XSD SCHEMA per elementi complessi <xs:complexType>..... </xs:complexType>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema> ..... </xs:schema>

Se non avesse attributi sarebbe un elemento semplice

ELEMENTO CHE CONTIENE SIA TESTO SIA ALTRI ELEMENTI

XSD schema	XML Valido
<?xml version="1.0" ?> **<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">** **<xs:element name="messaggio">** **<xs:complexType mixed="true">** **<xs:sequence>** **<xs:element name="codice" type="xs:string"/>** **<xs:element name="data" type="xs:date"/>** **<xs:element name="ora" type="xs:time"/>** **<xs:element name="luogo" type="xs:string"/>** **</xs:sequence>** **</xs:complexType>** **</xs:element>** **</xs:schema>**	<?xml version="1.0" ?> **<messaggio>** La spedizione **<codice>**152333**</codice>** è stata inviata in data **<data>**2018-10-10**</data>** alle ore **<ora>**13:30:05**</ora>** al **<luogo>**Zurigo**</luogo>** **</messaggio>**

Bisogna specificare l'attributo mixed="true" nel complexType

Tipi di dati predefiniti

I TIPI DI DATO PREDEFINITI

Valdonati uno solo tutti voi per pagamento che si sono ora e si convertirà statto.

Il Schema Definizione XML contiene tutte le definizioni, sia built-in che quelle a disposizione. Il documento Schema è scritto in linguaggio **generico** ed permette così l'estensione **anche ad entità non legali** quali:

- Regole in genere
- Regole in genere con attributi, o che siano di tipo semplice
- Regole in genere con attributi, o che siano di tipo semplice ma non vuoti
- Dati numerici (long, int, ecc)
- Dati numerici (long, int, ecc) con attributi
- Dati numerici (long, int, ecc) con attributi ma non vuoti
- Dati numerici (long, int, ecc) con attributi ma non vuoti con valore di default
- Dati numerici (long, int, ecc) con attributi ma non vuoti con valore fisso

Carattere alfanumerico (stringhe)

A voce anche: con lleggendo/leggo ecc? Lunghezza arbitraria, Vuol quindi con set quale il 0 max valore che rappresentiamo, altrimenti, se lo sai, usa NO, che è a long comunque prendiamo di valore negativo per le di processori

xs:boolean	dato numerico rappresentato con 1 bit. Inserito/vero se uguale a zero/falso
xs:byte	dato numero rappresentato con 8 bit. Inserito/vero se uguale a zero/falso
xs:decimal	dato numero rappresentato con 32 bit. Inserito/vero se uguale a zero/falso
xs:int	dato numero rappresentato con 32 bit. Inserito/vero se uguale a zero/falso
xs:integer	dato numero intero
xs:long	dato numero intero lungo
xs:negativeInteger	dato numero intero negativo
xs:nonNegativeInteger	dato numero intero non negativo (maggiore o uguale a zero)
xs:nonPositiveInteger	dato numero intero non positivo (minore o uguale a zero)
xs:positiveInteger	dato numero intero positivo (maggiore di zero)
xs:short	dato numero intero corto
xs:unsigned Long	dato numero intero senza segno (maggiore di zero)
xs:unsigned Int	dato numero intero senza segno (maggiore di zero)
xs:unsigned Short	dato numero intero senza segno (maggiore di zero)
xs:unsigned Byte	dato numero intero senza segno (maggiore di zero)
xs:decimal	dato numero decimale
xs:float	dato numero decimale
xs:double	dato numero decimale
xs:boolean	Vero/falso
xs:date	Data in formato aaaa-mm-gg
xs:time	Ora in formato hh:mm:ss
xs:dateTime	Data e ora in formato aaaa-mm-gg hh:mm:ss
xs:duration	Durata in formato aaaa-mm-gg hh:mm:ss
xs:gYear	Anno in formato aaaa
xs:gYearMonth	Anno e mese in formato aaaa-mm
xs:gMonth	Mese in formato mm
xs:gMonthDay	Mese e giorno in formato mm-gg
xs:gDay	Giorno in formato gg
xs:Name	Stringa di caratteri che iniziano con lettera o underscore
xs:QName	Stringa di caratteri con prefisso (Namespace)
xs:NCName	Stringa di caratteri senza prefisso (Namespace)
xs:anyURI	Stringa di caratteri che rappresenta una URI
xs:base64Binary	Stringa di caratteri che rappresenta un dato binario in base 64
xs:hexBinary	Stringa di caratteri che rappresenta un dato binario in esadecimale
xs:language	Stringa di caratteri che rappresenta un codice di lingua secondo RFC 1766
xs:NMTOKEN	Stringa di caratteri che rappresenta un token SGML
xs:NMTOKENS	Stringa di caratteri che rappresenta una lista di token SGML
xs:Name	Stringa di caratteri che rappresenta un nome SGML
xs:NCNAME	Stringa di caratteri che rappresenta un nome senza prefisso SGML
xs:ID	Stringa di caratteri che rappresenta un identificatore univoco SGML
xs:IDREF	Stringa di caratteri che rappresenta un riferimento a un identificatore SGML
xs:IDREFS	Stringa di caratteri che rappresenta una lista di riferimenti a identificatori SGML
xs:ENTITY	Stringa di caratteri che rappresenta un nome di entità SGML
xs:ENTITIES	Stringa di caratteri che rappresenta una lista di nomi di entità SGML
xs:NOTATION	Stringa di caratteri che rappresenta un nome di notazione SGML

URI

URI (uniform resource identifier) Indica anche come URL (uniform resource locator) ed indica una pagina web

URN (uniform resource name) Indica una risorsa come qualsiasi libro, persona, ecc

URN (uniform resource identifier) Indica una risorsa come qualsiasi libro, persona ecc (ISBN, ISSN, ecc)

<base>2013-01-01T00:**</base>**
<dur>P1Y2M3DT10H30M**</dur>**
(rappresenta: sudo 1 anno 2 mesi 3 giorni 10 ore 30 min)

<base>2013-01-01T00:**</base>**
<dur>P365D**</dur>**
(specificare il tipo secondo di durata, in questo caso 365 giorni)

ELEMENTO CHE CONTIENE ALTRI ELEMENTI

Tutti questi indicatori conentono UNA SOLA OCCORRENZA degli elementi indicati nella sequenza. Per consentire più occorrenze di un elemento all'interno di un elemento complesso, si devono utilizzare, nell'elemento contenuto gli attributi minOccur e maxOccur. Specificando maxOccur="unbounded" si specifica l'assenza di un numero massimo al numero di occorrenze

minOccur="0"

maxOccur="unbounded"

<?xml version="1.0" ?> **<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">** **<xs:element name="persona">** **<xs:complexType>** **<xs:sequence>** **<xs:element name="cognome" type="xs:string"/>** **<xs:element name="name" type="xs:string"/>** **<xs:element name="figlio" type="xs:string"** **minOccur="0" maxOccur="unbounded"/>** **</xs:sequence>** **</xs:complexType>** **</xs:element>** **</xs:schema>**	XML Valido: **<?xml version="1.0" ?>** **<persona>** **<cognome>**Lilla**</cognome>** **<nome>**Fernando**</nome>** **<figlio>**Rino**</figlio>** **<figlio>**Peppe**</figlio>** **</persona>** XML Valido: **<?xml version="1.0" ?>** **<persona>** **<cognome>**Lilla**</cognome>** **<nome>**Fernando**</nome>** **</persona>**

<?xml version="1.0" ?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:complexType name="persona">

<xs:sequence>

<xs:element name="cognome" type="xs:string"/>

<xs:element name="nome" type="xs:string"/>

<xs:element name="dataNascita" type="xs:date"/>

<xs:element name="luogoNascita" type="xs:string"/>

</xs:sequence>

</xs:complexType>

<xs:element name="studente" type="persona"/>

<xs:element name="docente" type="persona"/>

</xs:schema>

<xs:sequence> ..... </xs:sequence>	Nel documento XML l'elemento complesso DEVE contenere TUTTI gli elementi indicati nella sequenza NELLO STESSO ORDINE
<xs:choice> ..... </xs:choice>	Nel documento XML l'elemento complesso DEVE contere UNO SOLO fra gli elementi indicati nella sequenza
<xs:all> ..... </xs:all>	Nel documento XML l'elemento complesso DEVE contenere TUTTI gli elementi indicati nella sequenza ANCHE IN ORDINE DIVERSO

Se la sequenza viene indicata nel tag <xs:complexType name="...">..... </ xs:complexType> /> anziché nella definizione dell'elemento, ci definisce un nuovo tipo di dato complesso

ELEMENTO VUOTO (elemento che DEVE essere vuoto)

<?xml version ="1.0" ?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="elementoVuoto">

<xs:complexType>

</xs:complexType>

</xs:element>

</xs:schema>

Senza attributi

<?xml version ="1.0" ?> **<xs:schema xmlns:xs="**http://www.w3.org/2001/XMLSchema">** **<xs:element name="**elementoVuoto">** **<xs:complexType>** **<xs:attribute name="attributo" type="**xs:string"/>** **</xs:complexType>** **</xs:element>** **</xs:schema>**	commento di validare **<elementoVuoto attributo="**attributo di prova"/>**(elementoVuoto)

Con attributi

ELEMENTO CHE CONTIENE SOLO TESTO (di qualsiasi tipo, integer ecc.) MA CON ATTRIBUTI.

E' un elemento COMPLESSO con contenuto SEMPLICE (solo testo). IL CONTENUTO SEMPLICE VA SPECIFICATO CON IL TAG <xs:simpleContent> mentre l'aggiunta dell'attributo con il TAG <xs:extension base="....">

XSD schema	XML
<?xml version="1.0" ?> **<xs:schema xmlns:xs="**http://www.w3.org/2001/XMLSchema">** **<xs:element name="misura">** **<xs:complexType>** **<xs:simpleContent>** **<xs:extension base="**xs:decimal">** **<xs:attribute name="unita" type="**xs:string" **use="**required"/>** **</xs:extension>** **</xs:simpleContent>** **</xs:complexType>** **</xs:element>** **</xs:schema>**	Valido **<misura unita="**cm"**>**0.01**</misura>** Valido **<misura unita="**cm"**>**100**</misura>** Non valido **<misura>**100**</misura>**

Manca l'attributo unita

Qui specifico il type del contenuto semplice

Questo è il type dell'attributo

xs:extension base estende un tipo [type] o un elemento [xs:simple] che semplici che complessi] aggiungendo *uno o più attributi o elementi. Il concetto di estensione è analogo a quello dell'ereditarietà [infatti* ***l'unmarshalling [andrana] di un elemento con xs:extendi genera una classe [figlia].***

COME AGGIUNGERE UN ATTRIBUTO A UN ELEMENTO COMPLESSO (NON C'E' SPIEGATO BENE SUL LIBRO)

Si prevede che un attributo si definisce sempre come elemento semplice

<xs:attribute name="attributo" type="xs:string">

anche ad esso, come per gli elementi semplici, è possibile aggiungere delle restrizioni.

Per aggiungere un attributo ad un elemento complesso (così come, si è visto, per aggiungerlo all'elemento semplice) si opera "estensione".

La strategia è la seguente. Sopponiamo di voler aggiungere all'elemento complesso libro (che contiene gli elementi autore, titolo, prezzo) l'attributo "genere":

1. Nello schema XSD si definisce un tipo di elemento complesso "tipoLibro" contenente i vari elementi (autore, titolo, prezzo)
2. Si definisce, nello stesso schema XSD, l'elemento complesso "libro" di tipo "tipoLibro" estendendolo con l'aggiunta dell'attributo "genere"

schema XSD	XML
<?xml version="1.0" ?> **<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">** <!--definisco tipo di dati complesso "tipoLibro" -->** **<xs:complexType name="tipoLibro">** **<xs:sequence>** **<xs:element name="autore" type="xs:string"/>** **<xs:element name="titolo" type="xs:string"/>** **<xs:element name="prezzo" type="xs:float"/>** **</xs:sequence>** **</xs:complexType>** <!--definisco l'elemento "libro" di tipo "tipoLibro" e estendisco tale tipo con l'attributo "genere"-->** **<xs:element name="libro">** **<xs:complexType>** **<xs:simpleContent>** **<xs:extension base="tipoLibro">** **<xs:attribute name="genere" type="xs:string" use="required"/>** **</xs:extension>** **</xs:simpleContent>** **</xs:complexType>** **</xs:element>** **</xs:schema>**	**<?xml version="1.0"** ?> **<xs:schema xmlns:xs="**http://www.w3.org/2001/XMLSchema">** **<xs:element name="**libro">** **<xs:complexType>** **<xs:simpleContent>** **<xs:extension base="**tipoLibro">** **<xs:attribute name="genere" type="xs:string" use="required"/>** **</xs:extension>** **</xs:simpleContent>** **</xs:complexType>** **</xs:element>** **</xs:schema>**

Argomento dinamico