

## **XML** (Exensible Markup Language)

- Linguaggio informatico che consente di descrivere ogni contenuto multimediale mediante testo
- Si presta a: creazione, memorizzazione, catalogazione e distribuzione di documenti digitali
- Ha struttura gerarchica definita dai tag o marcatori, contenenti testo o altri elementi (ai quali possono essere associati degli attributi)
- Sviluppato da World Wide Web Consortium nel 1998 , deriva da SGML (1986), dal quale nasce anche HTML

## VANTAGGI:

- Sintassi rigorosa condivisa universalmente
- Struttura che lo rende chiaro e leggibile dall'utente
- Indipendente da hardware e software su cui viene usato
- Indipendente dal documento di passaggio
- Trasmesso facilmente via internet
- Linguaggio estensibile, sistema aperto e interoperabile
- Standard di pubblico dominio, usato per validazione documenti

XML è un metalinguaggio che segue regole sintattiche e semantiche, con le quali si può descrivere un linguaggio markup. Un documento XML che segue le regole sintattiche è detto *well-formed*, uno che segue le regole sia sintattiche che semantiche è detto *valido*.

Un documento XML deve essere sempre ben formato e per la codifica usa caratteri di tipo Unicode.

- Il nome di un elemento XML è così indicato :

*<elemento>*

- Il contenuto di un elemento è racchiuso in una coppia di marcatori:

*<elemento>contenuto</elemento>*

dove / indica la chiusura di quell'elemento

- Tutti i tag aperti devono essere chiusi

- Un tag può essere accompagnato da uno o più attributi , indicati con :

*<elemento attributo="valore">contenuto</elemento attributo="valore">*

- Elemento vuoto :

*<elemento>empty</elemento>* o *<elemento/>*

- I tag non possono avere nomi che iniziano con XML, numeri o caratteri speciali e non possono contenere spazi.

Un documento XML si compone di due parti :un prologo e un corpo.

*PROLOGO:*

dichiarazione XML, che contiene informazioni sulla versione e il set di caratteri utilizzati, del tipo:

`<?xml version="1.0"?>` oppure

`<?xml version="1.0"? encoding="UTF-8"?>`

altri tipi di carattere possono essere ad esempio: ASCII (7 bit) UTF-16 (Unicode a 16 bit).

Un documento XML può presentare anche dei commenti in ogni sua parte (nel prologo come nel corpo), utili per spiegare la struttura del documento o commentarlo durante le fasi di sviluppo. I commenti sono visibili solo se si guarda il codice sorgente del documento XML. La loro struttura è:  
`<!-- Questo è un commento -->` .

XML è case sensitive, cioè distingue tra caratteri maiuscoli e minuscoli nei tag.

## **DTD** (Document type definition)

- fornisce regole per indicare e descrivere gli elementi di un documento XML, risultando quindi uno strumento per la loro validazione
- non è un linguaggio XML.
- Per applicare un DTD a un documento XML, nel prologo di quest'ultimo va inserita una dichiarazione di questo tipo:  
`<!DOCTYPE root-element SYSTEM "filename.dtd">` se si tratta di un file locale  
`<!DOCTYPE root-element PUBLIC "http://www.namesite.org/filename.dtd">`  
se il file è accessibile tramite url pubblico.

Un DTD è formato da un elenco di dichiarazioni (markup declarations) che descrivono la struttura del documento XML: elementi presenti, attributi con i rispettivi tipi e valori, il content model (modello di contenuto) degli elementi.

## **LIMITI:**

- non garantisce nessun supporto per i namespace
- non è possibile creare tipi di dati
- presenta scarsa estensibilità
- non è in formato XML
- non è possibile elaborare dati al di fuori del set di caratteri a disposizione

Un elemento in un DTD (che può essere dichiarato una sola volta) è così presentato:

`<!ELEMENT element-name content-model>`

Il content model può essere di quattro tipi:

- EMPTY (indica un elemento vuoto)
- ANY (indica che c'è qualcosa all'interno di un elemento)
- CHILDREN e MIXED: questi non sono indicati con una parola chiave ma dalla notazione usata.

Il content model Children presenta una sequenza di elementi che devono comparire nel documento in un ordine preciso, separati da una virgola ( $E1, E2, E3, \dots E_n$ ), oppure separati da una barra verticale: ( $E1 | E2 | E3 | \dots | E_n$ ).

Il content model Mixed consente di inserire un testo senza markup più un elenco di elementi figli senza ordine specifico, secondo la notazione:  $(\#PCDATA | E1 | E2 | E3 | \dots | E_n)^*$ . #PCDATA deve essere sempre il primo elemento della lista, la quale deve poter comparire zero o più volte (si usa infatti \*).

Si può anche stabilire quante volte un elemento può comparire attraverso gli operatori:

? (zero o una volta)    + (una o più volte)    \* (zero o più volte)

#PCDATA (Parsed Character Data) indica che il contenuto dell'elemento è solo testo, indicato:  
<!ELEMENT *element-name* (#PCDATA)>

Gli attributi riferibili ai vari elementi di un documento possono essere infiniti e vengono indicati, presentando un nome, un tipo e un valore:

<!ELEMENT *ElementName* *AttributeName* *AttributeType* *AttributeValue*>

Il tipo di un attributo può essere:

- *CDATA* testo  
(*en1* | *en2* | *en3* | ... | *enn*) scelto da una lista
- *ID* identificatore univoco all'interno di un documento

Il valore di un attributo può essere:

- *VALUE* valore di default (attribuito se non viene assegnato un valore)
- *#IMPLIED* valore opzionale (se il valore non viene assegnato allora è nullo)
- *#REQUIRED* l'attributo deve essere obbligatoriamente presente
- *#FIXED VALUE* l'attributo deve avere un valore fisso, assegnato anche se omissivo.

Abbiamo applicato il linguaggio XML in ambito artistico , prendendo in riferimento il catalogo opere del Museo MADRE di Napoli, in particolare dedicandoci ad una delle opere presenti in esso: 'Black Brother' di Anish Kapoor.



## XML

```
<?xmlversion="1.0" encoding="UTF-8">  
<!DOCTYPE catalogo SYSTEM "catalogo.dtd">  
<collezione>  
  <operadiproprietà="SI" esposta="SI" codice="5561" quadro="NO">  
    <titolo> Dark brother </titolo>  
    <autore> Anish Kapoor </autore>  
    <data> 2005 </data>  
    <dimensione> 5x2x2 m </dimensioni>  
    <colore> nero </colore>  
  </opera>  
</collezione>
```

## DTD

```
<!ELEMENT collezione (opera*)>
<!ELEMENT opera (titolo, autore, data, dimensione, colore)>
<!ELEMENT titolo (#PCDATA)>
<!ELEMENT autore (#PCDATA)>
<!ELEMENT data (#PCDATA)>
<!ELEMENT dimensione(#PCDATA)>
<!ELEMENT colore(#PCDATA)>
<!ATTLIST opera
    diproprietà (SI|NO) #REQUIRED
    esposta (SI|NO) #REQUIRED
    codice ID #IMPLIED
    quadro (SI|NO) #IMPLIED >
```