

Introduzione e struttura

- Linguaggio di **markup**
 - Formattazione = Forma logica delle pagine/documenti
- eXtensible Markup Language (acronimo/sigla)
- Possibilità di creare dei tag **personalizzati** seguendo una struttura ad albero (document tree)

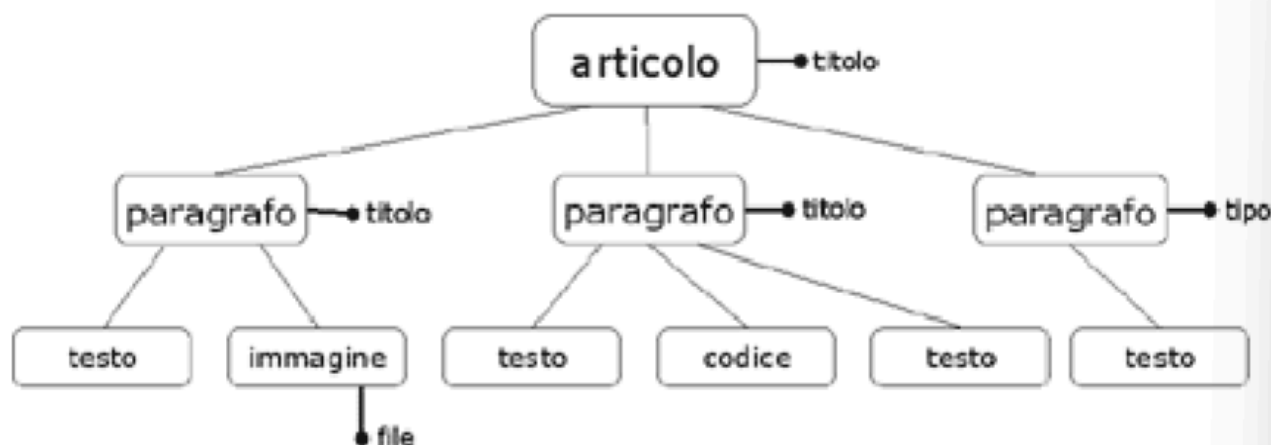
```
<!-- Questi sono tag FISSI (sempre quelli) in HTML -->
<html>

    <body>

    </body>

</html>
```

MA XML è diverso:



- Crei TU il fatto di avere ARTICOLI e PARAGRAFI!
 - E crei SEMPRE TU il fatto di avere testo/codice/immagine

```
<!-- CREI TU i tag in base al contesto e anche gli attributi! -->
<xml version="1.0" encoding="UTF-8">
<!-- VERSIONE e CODIFICA - UTF-8 (standard per tutti i caratteri ad 8 bit)
- per caratteri accentati, emoji -->
<articolo titolo="Sono il titolo">
    <paragrafo titolo="Titolo 1">
        <testo>ABCDEF</testo>
```

```

    <immagine file="file.jpg"></immagine>
</paragrafo>
<paragrafo titolo="Titolo 2">
    <testo>GHDHDH</testo>
    <codice>1234567</codice>
    <testo>ABCDEF</testo>
</paragrafo>
<paragrafo titolo="Titolo 2">
    <testo>URURURURU</testo>
</paragrafo>
</articolo>

```

Caratteristiche

Un documento è "ben formato" quando:

- È *case sensitive*! (distingue maiuscole/minuscole)

```

<PARAGRAFO> <Paragrafo> <paragrafo>
<!-- Sono tutti diversi (ed esistono tutti - volendo)! -->

```

- Ogni tag *deve essere APERTO e CHIUSO* e deve essere rispettato l'ordine! (come vedi dall'immagine sopra)

```

<paragrafo>
</paragrafo>

<!-- oppure -->

<br/> <!-- Chiusura da solo! -->

```

- Possiamo mettere dei commenti --> CDATA
 - Sia codice che testo!

```

<codice>
  `` `xml
  <![CDATA[
    testo da trattare letteralmente
  ]]>
</codice>

<![CDATA][
  HAHAAHIWEOJòFIWRòOIGOLEIQRGLUQE

```

- Possiamo leggere le *entità*! (caratteri speciali)

(C) -> ©

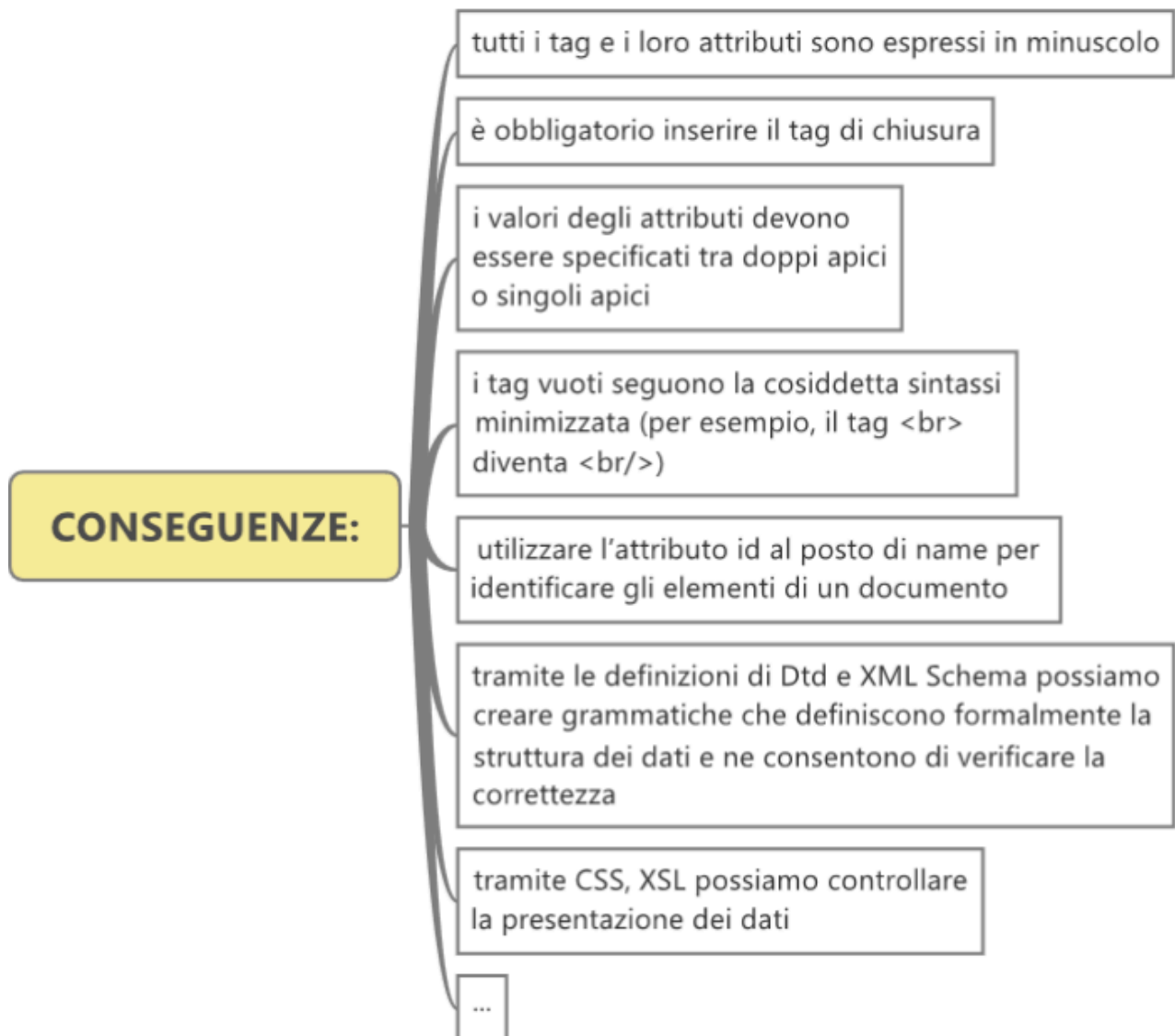
\& (and commerciale) -> &and

\> (segno maggiore) -> > (greater than - maggiore rispetto a "qualcosa")

\< (segno minore) -> < (lower than - minore rispetto a "qualcosa")

- Con XML gestiamo la *presentazione* dei dati!

- CSS = estetica
- XML = "estetica"
- HTML = formattazione
- JavaScript/PHP = logica



- Linguaggio "ben formato"?

- Ti serve per fare "parsing" (=elaborazione) di un insieme di dati!

Esempio classico:

```
File Modifica Formato Visualizza ?
<?xml version="1.0" encoding="UTF-8"?>
<utenti>
  <utente anni="20">
    <nome>Ema</nome>
    <cognome>Princi</cognome>
    <indirizzo>Torino</indirizzo>
  </utente>
  <utente anni="54">
    <nome>Max</nome>
    <cognome>Rossi</cognome>
    <indirizzo>Roma</indirizzo>
  </utente>
</utenti>
```



Ne leggi a migliaia/milioni!