

Fogli di stile

- I fogli di stile permettono la separazione della forma dal contenuto.
- Il concetto di foglio di stile è quello di creare un documento separato che contenga tutte le informazioni necessarie per la formattazione o la trasformazione di qualsiasi tipo di dati a cui si decida di applicarlo.

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Fogli di stile (2)

- Vantaggi:
 - Risparmio di tempo e maggior praticità di gestione del sito (se si desidera infatti modificare l'aspetto di un sito *web* non è necessario effettuare le correzioni in ogni pagina ma è sufficiente modificare il foglio di stile per rendere effettivi i cambiamenti).
 - Personalizzazione grafica, in quanto è l'aspetto degli elementi è modificabile.
 - Utilizzo di un unico foglio di stile per formattare più documenti.
 - Possibilità di adattare la formattazione al tipo di utente e *device* che ha accesso ai dati.

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Fogli di stile (3)

- I fogli di stile oggi utilizzati sono essenzialmente di tre tipi:
 - **CSS** (*Cascading Style Sheet*), utilizzati soprattutto per HTML.
 - **XSL** (*Extensible Stylesheet Language*), utilizzati per formattare un documento XML destinato alla stampa.
 - **XSLT** (*Extensible Stylesheet Language Transformation*), utilizzati per la trasformazione di un documento XML.

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

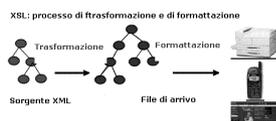
XSL

- **XSL** è un nuovo linguaggio creato per definire fogli di stile che definiscano la presentazione del documento. Il suo utilizzo implica l'esistenza di un processore XSL che codifica la sorgente XML nel modo definito dal foglio di stile.

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

XSL (2)



Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

XSL (3)

- Per ottimizzare la visualizzazione di un file XML su dispositivi differenti. Il processo di costruzione di un nuovo documento si articola in due passaggi:
 - **Trasformazione dell'albero:** costruire un albero di arrivo a partire dall'albero di partenza; la trasformazione dell'albero permette la creazione di documenti significativamente differenti rispetto al documento di base (permettendo ad esempio la trasformazione in linguaggi differenti). Della trasformazione dell'albero si occupa un sottolinguaggio chiamato XSLT.
 - **Formattazione:** i nodi dell'albero sono considerato come oggetti ai quali vengono applicate regole di formattazione.

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

XSL (4)

- Data la sostanziale indipendenza di queste operazioni, il W3C decise di separarle in due parti:
 - XSLT per definire le trasformazioni.
 - XSL per formattare il risultato.

XSL:fo

- XSL Formatting Object è stato creato per fornire agli sviluppatori uno strumento agile per controllare in modo preciso l'impaginazione e la disposizione della pagina, senza l'utilizzo di elementi complessi (ad esempio i frame).
- XSL:fo è simile a CSS per le proprietà utilizzate, anche se spesso le estende per garantire maggiore controllo.

XSL:fo (2)

- Esistono quattro sottoclassi di proprietà XSL:fo:
 - Proprietà interamente prese da CSS.
 - Proprietà CSS estese con altri valori.
 - Proprietà CSS suddivise.
 - Proprietà originali XSL.

XSLT

- XSLT consente di trasformare un albero iniziale XML (*source tree*) in un altro albero (*result tree*) i cui elementi possono differire molto da quelli iniziali.
- Generalmente viene utilizzato per operare trasformazioni tra linguaggi diversi (ad esempio da XML a HTML) o per codificare in modo automatico un documento basato su un linguaggio troppo complesso (ad esempio MathML).
- E' quindi sostanzialmente definibile come un linguaggio ad alto livello, in grado di navigare per la struttura del documento e di accedere ai nodi.

XSLT: sintassi

- I fogli di stile XSLT hanno una struttura modulare per permettere di riutilizzare i moduli, ricombinandoli in modo differente a seconda delle necessità.
- Un foglio di stile XSLT si apre con una dichiarazione che identifica il tipo di documento e lo spazio dei nomi utilizzato:
 - `<xsl:stylesheet xmlns:xsl=http://www.w3.org/1999/XSL/Transform version="1.0">`

XSLT: sintassi (2)

- Dopo la dichiarazione iniziale si trovano gli elementi di primo livello.
- L'elemento di primo livello deve necessariamente essere uno tra:
 - `<xsl:attribute-set>`
 - `<xsl:decimal-format>`
 - `<xsl:import>`
 - `<xsl:include>`
 - `<xsl:key>`
 - `<xsl:namespace-alias>`
 - `<xsl:output>`
 - `<xsl:param>`
 - `<xsl:preserve-space>`
 - `<xsl:strip-space>`
 - `<xsl:template>`
 - `<xsl:variable>`

XSLT: sintassi (3)

- E' anche possibile avere degli elementi di primo livello definiti da sviluppatori o dagli utenti. In generali questo tipo di elementi, che si riferiscono a uno spazio dei nomi differente, sono definiti dagli sviluppatori di processori XSL, che creano degli elementi proprietari utili per eseguire determinate funzioni complesse.

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

XSLT: trasformazione

- Il processo di trasformazione avviene tramite l'applicazione di regole agli elementi dell'albero. Ogni regola è definita dall'elemento `<xsl:template>` che possiede un attributo `match` che identifica l'elemento dell'albero a cui va applicata la regola.
- Ogni regola possiede poi un corpo, chiamato *template body*, nel quale sono espresse le istruzioni da eseguire e un elemento di chiusura (`</xsl:template>`).

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

XSLT: elementi

- Elementi utilizzati per definire le regole e controllare il modo in cui sono invocate
 - `<xsl:template>`
 - `<xsl:apply-template>`
 - `<xsl:call-template>`
- Elementi che definiscono la struttura del foglio di stile
 - `<xsl:stylesheet>`
 - `<xsl:include>`
 - `<xsl:import>`

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

XSLT: elementi (2)

- Elementi usati per generare un *output*
 - `<xsl:value-of>`
 - `<xsl:element>`
 - `<xsl:attribute>`
 - `<xsl:comment>`
 - `<xsl:processing-instruction>`
 - `<xsl:text>`
- Elementi utilizzati per definire variabili e parametri
 - `<xsl:variable>`
 - `<xsl:param>`
 - `<xsl:with-param>`

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

XSLT: elementi (3)

- Elementi utilizzati per copiare informazioni dal documento sorgente al documento destinazione
 - `<xsl:copy>`
 - `<xsl:copy-of>`
- Elementi utilizzati per creare istruzioni e processi condizionali e iterazioni
 - `<xsl:if>`
 - `<xsl:choose>`
 - `<xsl:when>`
 - `<xsl:otherwise>`
 - `<xsl:for-each>`

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

XSLT: elementi (4)

- Elementi utilizzati per controllare l'ordine e la numerazione
 - `<xsl:sort>`
 - `<xsl:number>`
- Elementi utilizzati per controllare il formato del documento *output*
 - `<xsl:output>`

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Microsoft e XSLT

- Microsoft ha aggiunto un processore XSL, chiamato MSXML, al suo browser fin dal 1998, anno della prima proposta pubblica relativa a XSL. Con l'avvento, a fine 1999, di XSLT, Microsoft ha però mantenuto invariato il processore interno a Internet Explorer, causando così seri problemi di compatibilità.
- Alcune delle discrepanze fondamentali sono:
 - Per inserire un foglio di stile all'interno di un documento, Internet Explorer richiede la dichiarazione `<?xml:stylesheet type='text/xsl' href='test.xsl' ?>` mentre la dichiarazione standard, secondo le specifiche W3C richiederebbe di assegnare come valore all'attributo `type` "text/xml".

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Microsoft e XSLT (2)

- All'inizio del foglio di stile XSL è necessario usare come spazio dei nomi `xmlns:xsl="http://www.w3.org/TR/WD-xsl"` invece che `xmlns:xsl=http://www.w3.org/1999/XSL/Transform` come previsto dal W3C.
- E' inoltre carente il supporto per le nuove istruzioni.

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Processori alternativi

- Esistono processori disponibili a titolo gratuito che garantiscono un supporto elevato e conforme alle specifiche W3C.
- Tra questi:
 - Sablotron (www.gingerall.com)
 - XT (www.jclark.com)
 - Xalan (xml.apache.org)
 - Expat (www.jclark.com)

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Xpath, Xpointer e Xlink

- Sono tre meccanismi correlati per il puntamento all'interno di un file XML:
- XPATH permette di navigare tra i nodi di un documento;
- XPOINTER utilizza XPATH per generare link HTML;
- XLINK utilizza XPOINTER per generare link HTML estesi.
- Rispetto ai link HTML permettono di:
 - Esprimere riferimenti relativi
 - Definire link a parti di albero.
 - Definire link con multiple sorgenti e destinazioni .

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Xpointer

- Permette di definire indirizzamenti relativi
- Permette di puntare a interi frammenti di albero senza che sia presente un'ancora.
- E' utilizzato da XLINK per individuare risorse remote.
 - URI:
`http://www.foo.org/bar.xml#xpointer(article/section[position() <= 5])`
 - Fragment identifier:
`#xpointer(article/section[position() <= 5])`
 - XPointer expression: `(article/section[position() <= 5])`

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Xpointer: funzionamento

- I fragment identifiers consistono in:
 - Un elemento denotato da un ID (identificato univocamente nel documento)
 - Una sequenza di numeri che definiscono il percorso dall'elemento radice all'elemento ID.
 - Una sequenza di termini `xpointer(...)` `xpointer(...)` ... che contengono una lista di espressioni Xpointer (XPATH).

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Xpath

- XPATH è un linguaggio dichiarativo utilizzato per l'indirizzamento (Xpointer) e per il pattern matching (XSLT).
- Il costrutto principale è chiamato **location path**, costituito da una sequenza di **location steps** separati da /.
- Ogni **location step** è valutato rispetto al suo contesto.
- **Location path** è valutato composizionalmente, da sinistra a destra.

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Location step

- Ogni **location step** ha la forma:
axis :: node-test [predicate]
- **axis** seleziona un insieme di nodi candidati.
- **node-test** effettua un primo filtro dei nodi basandosi sul loro tipo (chardata, processing instruction, etc.), o nome.
- **predicates** (zero o più) effettua il filtro più complesso
- I "nodi candidati" che "sopravvivono" costituiscono il risultato dell'espressione.

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Xpath: esempio

- Seleziona tutti gli attributi href negli elementi cite nelle prime 5 sezioni di un articolo.

```
child::section[position() < 6]/descendant::cite  
/attribute::href
```

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Xpath: axes

- **Axes disponibili:**
 - **child** seleziona i figli del nodo;
 - **descendant** seleziona tutti i discendenti;
 - **parent** seleziona i genitori;
 - **ancestor** seleziona tutti i discendenti dal nodo radice ai genitori;
 - **following-sibling** seleziona i gemelli a destra;
 - **preceding-sibling** seleziona i gemelli a sinistra;
 - **following** seleziona tutti i nodi successivi;
 - **preceding** seleziona tutti i nodi precedenti;
 - **attribute** seleziona gli attributi del nodo;
 - **namespace** seleziona la dichiarazione dello spazio dei nomi;
 - **self** seleziona il nodo stesso;
 - **descendant-or-self** seleziona l'unione di descendant e self;
 - **ancestor-or-self** seleziona l'unione di ancestor e self.

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Xpath: node-test

- E' possibile effettuare un test dei nodi per:
 - **Tipo:**
 - **text()** nodi carattere;
 - **comment()** nodi commento;
 - **processing-instruction()** nodi istruzione;
 - **node()** tutti i nodi.
 - **Nome:**
 - **name** nodi con un determinato nome
 - ***** qualsiasi nodo

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Xpath: predicati

- I predicati consistono essenzialmente in espressioni da valutare.
- Possono essere:
 - *exp -> \$variable*
 - *| (exp)*
 - *| literal*
 - *| numeral*
 - *| function (arguments)*
 - *| boolean-expression*
 - *| numerical-expression*

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Xpath per ricette

- `//ingredient[@name="flour"]/@amount`
 - 4
 - 0.5
 - 3
 - 0.25
- `//ingredient[@amount='0.5' and @unit='cup']/@name`
 - grated Pamesan cheese
 - shredded mozzarella cheese
 - shortening
 - flour
 - orange juice

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Xpath per ricette (2)

- `//ingredient[@name="stock"]/preparation/step[position()=3]/text()`
 - When the liquid is relatively clear, add the carrots, celery, whole onion,
 - bay leaf, parsley, peppercorns and salt. Reduce the heat, cover and let
 - simmer at least 2 hours to make a hearty stock.

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica

Xpointer

- XPOINTER permette di indirizzare non solo nodi ma frammenti di albero, nello specifico permette di indirizzare un qualsiasi punto del documento, che è costituito da un qualsiasi elemento che precede o segue un carattere in un nodo.
- Seleziona il primo carattere di tutti i nodi caattere del documento.
`/descendant::text()/point()[position()=0]`
- Seleziona tutto dall'inizio della prima sezione al termine della terza.
`/section[1] to /section[3]`

Andres Baravalle – Vitaveska Lanfranchi

Dipartimento di Informatica