xml

Un file xml

- Esempio di xml: strings.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Hello World, HelloWorld!</string>
    <string name="app_name">HelloWorld</string>
</resources>
```

Cosa è XML

- la sigla vuol dire EXtensible Markup Language
- XML è un (meta)linguaggio basato su marcatori (tag)
- XML è pensato per strutturare, organizzare e trasportare informazioni
  - non per visualizzarle
- I tag XML non sono predefiniti
  - li decidete voi
- XML è pensato per essere auto esplicativo
- XML è specificato dal W3C

Esempio

```
<bookstore>
  <book category="CHILDREN">
    <title>Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title>Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

Struttura di un documento xml

- Un documento xml rappresenta un albero

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

- Tutto ciò che è contenuto fra due tag, compresi i tag, si chiama elemento

Regole

- I tag devono essere chiusi
  - `<nome> Pierpaolo </nome>`
  - `<nome />`
- Xml è case sensitive
  - `<Messaggio> Questo è sbagliato</messaggio>`
  - `<messaggio> Questo è corretto </messaggio>`
- I tag vanno annidati in modo opportuno
  - `<b><i>Annidamento sbagliato</b></i>`
  - `<b><i>Annidamento corretto</i></b>`
- Ci deve essere un solo elemento di root

Altre regole

- Commenti
  - `<!-- Questo è un commento -->`
- Nomi per tag
  - possono contenere lettere numeri ed altri caratteri
  - Non possono iniziare con un numero
  - Non possono iniziare con le lettere xml
  - Non possono contenere spazi
  - Non ci sono parole riservate
- Ci sono cinque caratteri speciali in xml che vanno sostituiti

&lt;	<	minore
&gt;	>	maggiore
&amp;	&	ampersand
&apos;	'	apostrofo
&quot;	"	apici doppi

Attributi

- Forniscono informazioni che non sono parte dei dati
  - `<file type="gif">computer.gif</file>`

- Gli attributi vanno fra gli apici
  - `<messaggio id=21 > Questo è sbagliato</messaggio>`
  - `<messaggio id=”22”> Questo è corretto </messaggio>`
  - `<messaggio id=’22’> Questo è corretto </messaggio>`

Attributi ed Elementi

- ```
  <note date="10/01/2008">
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
  </note>
  ```

- ```
  <note>
    <date>10/01/2008</date>
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
  </note>
  ```

Attributi ed Elementi 2

```
<note>
  <date>
    <day>10</day>
    <month>01</month>
    <year>2008</year>
  </date>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

```
<note day="10" month="01" year="2008” to="Tove”
from="Jani" heading="Reminder”
body="Don't forget me this weekend!">
</note>
```

Documenti Validi

- Un documento è valido se rispetta le regole formali

- Posso specificare la struttura del documento?
  - Document Type Definition (DTD)
  - Xml Schema

- Un documento si dice corretto se rispetta le regole di un DTD o di un xml schema

Esempio di DTD

- ```
  <!DOCTYPE note
  [
  <!ELEMENT note (to,from,heading,body)>
  <!ELEMENT to (#PCDATA)>
  <!ELEMENT from (#PCDATA)>
  <!ELEMENT heading (#PCDATA)>
  <!ELEMENT body (#PCDATA)>
  ]>
  ```

- ```
  <?xml version="1.0" encoding="ISO-8859-1"?>
  <!DOCTYPE note SYSTEM "Note.dtd">
  <note>
      <to>Tove</to>
      <from>Jani</from>
      <heading>Reminder</heading>
      <body>Don't forget me this weekend!</body>
  </note>
  ```

XML Schema

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.w3schools.com"
xmlns="http://www.w3schools.com"
elementFormDefault="qualified">

<xs:element name="note">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="to" type="xs:string"/>
      <xs:element name="from" type="xs:string"/>
      <xs:element name="heading" type="xs:string"/>
      <xs:element name="body" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>
```

Namespace

- Sevono ad evtare il conflitto dei nomi

```
<root>
<h:table xmlns:h="http://www.w3.org/TR/html4/">
  <h:tr>
    <h:td>Apples</h:td>
    <h:td>Bananas</h:td>
  </h:tr>
</h:table>
<f:table xmlns:f="http://www.w3schools.com/furniture">
  <f:name>African Coffee Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>
</root>
```

- Si possono inserire nell'elemento in cui si usano o nel root