

HTML parte 2

Corso di Web Design

Fabio Pittarello, Università Ca' Foscari Venezia - DAIS pitt@unive.it

Nota: il materiale contenuto in questo documento è disponibile solo per uso interno nell'ambito del corso di Web Design.

CSS Zen Garden

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

So What is This About?

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has

Select a Design:

- Manhattan Edition by José Tomás Tocino García
- Peace Of Mind by Carlos Varela
- 45 RPM by Thomas Michaud
- Pretty in Pink by Jordi Romkema
- Vertigo by Antonio Cella
- Vin Rouge by Thorsten Bopp
- Pinups by Emiliano Pennisi
- Zen City Morning by Ray Henry

Archives:

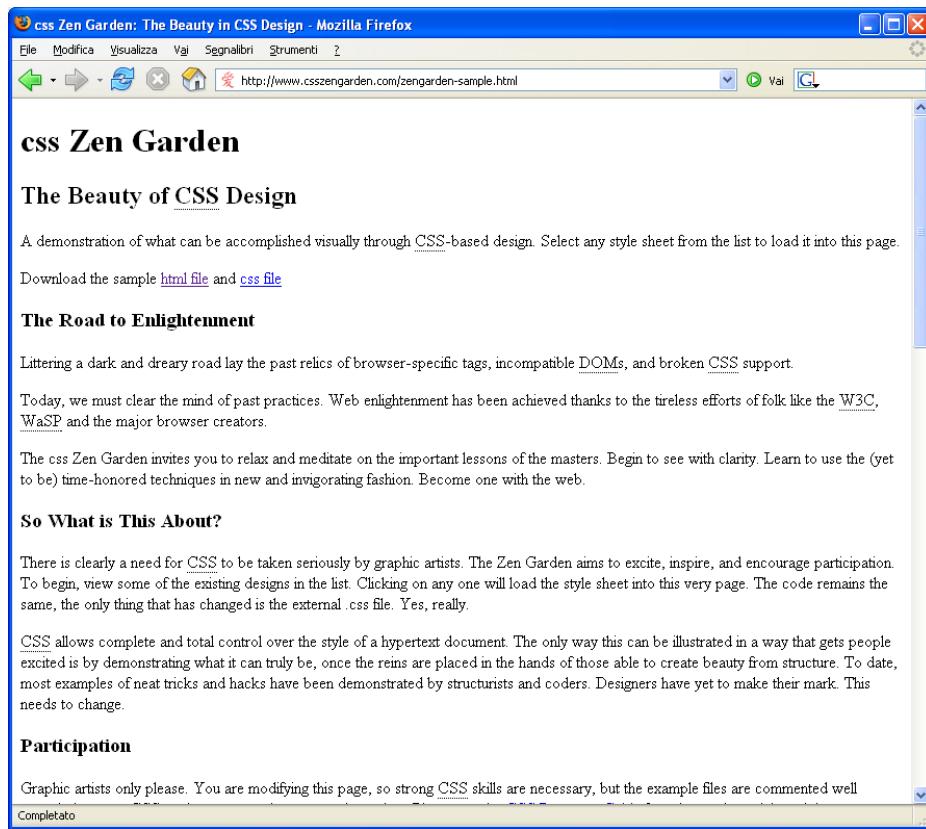
- next designs >
- View All Designs

Resources:

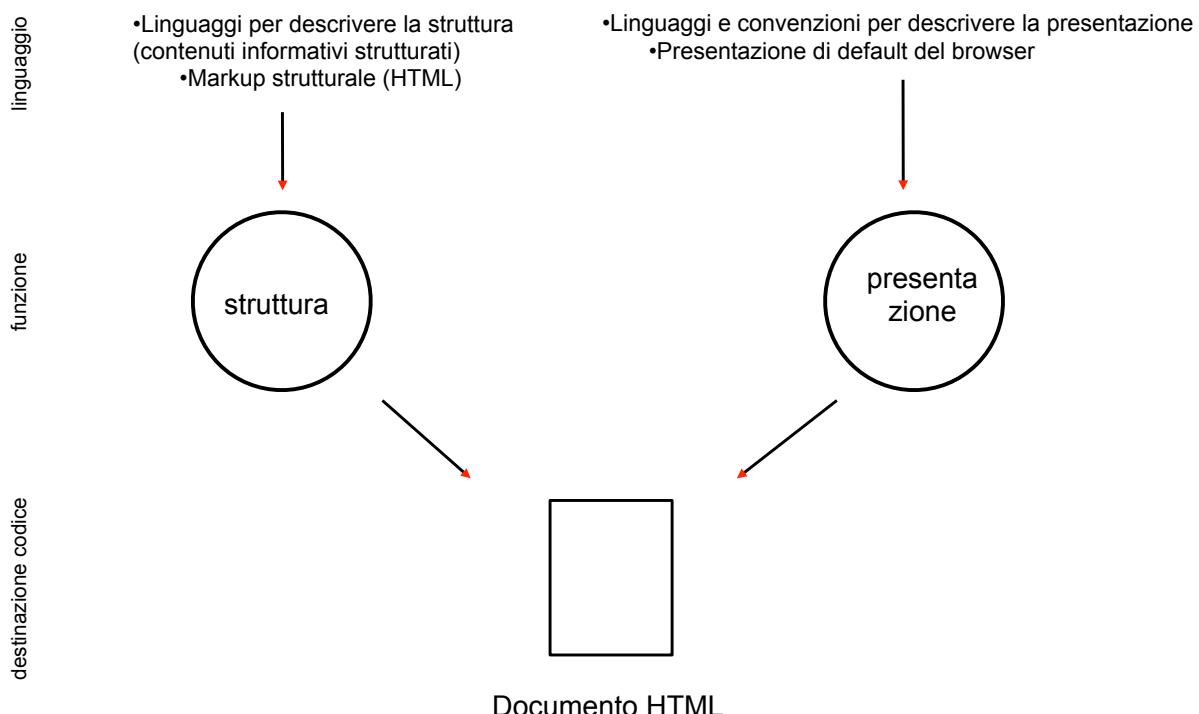
- View This Design's CSS
- CSS Resources
- FAQ

<http://www.csszengarden.com>

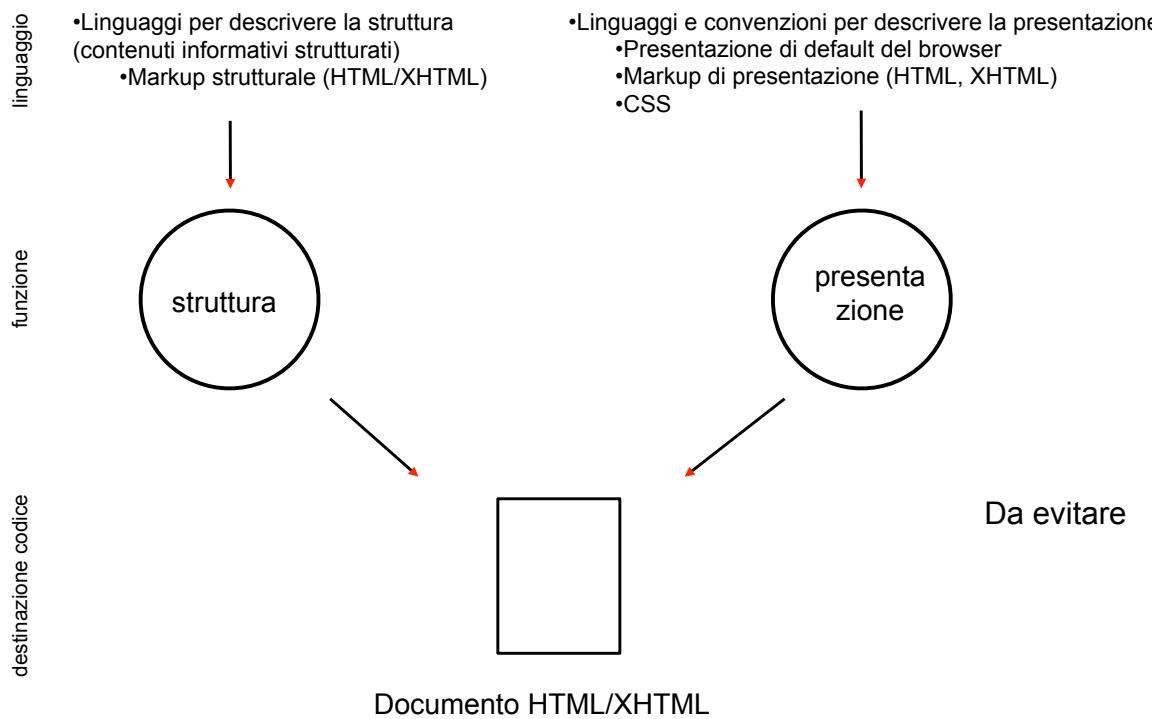
Lo stesso contenuto informativo ...



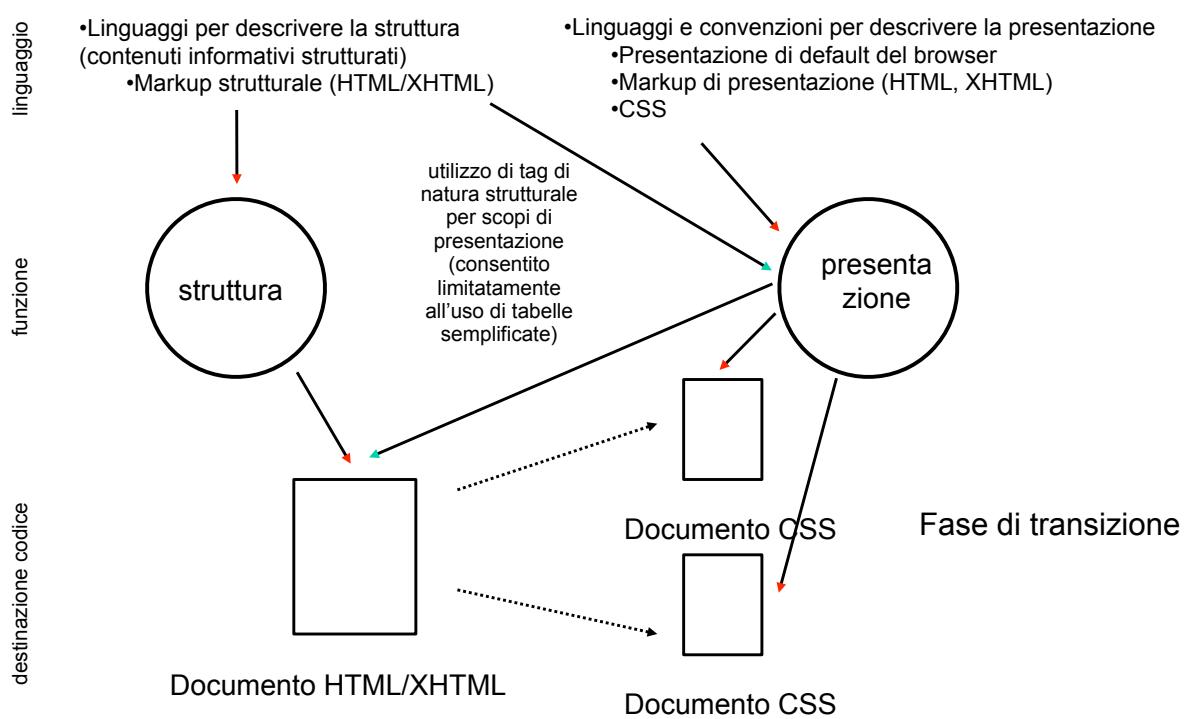
contenuti strutturati e presentazione: alle origini ...



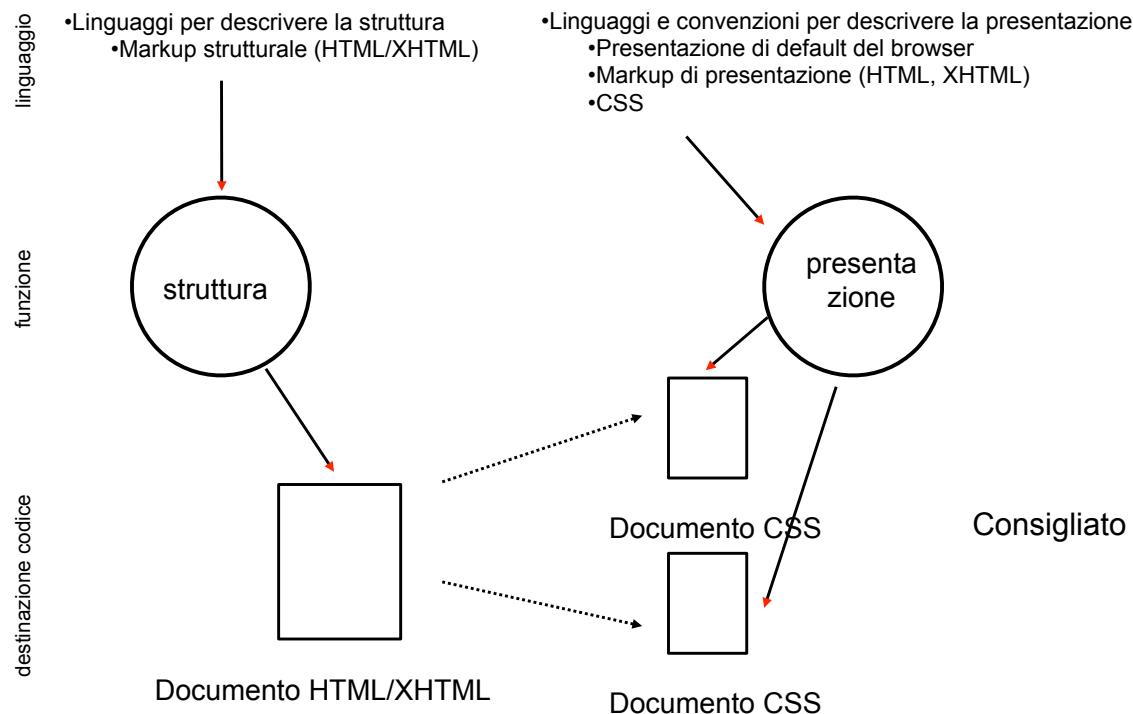
contenuti strutturati e presentazione: layout costruito solo con HTML / XHTML



contenuti strutturati e presentazione: layout ibridi progettati con HTML (tables) e CSS



contenuti strutturati e presentazione: layout CSS



Separare struttura e presentazione

- ◆ La **separazione di struttura e presentazione** è un requisito fondamentale della progettazione dei siti web.
- ◆ Tale metodologia di progettazione, che può essere adottata grazie alla disponibilità delle tecnologie XHTML e CSS permette di raggiungere i seguenti obiettivi:
 - ◆ **definizione unica dei contenuti**
 - ◆ possibilità di definire **molteplici stili di presentazione** a seconda dei dispositivi utilizzati e delle preferenze degli utenti
 - ◆ **maggior facilità nella manutenzione** dei contenuti e degli stili di presentazione
- ◆ Per ottenere questo risultato è **necessario utilizzare il markup HTML/XHTML**, che per ragioni storiche consentirebbe di esprimere (per XHTML fino a XHTML 1.0 compreso) sia proprietà strutturali che di presentazione, **per esprimere le sole proprietà strutturali** (markup strutturale), specificando successivamente la presentazione attraverso un set di regole CSS.



Il markup strutturale fino ad XHTML 1.0

Markup strutturale per il testo

- ◆ Possiamo suddividere i tag definiti per marcare le caratteristiche strutturali del testo in 2 gruppi
 - ◆ Elementi per definire blocchi strutturali (**block-level elements**)
 - ◆ Elementi interni (**inline styles**, limitatamente agli stili logici)

Block-Level Elements <p></p>

- ◆ Il significato del tag
 - ◆ Il tag <p> viene introdotto per denotare i paragrafi nei quali è suddiviso il testo di un documento HTML, es. <p>contenuto paragrafo</p>
 - ◆ Suddivisioni interne del paragrafo possono essere ottenute utilizzando il tag
 (break)
- ◆ Come vengono presentati i contenuti del tag
 - ◆ Nei browser visuali più diffusi il contenuto viene visualizzato partendo con una nuova linea di testo; inoltre sopra e sotto il paragrafo viene aggiunta una riga vuota
 - ◆ Nel caso di
 si ottiene un ritorno a capo senza aggiunta di righe vuote

Block-Level Elements <hn></hn>

- ◆ Il significato del tag
 - ◆ Il tag livello di intestazione più importante <h1> al livello meno importante <hn> viene utilizzato per rappresentare i livelli di intestazione di un testo, ordinate gerarchicamente in 6 livelli dal <h6>. La sintassi XHTML richiede che le **intestazioni di livello inferiore vengano precedute da quelle di livello superiore** (ad. es. non si può definire un contenuto di una pagina iniziando con <h3> e poi utilizzando solo successivamente <h1>). Non è consigliabile inoltre **saltare livelli per ottenere effetti di presentazione** (ad es. utilizzando <h1> e poi utilizzando <h4> per una sottosezione, trascurando <h2> ed <h3>)

Block-Level Elements <hn></hn>

- ◆ Come vengono presentati i contenuti del tag
 - ◆ Nei browser visuali più diffusi le intestazioni vengono visualizzate in grassetto; inoltre nella visualizzazione vengono aggiunti automaticamente un ritorno a capo e una riga vuota sopra e sotto l'intestazione.
 - ◆ <h1> viene rappresentato con il **carattere tipografico più grande** rispetto alla dimensione utilizzata di default per il testo contenuto nei paragrafi; i successivi con caratteri tipografici decrescenti
 - ◆ Generalmente <h5> e <h6> vengono rappresentati con **caratteri tipografici più piccoli**.
 - ◆ Spesso i designer utilizzano le intestazioni per ottenere un effetto di presentazione (ad esempio utilizzano h1 per ottenere un testo di grande dimensioni, anche se di importanza minima). E' una pratica sconsigliata.

Block-Level Elements Liste

- ◆ Il significato dei tag di lista
 - ◆ Le liste e gli elementi in esse contenuti costituiscono anch'essi elementi a livello di blocco. Perciò nella visualizzazione su browser standard tipicamente viene aggiunta una riga vuota prima e dopo la lista.
- ◆ XHTML prevede 3 tipologie di liste:
 - ◆ Liste non ordinate
 - ◆ Liste ordinate
 - ◆ Definizioni

Liste non ordinate

- ◆ Il significato del tag
 - ◆ Utilizzato per collezioni di elementi dove l'ordine non è importante
- ◆ Come vengono presentati i contenuti del tag
 - ◆ Davanti ad ogni elemento viene inserito un identificatore (un pallino, di default). La forma di default potrebbe essere cambiata utilizzando l'attributo **type="disc|circle|square"**, ma è una pratica sconsigliata (deprecated).
 - ◆ L'utilizzo di fogli di stile consente di ottenere una vasta gamma di identificatori, comprendendo anche l'uso di immagini definite dall'utente al posto delle forme predefinite
 - ◆ L'uso di forme tabellari che simulino la lista è altamente sconsigliato.

```
<ul>
  <li>elemento 1</li>
  <li>elemento 2</li>
  <li>elemento 3</li>
  <li>elemento 4</li>
</ul>
```

Liste ordinate

- ◆ Il significato del tag
 - ◆ Utilizzate per collezioni di elementi dove l'ordine è significativo
- ◆ Come vengono presentati i contenuti del tag
 - ◆ Davanti ad ogni elemento viene inserito un numero arabo (un pallino, di default). La forma di default potrebbe essere cambiata utilizzando l'attributo **type="1|A|a|I|i"**, ma è una pratica sconsigliata (deprecated).
 - ◆ L'utilizzo di fogli di stile consente di ottenere gli stessi identificatori di posizione
 - ◆ Anche in questo caso l'uso di forme tabellari che simulino la lista è altamente sconsigliato.

```
<ol>
  <li>elemento 1</li>
  <li>elemento 2</li>
  <li>elemento 3</li>
  <li>elemento 4</li>
</ol>
```

Definizioni

- ◆ Il significato del tag
 - ◆ Utilizzato per denotare termini e definizioni
- ◆ Come vengono presentati i contenuti del tag
 - ◆ I termini vengono solitamente visualizzati sul margine sinistro della pagina, e le definizioni vengono spostate a destra rispetto al margine
 - ◆ anche in questo caso va evitato l'uso ai fini di presentazione di queste strutture (ad es. utilizzando una struttura di lista per rappresentare un paragrafo indentato)

```
<dl>
  <dt>termine 1</dt>
  <dd>definizione 1 ...</dd>
  <dt>termine 2</dt>
  <dd>definizione 2 ...</dd>
</dl>
```

Liste nidificate

- ◆ E' possibile nidificare le liste ordinate e non ordinate:
 - ◆ Liste non ordinate
 - ◆ Viene utilizzata una tipologia diversa di identificatore a seconda del livello
 - ◆ Liste ordinate
 - ◆ Viene utilizzata una numerazione coerente per ogni livello di nidificazione
- ◆ E' possibile nidificare **liste miste**, costituite da liste ordinate e da liste non ordinate

Elementi interni (Inline Styles)

- ◆ Molti **tag HTML** vengono classificati come **Inline Styles** perchè si riferiscono ad aspetti di struttura o di presentazione relativi a stringhe di caratteri inseriti nel flusso di un testo (es. all'interno di un paragrafo).
Non vanno confusi con gli **Inline Styles CSS**, che si riferiscono invece alla specifica localizzata di regole CSS all'interno di un tag specifico.
- ◆ Questi tag condizionano il testo contenuto all'interno di essi, senza causare l'aggiunta di ritorni a capo o di righe vuote prima o dopo il testo marcato.
- ◆ La definizione di stili inline è **trasversale** rispetto alla nozione di markup strutturale e di presentazione.
Rispetto alla definizione di queste due tipologie di markup è utile suddividere gli Inline Styles in due categorie: **stili logici** (strutturali) e **stili fisici** (di presentazione)

Inline styles: stili logici

- ◆ Descrivono il significato, il contesto o l'uso dell'elemento racchiuso; più consoni alla formulazione originaria di HTML. L'aspetto di presentazione è a discrezione del browser; tuttavia i browser generalmente adottano le stesse convenzioni per la loro rappresentazione.

Tag	Descrizione	Solitamente visualizzato come
< abbr ></ abbr >	Abbreviazione	Testo normale
< acronym ></ acronym >	Acronimo	Testo normale
< cite ></ cite >	Riferimento a un altro documento	Corsivo
< code ></ code >	Codice	Font a larghezza fissa
< del ></ del >	Testo cancellato	Testo normale / Testo barrato orizzontalmente
< dfn ></ dfn >	Definizione di un'istanza	Testo normale
< em ></ em >	Enfatizzato	Corsivo
< ins ></ ins >	Testo inserito	Testo normale
< kbd ></ kbd >	Testo da tastiera	Font a larghezza fissa
< q ></ q >	Citazione breve	Corsivo
< samp ></ samp >	Testo di esempio	Font a larghezza fissa
< span ></ span >	Parte	Testo normale
< strong ></ strong >	Evidenziato	Grassetto
< var ></ var >	Variabile	Font a larghezza fissa / Corsivo

Block level-elements e inline styles per definire partizioni logiche

- ◆ Oltre ai tag già visti, che hanno una connotazione semantica legata al ruolo del testo contenuto all'interno, vengono definiti due tag strutturali addizionali che vengono utilizzati per definire **partizioni logiche**: `<div>` e ``
- ◆ Il significato dei tag
 - ◆ Il tag `<div>` è un marcitore generico utilizzato per dividere un documento in **sezioni**. E' un elemento a livello di blocco, per cui il testo marcato da `<div>` inizia su una riga nuova e una riga vuota viene inserita prima e dopo la fine del tag.
 - ◆ Il tag `` è un inline style generico da applicare ad una **stringa** (di testo) all'interno di un paragrafo o altri contenuti.

Block level-elements e inline styles per definire partizioni logiche

- ◆ Come vengono presentati i contenuti dei tag `<div>` e ``
 - ◆ Entrambi i tag vengono utilizzati in associazione con gli **attributi class e id**; le regole di stile associate a questi identificatori definiscono come devono essere presentati i contenuti di questi tag:
 - ◆ Attributo **class**, utilizzato per identificare l'appartenenza di diversi documenti a un gruppo
 - ◆ `risoluzione`
 - ◆ Attributo **id**, utilizzato per istanze uniche (un determinato valore di un id può essere **assegnato una sola volta in un documento X/HTML**)
 - ◆ `oggetto`
 - ◆ Nota: class e id possono essere utilizzati in quasi tutti i tag HTML, non sono specifici di div e span
 - ◆ Nota 2: è possibile utilizzare entrambi gli attributi in riferimento allo stesso tag HTML, ad es. `<div id="pinco" class="pallina">oggetto</div>`

Markup e attributi di presentazione per il testo

- ◆ Un certo numero di tag e attributi di markup sono stati definiti per identificare caratteristiche di presentazione. Attualmente il loro uso è fortemente sconsigliato (deprecated) dal consorzio W3C a favore della tecnologia CSS.
- ◆ Consideriamo 2 categorie:
 - ◆ Tag che identificano elementi interni
(Inline Styles, limitatamente agli stili fisici)
 - ◆ Attributi di presentazione da utilizzare insieme a tag strutturali per identificare caratteristiche di colore, sfondo, bordi, margini, caratteristiche dimensionali, ecc.)

Inline styles: stili fisici

Forniscono istruzioni specifiche sulle modalità di rappresentazione. Molti di questi stili sono ufficialmente sconsigliati da W3C a favore degli stili logici.

Tag	Descrizione	Funzione
	Grassetto (*)	Testo in grassetto
<big></big>	Grande	Testo leggermente più grande del testo adiacente
<blink></blink>	Lampeggiante	Testo lampeggiante (Netscape)
	Carattere	Specifica il tipo di carattere, la dimensione e il colore (sconsigliato)
<i></i>	Corsivo (*)	Testo in corsivo
<s></s>	Testo barrato orizzontalmente	Testo barrato orizzontalmente (sconsigliato)
<small> </small>	Piccolo (*)	Testo leggermente più piccolo del testo adiacente
<strike></strike>	Testo barrato orizzontalmente	Testo barrato orizzontalmente (sconsigliato)
	Pedice	Testo di dimensione più piccola, sopra la linea di base del testo adiacente
	Apice	Testo di dimensione più piccola, sotto la linea di base del testo adiacente
<tt> </tt>	Telescrivente	Carattere a larghezza fissa
<u> </u>	Sottolineato	Testo sottolineato (sconsigliato)

(*) ridefiniti in HTML 5

Il tag e la specifica del carattere tipografico

- ◆ La specifica del carattere tipografico in una pagina XHTML attraverso il tag rappresenta uno dei casi più evidenti di *inquinamento* del codice HTML con informazioni di presentazione, che ne ostacolano la lettura, la manutenzione e la portabilità dei contenuti a device diversi.
- ◆ Nelle slides seguenti viene tuttavia data come memoria storica una breve descrizione del tag e dei suoi attributi, nella consapevolezza che l'uso di questo elemento di markup è ufficialmente sconsigliato dal Consorzio W3C.

Il tag e la specifica del carattere tipografico

- ◆ Il tag appartiene agli inline style fisici, e viene utilizzato per specificare il tipo di carattere da utilizzare per renderizzare il testo contenuto all'interno del tag e alcune sue caratteristiche accessori, come la dimensione e il colore
- ◆ Sintassi del tag:
- ◆ `<font face="tipo di carattere" size="dimensione"
color="colore">contenuto`

Il tag e la specifica del carattere tipografico

- ◆ L'attributo **face** venne introdotto da Explorer 1.0, successivamente adottato da Netscape a partire dalla versione 3.0 e poi adottato nello standard
- ◆ L'attributo **face** NON garantisce che l'utente vedrà il font specificato.
Va considerato come un **consiglio** dato al browser, che visualizzerà il contenuto con diversi livelli di approssimazione a seconda dei font effettivamente installati nel sistema operativo dell'utente
- ◆ I valori dell'attributo si riferiscono a nomi di font specifici (es. Verdana) oppure a famiglie di font (es. Sans-serif); possono essere dati più valori, separandoli con una virgola.
- ◆ Es. testo
- ◆ Il browser confronta i valori dell'attributo con i font installati nel sistema. **Se nessuno dei font o delle famiglie suggerite è installato, viene utilizzato il font di default.**

Il tag e la specifica del carattere tipografico

- ◆ Attributo **size**
 - ◆ la dimensione del carattere viene misurata su una **scala da 1 a 7**, che non ha riferimento con le misure tipografiche convenzionali.
3 è il valore di default, che verrà mostrato nella dimensione specificata nelle Preferenze del browser dall'utente.
Gli altri valori danno luogo a visualizzazioni di dimensione maggiore (valori 4, 5, 6 e 7) o minore (valori 1 e 2), con incrementi o decrementi di dimensione del 20% tra valori adiacenti
 - ◆ Es. Carattere
 - ◆ La dimensione può essere specificata anche utilizzando **valori relativi (da -7 a +7)** al font di default o a quello definito nel tag <**basefont**>
 - ◆ Es. Carattere
 - ◆ In ogni caso un font non viene mai visualizzato più grande di 7 o più piccolo di 1, nemmeno se i valori relativi portano a valori superiori o inferiori)

Il tag e la specifica del carattere tipografico

◆ NOTA IMPORTANTE

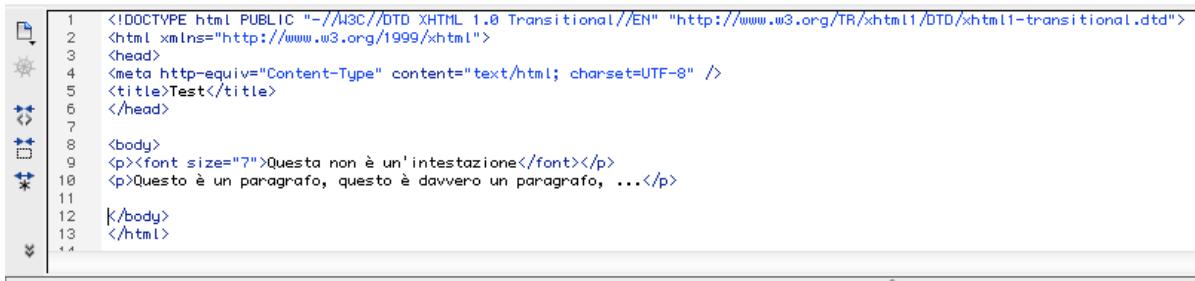
La specifica del carattere tipografico, in qualunque modo essa venga fatta (anche attraverso le regole CSS), NON offre la garanzia di visualizzazione del testo secondo le caratteristiche definite dagli autori del layout; la visualizzazione può essere influenzata da fattori diversi tra i quali il più importante è l'effettiva disponibilità del carattere specificato sul sistema hardware e software utilizzato dal browser che visualizza la pagina (anche se i nuovi web fonts promettono di risolvere questo problema nel futuro).

Contenuti strutturati e presentazione, note finali

- ◆ E' importante rimarcare come la definizione della struttura dei contenuti costituisca un livello più avanzato rispetto alla descrizione non strutturata del contenuto.
- ◆ Nella definizione della **struttura dei contenuti** vengono definite le **relazioni tra le differenti componenti testuali del documento, stabilendo precedenze, scale di gerarchia, raggruppamenti di elementi** (es. intestazioni, paragrafi, liste, ecc.).
- ◆ La fase di **presentazione non deve prescindere o contraddirre la strutturazione dei contenuti**. Al contrario i risultati di questa fase vanno utilizzati per presentare in maniera coerente i contenuti stessi.

Contenuti strutturati e presentazione, alcuni errori tipici

- ◆ **simulazione di caratteristiche strutturali attraverso l'uso di markup di presentazione** che produce un effetto esclusivamente visuale ma è privo di qualunque significato da un punto di vista strutturale (es. utilizzo di caratteri tipografici più grandi per simulare intestazioni, ecc.)



```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
5 <title>Test</title>
6 </head>
7
8 <body>
9 <p><font size="7">Questa non è un'intestazione</font></p>
10 <p>Questo è un paragrafo, questo è davvero un paragrafo, ...</p>
11
12 </body>
13 </html>
```

Questa non è un'intestazione

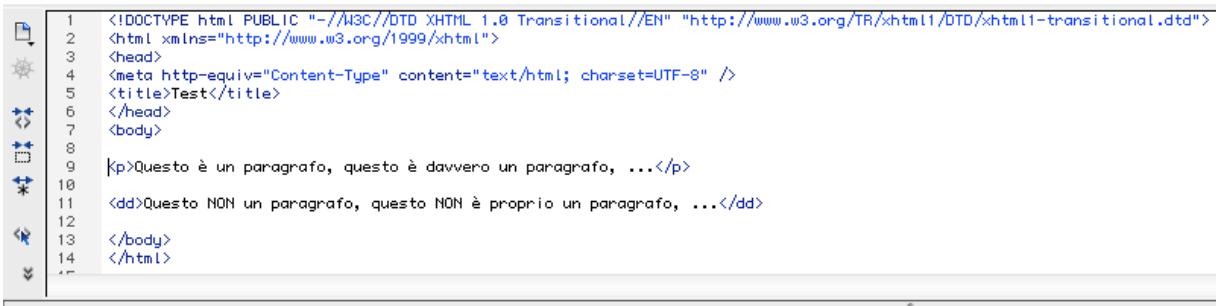
Questo è un paragrafo, questo è davvero un paragrafo, ...

Il tag e la specifica del carattere tipografico

- ◆ L'uso del tag font per fare apparire il testo più importante, simulando visualmente l'esito dell'uso delle intestazioni, è da sconsigliare perché **impedisce ai motori di ricerca di identificare gli elementi della pagina più importanti** da un punto di vista semantico.
L'indicizzazione del sito in questo caso avviene attribuendo pesi sbagliati ai diversi elementi, generando possibili problemi nella ricerca da parte degli utenti dei motori di ricerca stessi.

Contenuti strutturati e presentazione, alcuni errori tipici

- ◆ definizione di caratteristiche di presentazione attraverso l'uso scorretto di markup strutturale (es. uso di particolari marcatori come <dd> per ottenere un effetto visivo di tabulazione oppure utilizzo di immagini trasparenti per ottenere spaziature orizzontali e verticali).



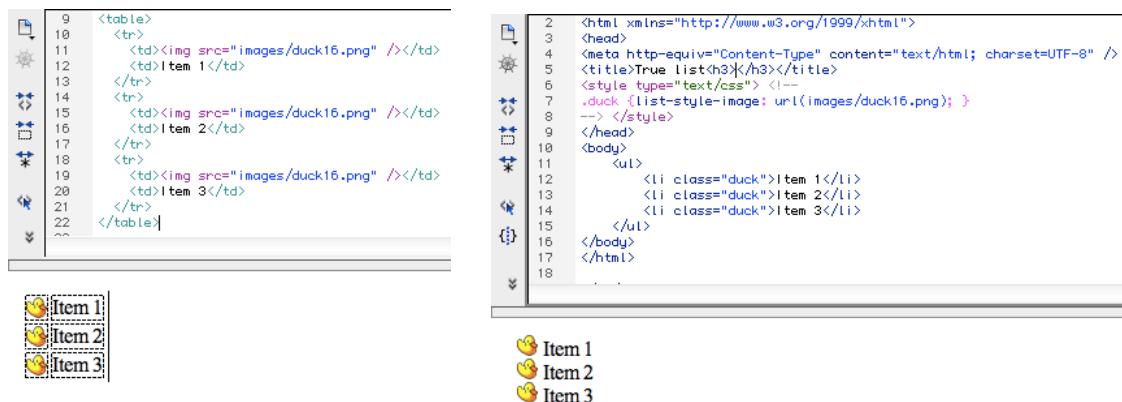
```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2  <html xmlns="http://www.w3.org/1999/xhtml">
3  <head>
4  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
5  <title>Test</title>
6  </head>
7  <body>
8
9  <p>Questo è un paragrafo, questo è davvero un paragrafo, ...</p>
10 <dd>Questo NON un paragrafo, questo NON è proprio un paragrafo, ...</dd>
11
12 </body>
13 </html>
```

Questo è un paragrafo, questo è davvero un paragrafo, ...

Questo NON un paragrafo, questo NON è proprio un paragrafo, ...

Contenuti strutturati e presentazione, alcuni errori tipici

- ◆ definizione di caratteristiche di presentazione attraverso l'uso scorretto di markup strutturale; un altro esempio di uso scorretto è la definizione di una lista con segnaposto costituito da un'immagine ottenuto attraverso l'uso combinato di una tabella nella quale una colonna viene utilizzata come contenitore per le immagini; lo stesso risultato visuale si può ottenere in modo più elegante attraverso l'uso di una lista e una regola di stile



```
9  <table>
10 <tr>
11 <td></td>
12 <td>item 1</td>
13 </tr>
14 <tr>
15 <td></td>
16 <td>item 2</td>
17 </tr>
18 <tr>
19 <td></td>
20 <td>item 3</td>
21 </tr>
22 </table>
```

```
2  <html xmlns="http://www.w3.org/1999/xhtml">
3  <head>
4  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
5  <title>True list</title>
6  <style type="text/css"> /*-
7  .duck {list-style-image: url(images/duck16.png); }
8  --> </style>
9  </head>
10 <body>
11     <ul>
12         <li class="duck">item 1</li>
13         <li class="duck">item 2</li>
14         <li class="duck">item 3</li>
15     </ul>
16 </body>
17 </html>
```



Item 1
Item 2
Item 3



Item 1
Item 2
Item 3

Contenuti strutturati e presentazione, alcuni errori tipici

- ◆ NOTA: anche l'**utilizzo di strutture tabellari per definire layout di pagina** costituisce una deviazione rispetto allo scopo per cui erano state introdotte le tabelle in HTML, cioè quello di rappresentare dati. L'utilizzo per tali scopi è stato tuttavia tollerato negli anni passati a causa della mancata disponibilità di implementazioni , mantenendo l'obiettivo finale di arrivare ad una separazione di contenuto strutturato e presentazione.

The screenshot shows two code snippets side-by-side in a code editor, illustrating the misuse of tables for layout.

Left Snippet: This shows a table used for data representation. The code defines a table with two columns: "Semestre" and "Energia prodotta". It contains two rows: one for "Primo" with value "2150" and one for "Secondo" with value "3170".

Semestre	Energia prodotta
Primo	2150
Secondo	3170

Right Snippet: This shows a table used for layout. The code defines a table with four columns. The first column is labeled "Logo", the second "Banner", the third "Menu", and the fourth "Content". The fourth column is further divided into two sub-cells labeled "Footer1" and "Footer2".

Logo	Banner	Menu	Content
Footer1	Footer2		

Link ipertestuali

- ◆ Il **tag <a> (anchor)** viene utilizzato come contenitore di una stringa di testo o un'immagine che serve come collegamento ipertestuale/ipermediale
- ◆ Es. Collegamento a pippo
- ◆ **URL assoluti**
 - ◆ Sono composti da un identificatore di protocollo, il nome del server (host name) e il percorso (path) per identificare il file da linkare sul server
 - ◆ Es. contenuto
- ◆ **URL relativi**
 - ◆ Vengono utilizzati quando il documento puntato si trova sullo stesso server del documento corrente. Sono costituiti da puntatori che identificano la posizione del documento puntato rispetto al documento corrente.
 - ◆ Es. contenuto
 - ◆ Di default l'URL relativo è basato sul documento corrente, ma è possibile definire un **tag <base>** all'interno della sezione <head> per definire l'URL di base per tutti gli URL relativi del documento.

Link interni a un documento

- ◆ Il tag <a> viene utilizzato nella maniera più semplice per collegarsi a documenti, che vengono mostrati a partire dal margine superiore.
E' possibile tuttavia collegarsi a punti specifici di un documento, come aiuto alla navigazione.
Per fare questo sono necessarie due fasi:
 - ◆ Identificazione del frammento di documento da collegare
 - ◆ Attraverso l'attributo name del tag <a> (rimosso in XHTML 1.1)
 - ◆ Es. testo
 - ◆ Attraverso l'attributo id in qualsiasi tag (non supportato nei browser più vecchi)
 - ◆ Es. <h1 id="point1">Intestazione</h1>
 - ◆ In entrambi i casi i frammenti identificati non assumono stili di presentazione particolari; inoltre i valori utilizzati per name e id devono essere unici nel documento

Link interni a un documento

- ◆ **Link interno allo stesso documento**
 - ◆ Il collegamento viene effettuato utilizzando un tag a con attributo href, che ha come valore il nome del frammento preceduto dal simbolo #
 - ◆ Es. Vai al punto uno
- ◆ **Link interno ad un altro documento**
 - ◆ Link ad un frammento contenuto in un documento diverso, residente anche su altri server
 - ◆ In questo caso il simbolo # va preceduto dall'URL relativo o assoluto del documento:
 - ◆ Es. Vai al punto uno
 - ◆ Es. Vai al punto uno

Stabilire l'obiettivo del collegamento

- ❖ E' possibile pilotare la visualizzazione della pagina linkata su una finestra diversa da quella di partenza, utilizzando l'attributo **target**
 - ❖ Il valore attributo è liberamente assegnabile dall'utente; vengono tuttavia predefiniti alcuni valori, come **_blank**, **_self**, **_parent** e **_top**
 - ❖ Es. `...`
 - ❖ Es. `...`
 - ❖ Nel primo caso verrà aperta una finestra nuova per ogni link con target="**_blank**". Questo può causare una proliferazione dannosa di finestre, nel caso in cui ci siano un certo numero di link con questo target.
 - ❖ Un metodo migliore consiste nel dare alla finestra bersaglio un nome specifico. In questo modo verrà creata una sola finestra addizionale; i documenti pilotati dall'attributo target verranno pilotati sempre verso essa.
 - ❖ L'attributo target viene spesso utilizzato nell'ambito dei **frameset**
 - ❖ E' bene considerare anche i punti negativi di questa tecnica sul piano dell'accessibilità
 - ❖ Ad es. la finestra non compare in primo piano, nel caso in cui essa venga richiamata da un link che usa come target il nome della finestra stessa

Link non ipertestuali

- ❖ **Mail link**
 - ❖ Il browser deve essere configurato per supportare questo tag, associando al click sul mail link l'attivazione di un client di posta elettronica
 - ❖ Il valore da associare all'attributo href deve avere la seguente sintassi:
 - ❖ mailto:username@domain
 - ❖ ES. `manda un'e-mail a Fabio`
 - ❖ I browser più recenti supportano funzioni addizionali per compilare automaticamente l'oggetto, il testo o l'opzione CC
 - ❖ mailto:username@domain?subject=subject
 - ❖ mailto:username@domain?cc=person1
 - ❖ mailto:username@domain?body=body
 - ❖ Non c'è in tutti i casi nessuna garanzia che poi l'utente mantenga invariati questi valori

Link non ipertestuali

- ◆ Telnet link
 - ◆ Viene aperta una sessione di terminale su un client
 - ◆ La sintassi prevede la possibilità di indicare anche username e password, oltre che il nome e la porta di accesso al server, ma è una pratica ovviamente sconsigliata ...
 - ◆ `...`

Link non ipertestuali

- ◆ FTP link
 - ◆ Anche in questo caso valgono le considerazioni precedenti a proposito dell'opportunità di inserire username e password nel valore dell'attributo href
 - ◆ `...`
- ◆ Other links
 - ◆ Altri link possono essere utilizzati per l'accesso ad un news server o ad un servizio gopher.
 - ◆ `...`
 - ◆ `...`
 - ◆ La sintassi file:// si può trovare in situazioni dove l'unico riferimento possibile è quello del filesystem locale (ad es. Dreamweaver lo utilizza quando associamo un'immagine ad un file HTML che non è stato salvato almeno una volta e quindi ha una posizione indefinita nel filesystem)
 - ◆ `...`

Il tag <link>

- ◆ Il tag <link> viene utilizzato per definire una relazione tra il documento corrente e un altro documento esterno
- ◆ Possono esserci più tag <link> in uno stesso documento
- ◆ Attributi più importanti sono **href**, **rel**, **rev** e **type**
- ◆ Utilizzato soprattutto per riferirsi a fogli di stile esterni
 - ◆ Es. <head><link href="webstyle.css" rel="stylesheet" type="text/css"></head>
- ◆ Un altro uso è quello di riferimento ad una versione alternativa del documento in un altro linguaggio
 - ◆ <head><link href="traduzioni/francese.htm" rel="alternate" type="text/html" hreflang="fr"></head>