UNIVERSITÀ DEGLI STUDI
DI GENOVA

Ufficio sviluppo risorse umane e organizzazione - Ufficio dirigenziale affari generali e comunicazione - Delegato del Rettore per l'integrazione degli Studenti disabili - CSITA Centro Servizi Informatici e Telematici di Ateneo - Servizio Orientamento

Corso di formazione sull'accessibilità dei siti web

Il concetto di ipertesto e il documento HTML

revisione 1.0

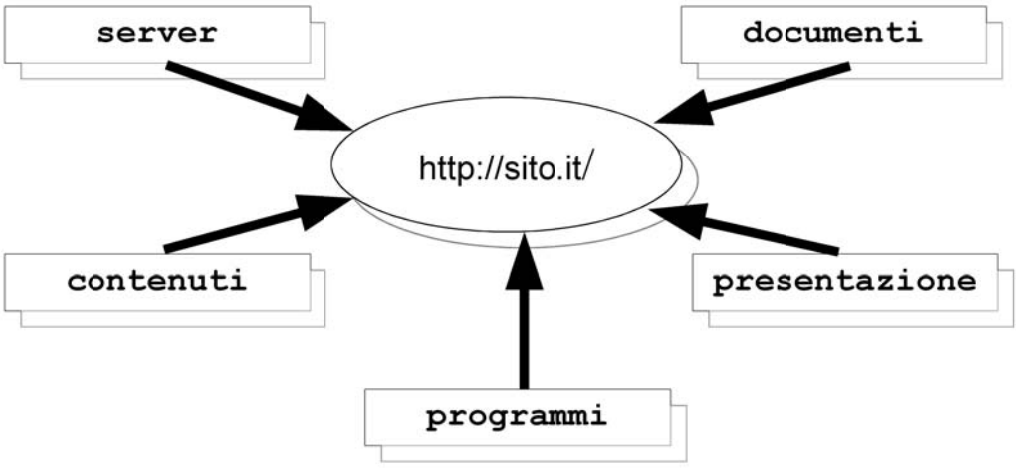
Marco Ferrante, CSITA

Nota di copyright/Disclaimer

Agenda

- Spegnere i cellulari ✓
- Elementi di un sito web
- Risorse, URL e URI
- Codifiche dei file
- I set di caratteri
- Documenti strutturati
- XML e XHTML

Elementi di un sito web



Agenda

- Elementi di un sito web ✓
- Risorse, URL e URI
- Codifiche dei file
- I set di caratteri
- Documenti strutturati
- XML e XHTML

Risorse

Una “risorsa” è un qualsiasi elemento dotato di identità:

- documenti
- immagini, contenuti multimediali, software
- sorgenti *stream*
- funzioni remote
- collezioni di risorse

Non tutte le risorse sono accessibile dalla rete

URI *Uniform Resource Identifier*

URI è una sintassi generica per identificare una risorsa (RFC 2396)

Due tipi:

- "Uniform Resource Locator" (URL)
- "Uniform Resource Name" (URN)

Sintassi generale:

- <schema>:[/ | //]<autorità><path>?<query>

La sintassi URI ha tra i suoi scopi principali la “trascrivibilità”, e ogni URI dovrebbe essere digitabile su una tastiera qualsiasi

URN *Uniform Resource Name*

Identificano in modo persistente una risorsa (ad esempio, nomi di file per documenti)

Sintassi (RFC 2141):

- urn: <NID> : <NSS>

Esempi:

- urn:mace:washington.edu:confocalMicroscope
- urn:nir:stato:decreto.legislativo:2005-08-19;192
- URN:ISBN:0395363411

Problemi di registrazione del NID (IANA)

- urn:oid:1.3.6.1.2.1.27

URL *Uniform Resource Locators*

Identificano una risorsa raggiungibile su Interne

Sintassi (RFC 1738):

- <schema>://<user>:<pw>@<host>:<porta>/<path>

Esempi:

- file://vms.host.edu/disk$user/my/notes/note12345.txt
- mailto:marco@csita.unige.it
- http://www.unige.it/

Università degli Studi di Genova
Corso di formazione sull'accessibilità dei siti web

Indirizzi web

Ogni risorsa di un sito web è identificata da un URL, detto comunemente “indirizzo”

- http://www.unige.it/regolamenti/org/statuto.shtml
- http://www.unige.it/

Indirizzi (quasi) equivalenti:

- http://130.251.21.166/staff/
- http://2197493158/staff/
- http://www.google.com@www.unige.it/staff/
- http://www.unige.it/regolamenti/../staff/
- http://www.unige.it/%73%74%61%66%66/

Agenda

- Elementi di un sito web ✓
- Risorse, URL e URI ✓
- Codifiche dei file
- I set di caratteri
- Documenti strutturati
- XML e XHTML

Cos'è un file

Un file è (approssivativamente) una sequenza di numeri compresi tra 0 e 255

Il modo in cui questi numeri vanno interpretati è detto "formato" o "protocollo"

- se ad ogni numero sostituisco una lettera, ottengo un file di testo (.txt o ASCII)
- se ogni numero corrisponde un colore, ottengo un'immagine
- se vengono ordinati, corrisponde ad una tabella

la maggior parte dei formati usa strutture complesse

Riconoscere i file

L'interpretazione corretta del contenuto di un file è necessaria per poterlo trattare nel modo più adatto

Da una sequenza tipo:

```
47 49 46 38 39 61 7D 00 72 00 F7 00 00 06 0C 04
00 14 0C 08 10 10 10 08 08 10 10 00 10 10 08 10
10 10 10 18 00 08 18 08 10 18 08 08 18 10 ...
```

come posso dedurre il formato del file?

- da qualche sua proprietà intrinseca
- da qualche sua proprietà contestuale
- perché qualcuno “me lo dice”

Università degli Studi di Genova
Corso di formazione sull'accessibilità dei siti web

Proprietà intrinseche

Un file può dichiarare il proprio formato nei propri byte (usualmente iniziali):

- **47 49 46 38 39 61 7D 00** ⇔ GIF89a}_
- **25 50 44 46 2D 31 2E 34 0D** ⇔ %PDF-1.4↵
- **CA FE BA BE ⇔ classi Java**
- <?xml version=”1.0” encoding=”UTF-8”?>
- MZ ⇔ Eseguibili DOS 2.x
- **7F 45 4C 46** ⇔ .ELF

Questo meccanismo è noto come “magic number” o String ID. Vedi /usr/share/misc/magic

Proprietà contestuali

Quando un file è memorizzato su un dispositivo, gli viene assegnato un nome e altre proprietà

Casi tipico:

- file1.zip, file7.jar
- file2.doc
  - MS Word, FrameMaker, WordStar, WordPerfect, ecc...
- file3.html, file4.htm, file5.shtml, file6.jsp, ecc...

I sistemi Unix usano un apposito marcatore (bit x) per identificare i file eseguibili

MacOS X usa gli attributi estesi dei file per memorizzarne il tipo MIME

Tipi MIME

La *trasmissione* di file ha introdotto nuovi problemi

- mancanza di un organismo normatore
- perdita di informazioni contestuali

I meccanismi di trasmissione trasportano anche le informazioni:

```
--------------070409090403070609000803
Content-Type: image/jpeg;
 name="lowres.jpg"
Content-Transfer-Encoding: base64
Content-Disposition: inline;
 filename="lowres.jpg"

/9j/4AAQSkZJRgABAQEASABIAAD/2wBDAAYEBQYFBAYGBQY...
GBcUFhYaHSUfGhsjHBYWICwgIyYnKSopGR8tMC0oMCUoKSj...
```

Agenda

- Elementi di un sito web ✓
- Risorse, URL e URI ✓
- Codifiche dei file ✓
- I set di caratteri
- Documenti strutturati
- XML e XHTML

I formati dei file di testo

Tradizionalmente, i file sono classificati come:

- file di testo (*plain text*)
- file binari

I file plain text non contengono codici non stampabili e hanno la struttura interna ordinata come il testo visualizzato.

Possono contenere informazioni di formattazione, come HTML e RTF

Codifica dei caratteri

La codifica è l'associazione tra un numero e un glifo o tra un numero e un carattere

L'aspetto che prenderà la mappa tra numero e carattere dipende da:

- sistema operativo
- architettura del processore
- set di caratteri installato sul client
- ...

Nell'Internet del XXI secolo, *non esiste nulla che corrisponde al plain text*

La codifica deve quindi sempre essere esplicitata

I testi ASCII

Il formato ASCII è nato negli anni '60 per le telescriventi

Usa 7 bit (codici da 0 a 127) per definire i glifi dei caratteri

I primi 32 codici e l'ultimo corrispondono a controlli

```
 !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~
```

Usi della codifica ASCII

La codifica ASCII non contiene informazioni sulla struttura o sulla presentazione (escluso i <cr>)

Il formato ASCII è usato per file di configurazione, codice sorgente (non Java) ed è alla base di

- linguaggi di markup: SGML, HTML, XML
- formati complessi: PostScript, RTF
- altre codifiche: MIME

I set ASCII estesi

Con la diffusione dei PC, si pensò di usare i 128 caratteri lasciati vuoti per codificare i caratteri della lingua nativa dell'utente. Ogni set di codifiche è associata ad un codice (*page code*)

- CP437: 0x80 (128) ⇔ Ç
- CP855: 0x80 ⇔ ђ, CP866: 0x80 ⇔ А

ISO/IEC standardizzo la proposta 8859-1, ±adatta a tutti le lingue occidentali, che divenne subito:

- ISO-8859-1 (iso-latin-1): 0x80 ⇔ n.u., 0xA4 ⇔ ¤
- Windows-1252: 0x80 ⇔ €, 0xA4 ⇔ ¤
- ISO-8859-15: 0x80 ⇔ n.u., 0xA4 ⇔ €

Unicode

Con Unicode, ISO ha cambiato approccio, prevalentemente per consentire testi multilingua.

A ogni carattere (non glifo) è associato un codice numerico (*codepoint*):

- ASCII: codice 0x41: A (in tutte le lingue)
- Unicode: *codepoint* U+0048: “a maiuscola lingue occidentali”
- Unicode: *codepoint* U+0391: “alfa maiuscola greco”

Per i testi in italiano non cambia molto

Esempi *codepoint* Unicode

codepoint U+041F “p cirillico”

- maiuscolo П
- minuscolo п
- russo corsivo *п*
- serbo corsivo *ӣ*

codepoint U+03A0 “p greco”

- maiuscolo Π
- minuscolo π

Codifiche di Unicode

Unicode definisce astrattamente i caratteri

Per l'uso effettivo devono essere codificati:

- UCS-2 o UTF-16: un carattere è rappresentato da due byte. Meccanismo nativo di Windows NT e sup.
- UTF-8: in Unicode, i *codepoint* dei caratteri ASCII sono uguali ai loro codici. I caratteri fino a 127 sono codifica a un byte, gli altri con un numero crescent di byte

	UTF-16	UTF-8
000000-00007F	00000000 0xxxxxxx	0xxxxxxx
000080-0007FF	00000xxx xxxxxxxx	110xxxxx 10xxxxxx

Esisto altre rappresentazioni: UTF-7, UCS-4, ...

BOM *Byte Order Mark*

UCS-2 pone il problema dell'ordine

“Hello” si codifica:

- U+0048 U+0065 U+006C U+006C U+006F

che si può scrivere (a seconda dell'architettura)

- 00 48 00 65 00 6C 00 6C 00 6F
- 48 00 65 00 6C 00 6C 00 6F 00

Per distinguere, i primi due byte del file possono essere usati per indicare l'ordine (BOM)

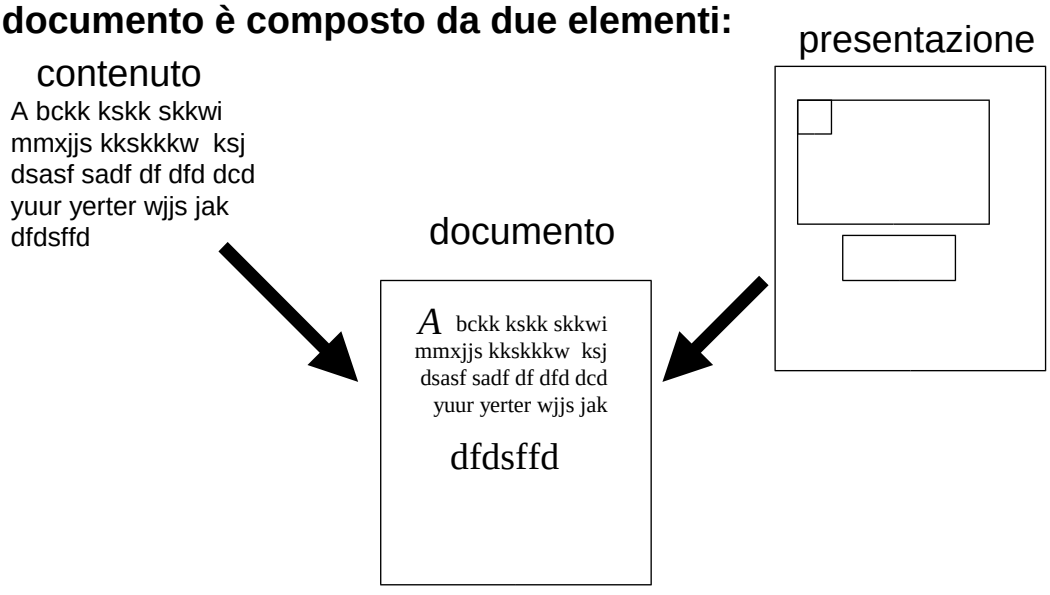
- *codepoint* U+FEFF "zero-width no-break space"
- FEFF su architetture Big Endian (Sparc, 68000)
- FFEF su architetture Little Endian (Intel x86)

Agenda

- Elementi di un sito web ✓
- Risorse, URL e URI ✓
- Codifiche dei file ✓
- I set di caratteri ✓
- Documenti strutturati
- XML e XHTML

Cos'è un documento (elettronico)

Un documento è composto da due elementi:



La struttura logica dei documenti elettronici

Nei documenti elettronici, a differenza della macchina da scrivere, viene enfatizzata la struttura logica del documento

Alcuni formati mischiano struttura logica e aspetto, privilegiando l'aspetto (Word)

Altri privilegiano la struttura (HTML)

Altri sono puro contenuto (ASCII e XML)

La strutturazione logica permette di trattare automaticamente il contenuto del documento

Registrare dati e formattazione

Esistono diverse tecniche per memorizzare sia il contenuto sia l'aspetto di un documento:

- formati binari
  - uso di codici estesi: MS Word
  - tabella dei contenuti: pdf
- uso particolare di codici standard
  - markup descrittivo: SGML, XML, HTML
  - markup procedurale: T_EX, LaTeX
- file separati
  - fogli di stile: CSS, XSL
  - oggetti di formattazione: FO, DSSSL

Linguaggi di *markup*

Nei sistemi di *markup*, elementi ordinari possono assumere un significato speciale (marcatori)

RTF

- {\rtf Ciao!\par Testo in {\b grassetto}.\par}

XML

- "&" e ";" identificano gli elementi speciali del testo
- "<", ">" racchiudono gli elementi strutturali (*tag*)
- non è definita la presentazione degli elementi

HTML

- i *tag* sono predefiniti

Organizzazione del contenuto

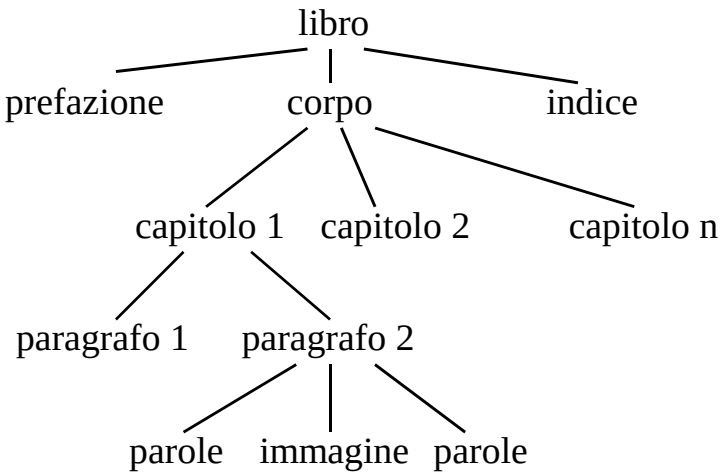
Il contenuto di un documento può essere suddiviso (strutturato) in parti logiche

- in una pagina di un giornale ci sono titoli, articoli, foto, pubblicità, ecc…
- nei documenti d'identità ci sono nome, cognome, indirizzo, ecc…
- in un bando ci sono i riferimenti normativi, la scadenza, i requisiti, il firmatario, ecc…

Non sempre i programmi permettono di gestire la struttura logica del documento prescindendo dall'aspetto

Struttura gerarchica dei documenti

La struttura di un documento si può generalmente rappresentare con un albero:



Agenda

- Elementi di un sito web ✓
- Risorse, URL e URI ✓
- Codifiche dei file ✓
- I set di caratteri ✓
- Documenti strutturati ✓
- XML e XHTML

XML *eXtensible Markup Language*

Un documento XML è un file *plain text*

Se la codifica non è esplicita, si considera UTF-8

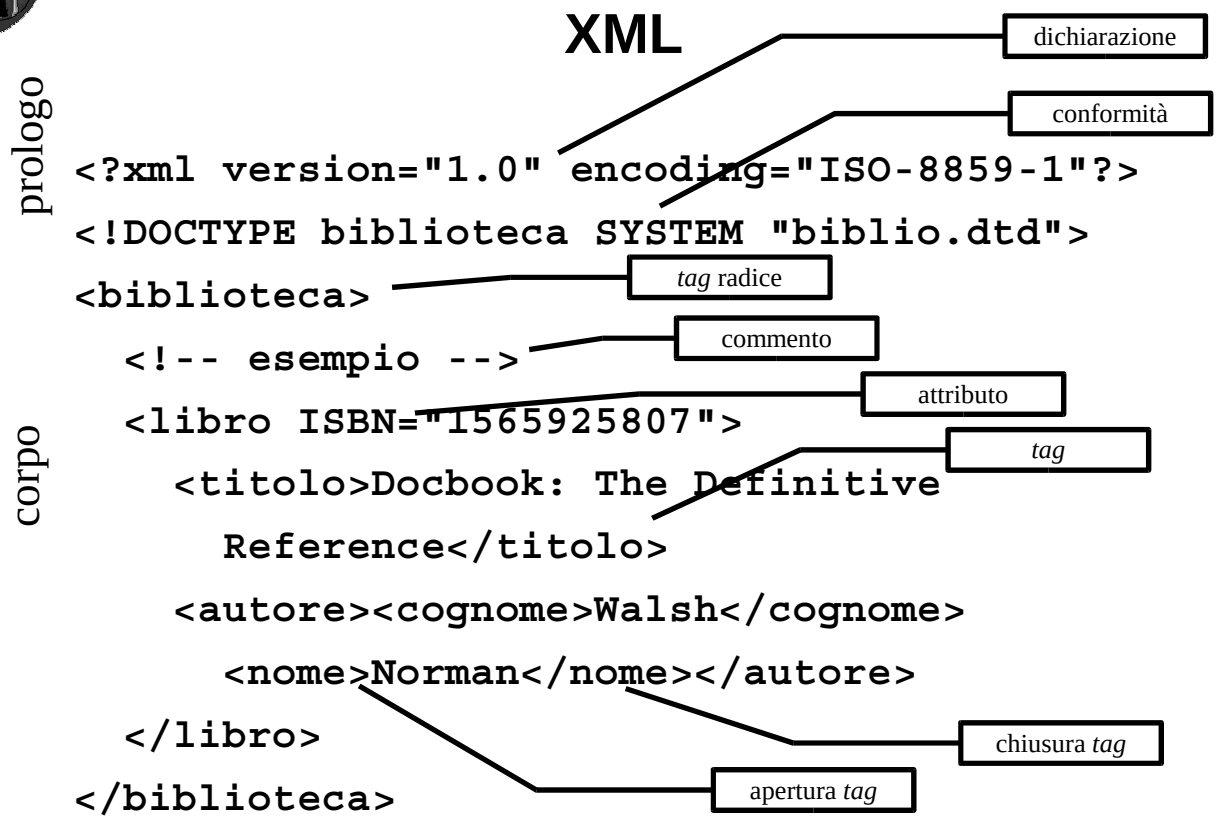
I caratteri:

- <, > e &

hanno un significato speciale e devono essere rimpiazzati da:

- < > &

Gli spazi vengono “normalizzati”



Benefici dell'XML

I documenti XML possono essere validati automaticamente

I documenti XML possono essere generati trasformando altri documenti XML, per produrre copie organizzate diversamente

XML si può convertire per differenti media

XML permette di inserire elementi calcolati

Esistono diversi linguaggi di interrogazione e estrazione dei dati

Da XML a HTML

Nelle specifiche più recenti, HTML è formalizzato come un dialetto XML (XHTML)

Rispetto a “spaghetti-HTML”, occorre correggere parecchi elementi

- Definire il tipo di documento
- Sostituire alcuni *tag*
- Convertire i *tag* in minuscolo
- Delimitare gli attributi
- Chiudere i *tag* e evitare sovrapposizioni

Dichiarazione

Il documento deovrebbe iniziare con

- <?xml version=”1.0” ?>

Opzionalmente, ma raccomandato, può definire la codifica:

- <?xml version=”1.0” encoding=”UTF-8” ?>

Opzionalmente, ma consigliato, può definire la lingua e il *namespace* nell'elemento radice:

- <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="it" lang="it">

Definizione DTD

Definire il tipo di documento:

- <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
- <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

HTML 4.01

Definizione del DTD

- <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

Codifica

- <meta http-equiv="Content-type" content="text/html; charset=ISO-8859-1" />

Lingua

- <html lang=”it”>

Tag 4.01 deprecati

Alcuni *tag* HTML 4.01 non sono definiti nei DTD

Transitional:

- frame, frameset

Strict:

- applet, basefont, center, dir, font, iframe, isindex, menu, noframes, s, strike, *u*

Se utilizzati, la pagina non è valida e quindi non può essere trattata automaticamente

Sintassi *tag*

In XHTML, tutti i *tag* devono essere scritti in minuscolo

- `<BODY>` ✘
- `<Body>` ✘
- `<body>` ✓

In XML, i nomi di *tag* e attributi sono *case sensitive*

Sintassi attributi

Tutti gli attributi devono essere racchiusi tra ' (apice) o “ (virgolette)

- `<input type=checkbox name=b1 checked>` ✘
- `<input type=”checkbox” name=”b1” checked>` ✘
- `<input type=”checkbox” name='b1' checked=”1”>` ✓

In XML, è accettabile:

- <persona nome=”Dall'Orto”>

Le virgolette non sono tutte uguali!

Chiusura dei tag

In XHTML tutti i tag devono chiudersi

```
<ul>
  <li><img src="/logo.gif"> ✘
</ul>

<ul>
  <li><img src="/logo.gif"></li> ✘
</ul>

<ul>
  <li><img src="/logo.gif" /></li> ✓
</ul>
```


Sovrapposizione *tag*

In XHTML, i *tag* devono essere annidati e non possono mai sovrapporsi (*overlapping*)

- `Questo &egrave; un testo <i>corsivo e <b>grassetto</i></b>` ✘
- `Questo &egrave; un testo <i>corsivo e <b>grassetto</b></i>` ✓
- `Questo &egrave; un testo <i>corsivo e</i> <b><i>grassetto</i></b>` ✓

JavaScript

Gli script vanno inseriti in sezioni PCDATA

```
<script type="text/javascript">
<![CDATA[
  ... codice JavaScript ...
]]>
</script>
```

L'attributo *name* va sostituito con *id*

PHP e XML

Scrivendo:

- <?xml version="1.0"' ?>

l'interprete PHP cercherà di elaborare il contenuto

Per produrre documenti XML validi da PHP, occorre, in alternativa:

- produrre la dichiarazione da script
- usare i tag PHP lunghi
  - short_open_tag = 1
  - <?php ... ?>

Il modello ad oggetti XHTML

I documenti XML/XHTML ben formati possono essere rappresentati con un albero

Uno stile (font, colore, ecc…) applicato ad un elemento può propagarsi agli elementi in esso contenuti

I fogli di stile (CSS Cascade Style Sheet) usano questa proprietà per definire gli stili all'esterno delle pagine

È il principio è utilizzato dagli script JavaScript per individuare gli elementi nella pagina

Passo a XHTML! E in cambio?

I documenti XHTML possono essere validati automaticamente

I documenti XHTML possono essere generati da altri documenti XML

Posso usare trasformazioni XSL per convertire il contenuto per differenti media

Potenziali problemi

I browser meno recenti possono aver difficoltà a:

- nascondere il prologo
- trattare i tag che in HTML 3.x non avevano chiusura

Alcuni elementi non sono disponibili in XHTML

I form secondo specifiche XHTML non sempre si comportano omogeneamente tra browser

La codifica UTF-8 può essere problematica con editor su sistemi Windows.

Pulizia automatica

Alcuni strumenti utili nel passaggio da HTML a XHTML:

- Tidy: chiude i tag, li ordina, indenta il codice, ecc...
  - http://tidy.sourceforge.net/
  - http://www.w3.org/People/Raggett/tidy/

Validatori DTD

Motori XSLT

- Apache Xalan

Agenda

- Elementi di un sito web ✓
- Risorse, URL e URI ✓
- Codifiche dei file ✓
- I set di caratteri ✓
- Documenti strutturati ✓
- XML e XHTML ✓
- domande e discussione