

HTML

HTML, markup language, usa tag per aggiungere informazioni al contenuto di un documento

- molti tag forniscono *informazioni di visualizzazione* dei dati (
,)
- alcuni specificano il *significato* dei dati (<a>, ...)
- i browser sanno come interpretare i tag
- MA: **non si possono aggiungere nuovi tag** (es: per definire un nuovo stile di visualizzazione dei contenuti)

⇒ serve un linguaggio di markup estensibile per definire il proprio linguaggio di descrizione dei dati

XML - Extensible Markup Language - I

- **linguaggio di markup** testuale
- **versione semplificata di SGML** (Standard Generalized Markup Language), linguaggio usato per gestione documenti (annotazioni, ...)
- **standard W3C** (WWW Consortium: <http://www.w3.org>) per rappresentazione di **documenti e dati, e scambio di dati nel Web**
- alcuni **obiettivi di design** di XML:
 - direttamente utilizzabile su internet, supportare un'ampia gamma di applicazioni, compatibile con SGML, human-readable, facilmente processabile da programmi,

XML - Extensible Markup Language - II

- linguaggio mediante cui si possono **definire altri linguaggi**
 - es: XHTML, SMIL (Synchronized multimedia Integration Language), WSDL (Web Services Definition Language), ...
- **estensibile**
 - permette **creazione di nuovi tag** (non necessariamente legati a visualizzazione di informazioni)
- **strutturato**
 - mediante tag si può rappresentare la struttura delle informazioni
- **validante**
 - permette di definire **regole grammaticali** per verificare la correttezza sintattica delle informazioni rappresentate in XML (DTD e XML schema)

Esempio

Consideriamo la descrizione di una pizza

peperoni (extracheese)
8.00
large

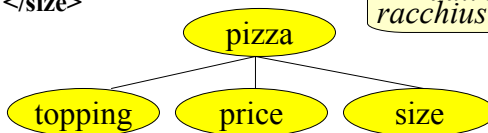
Non strutturata

La descrizione può essere **strutturata** come segue:

```
<pizza>  
  <topping extracheese='yes'> peperoni </topping>  
  <price> 8.00 </price>  
  <size> large </size>  
</pizza>
```

*Attributo di
elemento*

*I dati sono
racchiusi tra tag*



XML - Extensible Markup Language - III

- XML si è imposto come standard per condivisione di dati su internet
- Basato su rappresentazione dei dati come stringhe di caratteri ⇒ facilmente trasferibili via HTTP
- Essendo standard *de facto* sono stati sviluppati strumenti per scrittura, interpretazione, gestione di documenti XML

```
<piatti>
  <pizza>
    <tipo> margherita </tipo>
    <prezzo> 8.00 </prezzo>
    <grandezza> media </grandezza >
  </pizza>
  <pasta>
    <tipo> spaghetti al pesto </tipo>
    <prezzo> 10.00 </prezzo>
    <grandezza > normal </grandezza >
  </pasta>
</piatti>
```

Documenti XML

- composti di entità (unità fisiche di memorizzazione) che contengono dati analizzabili sintatticamente o meno (codifica UNICODE dei caratteri)
- **PCDATA** (*Parsed Character Data*): dati analizzabili sintatticamente - contengono caratteri
 - dati di tipo carattere
 - simboli di markup (tag di apertura e chiusura, commenti, ...; descrivono la struttura e le proprietà dei documenti)
- **CDATA** (*unparsed data*): dati non analizzabili sintatticamente - possono contenere testo, codice di scripting (e.g., Javascript, VBscript) o altri tipi di dati. Il contenuto di CDATA non è interpretato secondo la sintassi di XML

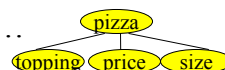
CDATA e PCDATA

```
<?xml version='1.0' encoding='utf-8'?>
<!-- commento... -->
<priceList>
  <coffee>
    <name> Mocha Java </name>
    <price> 11.95 </price>
  </coffee>
  <coffee>
    <name> Sumatra </name>
    <price> 12.50 </price>
  </coffee>
  <![CDATA[
    // qui posso inserire per esempio del codice di programmazione
  ]]>
</priceList>
```

Documenti XML ben formati - I

• *Documento ::= prologo elemento Misc**

- **prologo**: specifica la versione di XML e i caratteri usati. Es:
 - <?xml version='1.0' encoding='utf-8'?>
- **elemento**: corrisponde ad una componente logica del documento XML e viene rappresentato mediante tag
 - *elemento: EmptyElementTag | StartTag content EndTag*
 - ogni documento ha un elemento principale, la radice, non incluso in alcun altro elemento
 - un elemento può essere composto da altri elementi (*contenitore*) – struttura gerarchica (albero)
- **Misc**: commenti, processing instructions, ...



- 1 documento XML **deve** contenere **prologo** e elemento **radice**

Documenti XML ben formati - II

- Intuitivamente, documento è ben formato *sse*
 - contiene tutte le componenti necessarie (1 prologo, 1 solo elemento radice, ...)
 - i suoi tag sono bilanciati. E.g.
 - ogni tag aperto viene chiuso (ad eccezione di tag vuoti, che possono essere chiusi direttamente: `<t></t>` è equivalente a `</t>`)
 - i tag sono annidati in modo che l'ultimo tag aperto sia il primo chiuso: `<t1> <t2> <t3></t3> </t2> </t1>`
 - ...
- XML è *case-sensitive*

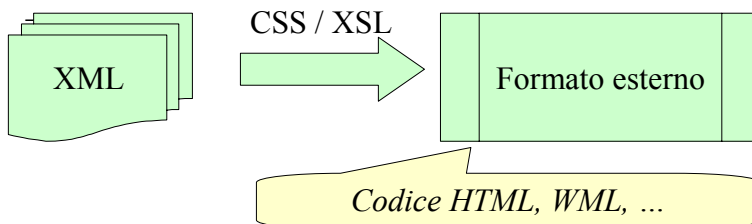
coffee.xml: ben formato e non

```
<?xml version='1.0' encoding='utf-8'?>
<!-- commento... -->
<priceList>
  <coffee>
    <name> Mocha Java </name>
    <price> 11.95 </price>
  </coffee>
  <coffee>
    <name> Sumatra </name>
    <price> 12.50 </price>
  </coffee>
</priceList>
```

```
<?xml version='1.0' encoding='utf-8'?>
<commento... --> ←
<priceList>
  </coffee> ←
  <name> Mocha Java ←
    <price> 11.95 </price>
  </coffee>
  <coffee>
    <names> Sumatra </name> ←
    <price> 12.50 </price>
  </coffee>
</priceList>
```

XML – Esempi di uso - I

- Rappresentazione interna di dati e documenti, visualizzabile in modalità diverse di contenuti (formati esterni diversi, selezioni diverse di sottoinsiemi di contenuto, ...)
 - Caratteristiche di prodotto, documentazione, ...
 - Formato interno dell'interfaccia utente di applicazione



XML – Esempi di uso - II

- Condivisione di dati tra applicazioni
 - Formato comune di rappresentazione dei dati
 - Eventuale trasformazione dei dati per tradurre dal formato di un'applicazione al formato di un'altra, basata su tecnologie standard
- Rappresentazione del contenuto di un messaggio (e.g., SOAP) inviato da applicazione ad applicazione
 - e.g., ordine di acquisto, dati relativi a clienti, ...
 - ...

Fogli Stile – CSS - CENNI

XML permette di distinguere contenuto di documento da presentazione

- **CSS: Cascading Style Sheets**

- Fogli stile: specificano *regole di visualizzazione* degli elementi di un documento XML
- Si può associare regola che definisce colore, stile, dimensione del contenuto tra tag aperto e chiuso
- A parità di contenuto, si possono fare visualizzazioni diverse dei dati
- Quando il browser utente carica un documento, applica le regole di visualizzazione al contenuto del documento

CSS – Esempio – documento XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/css" href="stile.css"?>
<studenti>
  <studente>
    <nome>Mario</nome> <cognome>Mario</cognome>
    <matricola>101111</matricola>
    <recapito>
      <via>via Risorgimento</via> <cap>33100</cap>
      <comune>Torino</comune> <provincia>TO</provincia>
    </recapito>
  </studente>
  <studente>
    <nome>Luisa</nome> <cognome>Bianchi</cognome>
    <matricola>202222</matricola>
    <recapito>
      <via>via Roma</via> <cap>10100</cap>
      <comune>Torino</comune> <provincia>TO</provincia>
    </recapito>
  </studente>
</studenti>
```

*Processing
instruction*

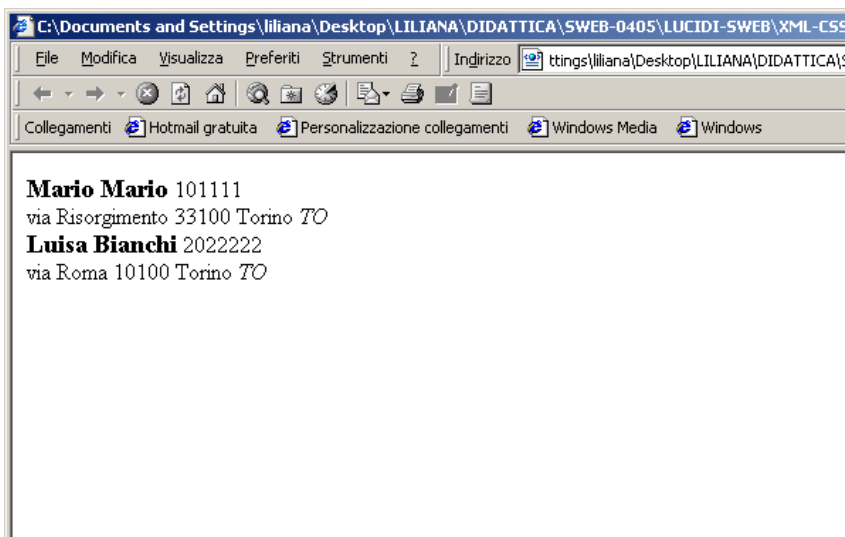
CSS – Esempio – foglio stile (stile.css)

```
studente
{
background-color: #ffffff;
}
nome
{
font-size: larger;
font-weight: bold;
}
cognome
{
font-size: larger;
font-weight: bold;
}
recapito
{
display: block;
}
provincia
{
font-style: italic;
}
```

*Regola di presentazione
del tag "studente"*

*Regola di presentazione
del tag "cognome"*

CSS – Esempio – visualizzazione



Fogli Stile – CSS – concetti - 1

Selettori di tipo

- **Selettore universale (*)** permette di identificare tutti gli elementi di un documento
 - * {font_weight: bold}
- **Selettore di elemento specifico:** individua gli elementi di un certo tipo, indicandone il nome
 - NOME {font_weight: bold}
- **Raggruppamento di selettori**
 - NOME, COGNOME {...}
 - NOME, COGNOME {font_weight: bold, font_size: larger}

Fogli Stile – CSS – concetti - 2

Modalità di presentazione (alcune...)

- **none:** elemento non viene visualizzato
- **block:** elemento viene visualizzato e completato con un a-capo
- **font-size:** specifica dimensione in modo assoluto (x-small, small, medium, large, ...) o relativo (smaller, larger)
- **font-weight:** per impostare bold
- **Font-style:** per impostare corsivo e altri stili