XML – Introduzione

XML stà per **eXtensible Markup Language** ed è stato realizzato per registrare e trasportare dati tra device e piattaforme differenti attraverso il protocollo http, che è quello che affronteremo nel presente tutorial.

Per rappresentare il metalinguaggio XML occorre dotarsi di un semplice editor di testi (es. note) ed avere le cognizioni basilari di HTML e DOM.

(fonte: Università di Bologna)

- XML è stato sviluppato dal World Wide Web Consortium
- Nel 1996 è stato formato un gruppo di lavoro con l'incarico di definire un linguaggio a markup estensibile di uso generale
- Le specifiche sono state rilasciate come W3C Recommendation nel 1998 e aggiornate nel 2004
- XML deriva da SGML, un linguaggio di mark-up dichiarativo sviluppato dalla International Standard Organization (ISO) e pubblicato ufficialmente nel 1986 con la sigla ISO 8879
- XML nasce come un sottoinsieme semplificato di SGML orientato all' utilizzo su World Wide Web
- Ha assunto ormai un ruolo autonomo e una diffusione ben maggiore del suo progenitore
- SGML (Standard Generalized Markup Language) è (Standard Generalized Markup Language) è il padre sia di HTML che di XML.
    - Pregi
        - Potente e flessibile (standard ISO, espandibile, fortemente strutturato, non proprietario, indipendente dalla piattaforma)
    - Difetti
        - Struttura pesante
        - Sono obbligatori un DTD e uno StyleSheet
        - Obbligatoria la validazione del documento
        - Le istanze SGML sono troppo pesanti e non robuste per applicazioni WEB
- Come SGML, XML è un linguaggio a marcatori (markup)
- Un linguaggio di markup è composto di istruzioni, definite tag o marcatori, che descrivono la struttura e la forma di un documento
- Ogni marcatore (o coppia di marcatori) identifica un elemento o componente del documento
- I marcatori vengono inseriti all'interno del documento
- Sia il testo, sia i marcatori sono memorizzati in formato ASCII
- XML usa la codifica dei caratteri UNICODE
- Un documento XML è "leggibile" da un utente umano senza la mediazione di software specifico

Esempio di file xml con estensione .xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="xml_css.xsl" ?>
<scuola>
 <classe>
   <utente>
        <nome>Pino </nome>
        <cognome>Rossi</cognome>
   </utente>
 </classe>
 <classe>
   <utente>
        <nome>Nino</nome>
        <cognome>Del Fabbro</cognome>
   </utente>
 </classe>
 <classe>
   <utente>
        <nome>Eolo</nome>
        <cognome>Vento</cognome>
   </utente>
 </classe>
 <classe>
   <utente>
        <nome>Gino</nome>
        <cognome>Bianchi</cognome>
   </utente>
 </classe>
</scuola>
```

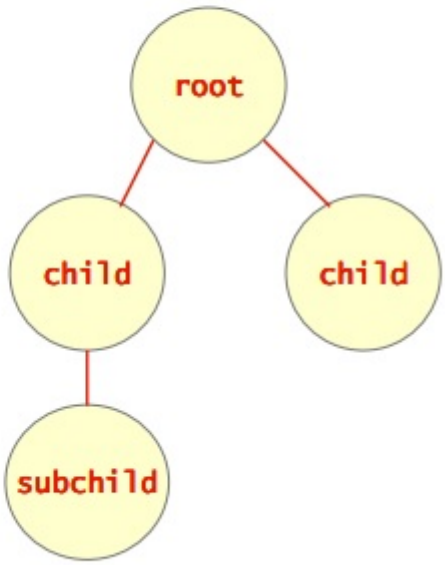
Attenzione: XML è case sensitive per cui i tag sono differenti se utilizzati come maiuscolo o minuscolo

- Gli elementi sono organizzati in un albero con una radice singola (root)
- Ogni documento XML può essere rappresentato come un albero che prende il come di document tree
- Struttura fisica di un documento XML
    - Un documento XML è un semplice file di testo
    - La struttura del documento viene rappresentata mediante marcatori (markup)
    - Gli elementi sono rappresentati mediante tag: coppie di marcatori che racchiudono il contenuto dell'elemento elemento
    - I sottoelementi sono tag rappresentati sotto forma di coppie nome- coppie nome-valore all valore all valore all'interno dei interno contenuti all'interno di un altro tag
    - Gli attributi vengono rappresentati sotto forma di coppie nome.valore all'interno dei tag
    - La radice è un tag che racchiude tutto il resto del documento (e quindi tutti gli altri tag
    - Un documento può inoltre contenere spazi bianchi, a capo e commenti

Struttura logica e fisica

- Esiste una corrispondenza diretta fra struttura fisica e struttura logica (tree)

```
<root>
  <child>
     <subchild>
          ...
     </subchild>
  </child>
  <child>
  ...
  </child>
</root>
```



Il documento XML è costituito da due parti:

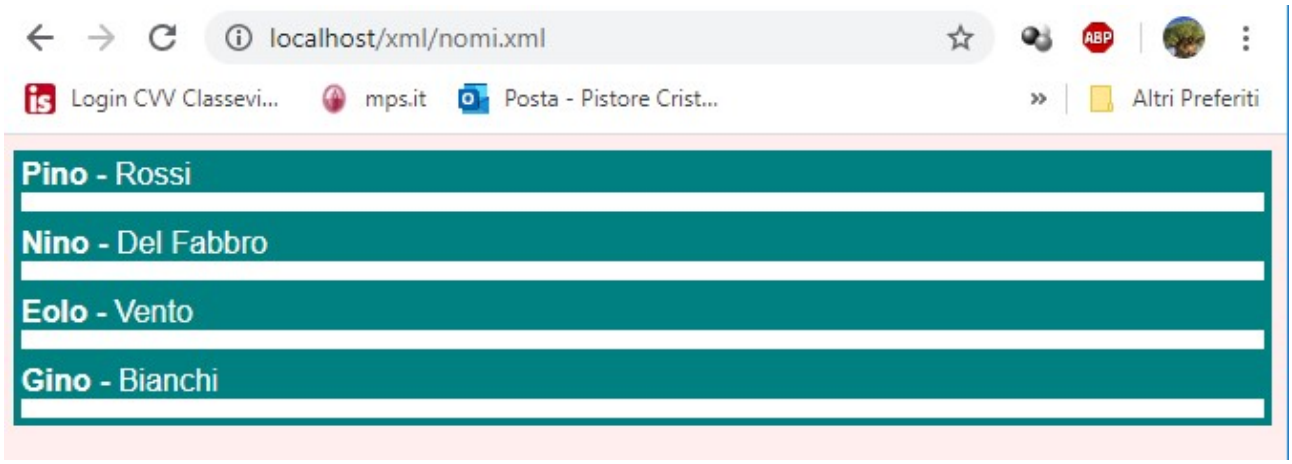
- **Prologo**: dichiarazione XML e riferimento (opzionale) ad altri documenti che ne definiscono la struttura
- **Corpo**: è il documento vero e proprio

Es di documento di definizione struttura riferita all'esempio precedente, con **estensione xsl**:

```
<?xml version="1.0" encoding="UTF-8"?>
<html xsl:version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<body style="font-family:Arial;font-size:12pt;background-color:#EEEEEE">
<xsl:for-each select="scuola/classe/utente">
  <div style="background-color:teal;color:white;padding:4px">
    <span style="font-weight:bold"><xsl:value-of select="nome" /> - </span>
    <xsl:value-of select="cognome" />
          <div style="height:10px;background-color:#ffffff;"></div>
  </div>

</xsl:for-each>
</body>
</html>
```

Risultato grafico:



I tag xml devono necessariamente essere chiusi, tranne nel caso il contenuto sia vuoto

Attributi:

- A ogni elemento possono essere associati uno o più attributi che ne specificano ulteriori caratteristiche o proprietà non strutturali.
- Ad esempio:
    - la lingua del suo contenuto testuale
    - un identificatore univoco
    - un numero di ordine
    - Gli attributi XML sono caratterizzati da un nome che li identifica e da un valore

  Es: Cognome=Capofamiglia

Oggetto **XMLHttpRequest**:

L'esempio riportato nella pagina seguente descrive l'utilizzo dell'oggetto **XMLHttpRequest** necessario per leggere i dati registrati nel file nomi.xml che abbiamo descritto più sopra.
Per l'utilizzo di questo oggetto occorre scomodare AJAX che è uno strumento di sviluppo di applicazioni web, che descriveremo in un altro tutorial.
Per descrivere lo script seguente introduciamo solo la funzione callback: é' necessario implementare la funzione callback per supervisionare lo stato della richiesta e accedere al risultato con le proprietà response solo se la richiesta ha avuto successo (vedi esempio quando si parla della proprietà onreadystatechange dell'oggetto XMLHttpRequest). In Javascript le funzioni consentono di passare come parametro sia dei valori di tipo variabile o costante, che delle funzioni. Le funzioni passate come parametro prendono il nome di funzioni di callback (funzioni di richiamo). Tali funzioni di callback vengono eseguite dopo la funzione/routine principale, quindi al termine dell'esecuzione primaria lanciata dalla funzione chiamante. Nel nostro caso, quando il server risponderà, con i suoi tempi, verrà eseguita la funzione di callback passata come argomento alla funzione asincrona e si inizierà a interpretare la stringa ottenuta con respone.

```
<p id="testo"></p>
	<script>
		var filehttp = new  XMLHttpRequest();
		filehttp.onreadystatechange = function() {
		  if (this.readyState == 4 && this.status == 200) {
		  myFunction(this);
		  }
		};

		filehttp.open("GET", "nomi.xml", true);
		filehttp.send();

		function myFunction(file) {
		var xmlDoc;
		xmlDoc = file.responseText;
		 document.getElementById("testo").innerHTML = xmlDoc;
		}
	</script>
```

Per scrivere un file in formato XML occorrerà riferirsi ad un linguaggio lato server (tipo php) che potrà trasferire i dati presenti in un ipotetico database protetto in un file di testo di formato standard xml.