere un `id`, che viene associato alla tabella mediante l'attributo `aria-describedby`;
2. Utilizzare i tag `thead`, `tbody`, `tfoot` per dividere semanticamente le informazioni. Infatti, per esempio, html rende sempre disponibile il `tfoot`;
3. Utilizzare l'attributo `lang` per segnalare agli *screen reader* le parole che differiscono dalla lingua di default della pagina;



4. Utilizzare `th` per le celle che descrivono i dati in una riga o una colonna. Oltre a `th`, bisogna utilizzare anche `scope`, per indicare se quell'informazione vale per la riga o per la colonna. Si consiglia anche l'adozione di `abbr` per consigliare ad uno screen reader l'adozione di un'abbreviazione piuttosto che il titolo della riga o della colonna completo;
5. Utilizzare il tag `caption` per scrivere la didascalia della tabella.

Inoltre si consiglia di alternare i colori delle righe o delle colonne per le tabelle molto lunghe o molto larghe, prestando attenzione al contrasto dei colori.

### Esempio di tabella accessibile

```
1 <table aria-describedby="table_occupazione_palestra">
2 <caption>Occupazione della palestra</caption>
3
4 <thead>
5 <tr>
6 <th></th>
7 <th scope="col" abbr="9 - 10">9.00-10.00</th>
8 <th scope="col" abbr="10 - 11">10.00-11.00</th>
9 <th scope="col" abbr="11 - 12">11.00-12.00</th>
10 <th scope="col" abbr="12 - 15">12.00-15.00</th>
11 <th scope="col" abbr="15 - 16">15.00-16.00</th>
12 </tr>
13 <tbody>
14 <tr>
15 <th scope="row" abbr="lun">Lunedì</th>
16 <td colspan="3">Hip Hop</td>
17 <td></td>
18 <td>Pilates</td>
19 </tr>
20 <tr>
21 <th scope="row" abbr="mar">Martedì</th>
22 <td colspan="2">Karate</td>
23 <td>Hip Hop</td>
24 <td></td>
25 <td>Pilates</td>
26 </tr>
```

```

27 <tr>
28 <th scope="row" abbr="mer">Mercoledì</th>
29 <td></td>
30 <td></td>
31 <td>Hip Hop</td>
32 <td rowspan="3">Ginnastica Artistica</td>
33 <td></td>
34 </tr>
35 <tr>
36 <th scope="row" abbr="gio">Giovedì</th>
37 <td colspan="2">Karate</td>
38 <td></td>
39 <td></td>
40 </tr>
41 <tr>
42 <th scope="row" abbr="ven">Venerdì</th>
43 <td></td>
44 <td></td>
45 <td>Hip Hop</td>
46 <td>Pilates</td>
47 </tr>
48 <tbody>
49 <table>
50
51 <span id="table_occupazione_palestra">Si tratta di una tabella che
52 illustra
53 quali corsi occupano la palestra durante la settimana dal lunedì al
venerdì,
dalle 9 di mattina fino alle 16 di sera.</span>

```

## 2.2 CSS

### Differenze fondamentali tra id e class

- id univoco per tag, class raggruppa i tag;
- un tag può avere al massimo un solo id, mentre può avere tante classi;
- l'id si può riferire nell'url come "link\_della\_pagina#id\_elemento";

- in css il selettore dell'id è "#", mentre per le classi è ".";
- l'id può iniziare solo con "\_" oppure con una lettera;
- si può usare `.selectById(id)` per selezionare un elemento tramite id in js;
- ha valenza massima nel calcolo della specificità, dopo `!important`.

## Ordine di applicazione degli stili in CSS

1. stili di default del browser;
2. fogli di stile esterni (.css) definiti dall'autore del sito;
3. fogli di stile embedded all'interno del documento;
4. stile definito inline dall'autore, all'interno dei tag;
5. se presente `!important`;
6. stile dell'utente.

## Di che colore è il testo all'interno del tag?

Per rispondere al quesito si procede calcolando la tupla di specificità per ciascuna regola definita in css: (`!important` (se presente), id, attributi o classi, tag e pseudoclassi). Le tuple sono riordinate dalla maggiore alla minore e gli stili con specificità maggiore sovrascrivono gli stili con specificità minore. In caso di tuple con la medesima specificità, viene applicata l'ultima.

## 2.3 Web Design

### Le convenzioni interne non devono essere rotte, per non disorientare l'utente

**Vero**, le convenzioni interne non devono mai essere rotte, perché aiutano l'utente a famialiarizzare con il sito e a prendere confidenza con esso.

**La struttura organizzativa deve essere poco ampia e molto profonda**

**Falso.** L'ampiezza massima consigliata è di 7 voci, comunque si sconsiglia un'ampiezza superiore alle 10 voci. Mentre la profondità massima consigliata è di 5 voci e fino a 7 voci nel caso si arrivi ad una specificità elevata. Detto questo, la profondità corrisponde al numero di click che l'utente è tenuto a fare per passare dalla homepage alla pagina. Se la struttura organizzativa del sito è poco profonda, vuol dire che l'utente ha modo di raggiungere la pagina che cerca con meno click e quindi più velocemente. Inoltre, risulta più semplice mantenere e gestire nel tempo una gerarchia ampia e poco profonda. Bisogna ricordare però che un numero di voci troppo elevato rischia di causare il "sovraccarico cognitivo", perché l'utente ha bisogno di memorizzare troppe informazioni nell'arco di poco tempo e rischia di disorientare l'utente.

**Una buona regola è segnalare i link che portano a pagine in lingua diversa da quella del sito utilizzando le bandiere nazionali.**

**Falso,** perché la bandiera è collegata alla nazione e non alla lingua, infatti ci sono stati in cui si parlano più lingue e ci sono stati diversi in cui si parla la medesima lingua. Piuttosto, si consiglia di segnalare tali link mediante l'uso di ISO che segnalano la lingua.

**Il numero massimo di voci in un menù è 6.**

**Falso,** il numero di voci consigliato di un menù è 7, ma si possono aggiungere ulteriori 3 voci. In ogni caso, bisogna tenere a mente il rischio di causare il "sovraccarico cognitivo" all'utente, ovvero la condizione in cui sono fornite troppe informazioni all'utente, e questo lo porta a confondersi o a perdersi nel sito.

**Una struttura organizzativa gerarchica deve essere ampia e poco profonda.**

**Vero,** infatti tale struttura organizzativa diminuisce il numero di click che l'utente è tenuto ad eseguire per raggiungere la pagina che cerca. Inoltre risulta più semplice gestire un sito con questo tipo di struttura organizzativa.

**Il layout a schede può andare incontro a problemi di manutenibilità nel tempo.**

**Vero:**

- Con l'aumento delle schede diventa difficile per gli utenti trovare i contenuti che stanno cercando;
- Quando qualche contenuto viene aggiunto, rimosso o modificato, è necessario aggiornare il layout a schede per riflettere i cambiamenti. Ciò spesso comporta l'aggiornamento delle etichette e del contenuto delle singole schede;
- Se non sono progettate tenendo conto dell'accessibilità, causano problemi agli utenti con disabilità visive oppure potrebbero non funzionare bene in schermi piccoli. Per questo motivo può diventare molto complesso ristrutturare il layout del sito per renderlo più accessibile.

Si ricorda inoltre che le regole dell'accessibilità sono continuamente in aggiornamento, per questo motivo è necessario ristrutturare spesso un sito.

## 3 Riassunto

### 3.1 HTML

#### Tabelle

- si utilizzano solo se i dati hanno struttura tabellare;
- NON per il layout;
- applicando con rigore le regole per renderle accessibili:
  1. `aria-describedby`;
  2. `caption`;
  3. dividere la tabella in `thead`, `tbody`, `tfoot`;
  4. le celle `th` devono avere anche `scope`, `abbr`;
  5. le celle in lingua diversa devono essere segnalate; l'attributo `lang`;
  6. `colspan`, `rowspan`.