XML and Semantic Web Technologies

II. XML / 1. Unicode, URIs, and XML Syntax

Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Institute of Economics and Information Systems
& Institute of Computer Science
University of Hildesheim
http://www.ismll.uni-hildesheim.de

XML and Semantic Web Technologies

II. XML / 1. Unicode, URIs, and XML Syntax

Lars Schmidt-Thieme, Information Systems and Machine Learning Lab (ISMLL), University of Hildesheim, Germany,
Course on XML and Semantic Web Technologies, summer term 2009 1/43

Semantic Web Layer Cake

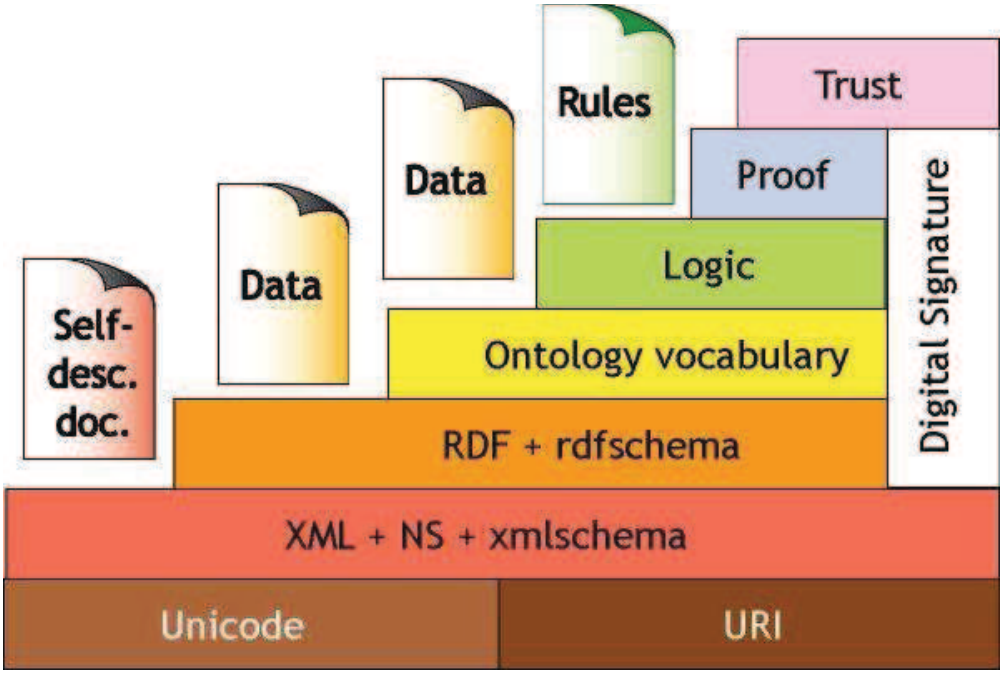


Figure 1: Semantic Web Layers (Berners-Lee).

Coded Character Sets

name	codes	examples
ASCII code	0–127	$64 \mapsto A$
ISO-8859-1, ISO-LATIN-1	0–255	0–127 as ASCII, $196 \mapsto$
ISO-8859-7	0–255	0–127 as ASCII, $225 \mapsto \alpha$
Unicode	0–$(2^{32}-1)$	0–255 as ISO-8859-1

Unicode is organized in 256 groups à 256 planes à 256 rows à 256 cells.

Plane 0 (codes 0–65535) is called **basis multilingual plane (BMP)**.

Non ISO-8859-1 characters are mapped to higher codes, e.g., $945 \mapsto \alpha$.

Unicode

Assigned characters of the Unicode standard (v5.1.0, 12/2008) can be found at http://www.unicode.org/charts/.

Unicode also specifies character classes for each character, as

- letters (capital and small),
- digits,
- punctuation,
- control characters.

Unicode / Scripts

The Unicode Character Code Charts

SCRIPTS | SYMBOLS AND PUNCTUATION | INDEX | CONVENTIONS AND RELATED LINKS

European Alphabets	African Scripts	Indic Scripts	East Asian Scripts	Central Asian Scripts
(see also **Comb. Marks**)	***Ethiopic***	Bengali	***Han Ideographs***	Kharoshthi
Armenian	Ethiopic	Devanagari	Unified CJK Ideographs (5MB)	Mongolian
Armenian	Ethiopic Supplement	Gujarati	CJK Ideographs Ext. A (2MB)	Phags-Pa (5.0)
Armenian Ligatures	Ethiopic Extended	Gurmukhi	CJK Ideographs Ext. B (13MB)	Tibetan
Coptic	***Other***	Kannada	Compatibility Ideographs (.5MB)	
Coptic	N'Ko (5.0)	Limbu	Compatibility Ideo. Suppl. (.5MB)	
Coptic in Greek block	Tifinagh	Malayalam	Kanbun	
Cyrillic	**Middle Eastern Scripts**	Oriya	(see also **Unihan Database**)	**Ancient Scripts**
Cyrillic	***Arabic***	Sinhala	***Radicals and Strokes***	***Ancient Greek***
Cyrillic Supplement	Arabic	Syloti Nagri	CJK Radicals	Ancient Greek Numbers
Georgian	Arabic Supplement	Tamil	KangXi Radicals	Ancient Greek Musical
Georgian	Arabic Presentation Forms A	Telugu	CJK Strokes	***Cuneiform***
Georgian Supplement	Arabic Presentation Forms B		Ideographic Description	Cuneiform (5.0)
Greek	***Hebrew***	**Philippine Scripts**	***Chinese-specific***	Cuneiform Numbers (5.0)
Greek	Hebrew	Buhid	Bopomofo	Old Persian
Greek Extended	*Hebrew Presentation Forms*	Hanunoo	***Japanese-specific***	Ugaritic
(see also ***Ancient Greek***)	***Other ME Scripts***	Tagalog	Hiragana	***Linear B***
Latin	Syriac	Tagbanwa	Katakana	Linear B Syllabary
Basic Latin	Thaana		Katakana Phonetic Ext.	Linear B Ideograms
Latin-1	**American scripts**	**South East Asian**	*Halfwidth Katakana*	***Other Ancient Scripts***
Latin Extended A	Canadian Syllabics	Buginese	***Korean-specific***	Aegean Numbers
Latin Extended B	Cherokee	Balinese (5.0)	Hangul Syllables (7MB)	Counting Rod Num. (5.0)
Latin Extended C (5.0)	Deseret	Khmer	Hangul Jamo	Cypriot Syllabary
Latin Extended Additional		Lao	Hangul Compatibility Jamo	Gothic
Latin Ligatures	**Other Scripts**	Myanmar	*Halfwidth Jamo*	Old Italic
Fullwidth Latin Letters	Shavian	New Tai Lue	***Yi***	Ogham
Small Forms	Osmanya	Tai Le	Yi (.6MB)	Runic
(see also **Phonetic Symbols**)	Glagolitic	Thai	Yi Radicals	Phoenician (5.0)

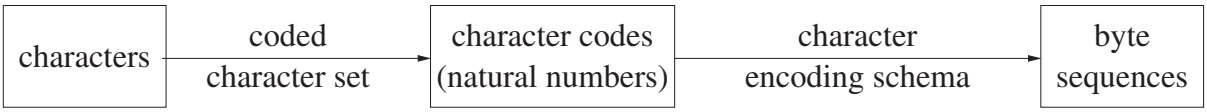
Unicode / Symbols

The Unicode Character Code Charts

SCRIPTS | SYMBOLS AND PUNCTUATION | INDEX | CONVENTIONS AND RELATED LINKS

Punctuation	Mathematical Symbols	Symbols	Private Use
General Punctuation	***Numbers and Digits***	***Miscellaneous Symbols***	Private Use Area
ASCII Punctuation	(see also specific scripts)	Dingbats	Suppl. Private Use Area A
Latin-1 Punctuation	*ASCII Digits*	Miscellaneous Symbols	Suppl. Private Use Area B
General Punctuation	*Fullwidth ASCII Digits*	Tai Xuan Jing Symbols	**Surrogates**
Supplemental Punctuation	Number Forms	Yijing Hexagrams	High Surrogates
CJK Punctuation	Super and Subscripts	Braille Patterns	High Private Use Surrogates
CJK Punctuation	***Letterlike Symbols***	***Musical Notation***	Low Surrogates
Fullwidth ASCII Punctuation	Letterlike Symbols	Ancient Greek Musical...	**Noncharacters in Charts**
Vertical Forms	Math Alphanumeric Symbols	Byzantine Musical Symbols	*Reserved range*
Enclosed and Square	***Arrows and Operators***	Western Musical Symbols	*At End of BMP*
Enclosed Alphanumerics	Arrows	***Currency Symbols***	*At End of Plane 1*
.... CJK Letters and Months	Mathematical Operators	(see also specific scripts)	*At End of Plane 2*
CJK Compatibility	Suppl. Math Operators	*Dollar Sign*	*At End of Plane 3*
(see also **Letterlike Symbols**)	Misc. Math Symbols A	*Yen, Pound and Cent*	*At End of Plane 4*
Combining Diacritical Marks	Misc. Math Symbols B	Currency Symbols	*At End of Plane 5*
Combining Diacritical Marks	Supplemental Arrows A	*Fullwidth Currency Symbols*	*At End of Plane 6*
.... for Symbols	Supplemental Arrows B	*Mark and Pfennig (historic)*	*At End of Plane 7*
.... Supplement	Misc. Symbols and Arrows	*Rial Sign*	*At End of Plane 8*
Combining Half Marks	***Geometrical Symbols***	**Specials**	*At End of Plane 9*
Phonetic Symbols	Geometrical Shapes	*Controls: C0, C1*	*At End of Plane 10*
IPA Extensions	Box Drawing	*Layout Controls*	*At End of Plane 11*
Phonetic Extensions	Block Elements	*Invisible Operators*	*At End of Plane 12*
Phonetic Extensions Supplement	***Technical Symbols***	Specials	*At End of Plane 13*
Modifier Tone Letters	Control Pictures	Tags	*At End of Plane 14*
Spacing Modifier Letters	Miscellaneous Technical	Variation Selectors	*At End of Plane 15*
(see also **Super and Subscript**)	OCR	Variation Selectors Supplement	*At End of Plane 16*

Character Encoding Schemata



Character Encoding Schemata are trivial for 1-byte coded character sets.

Direct representations of Unicode:

UCS-2: direct representation of codes 0–65535 with 2 bytes.

UCS-4: direct representation of all codes with 4 bytes.

Drawbacks of direct representations:

- bytecode `0x00` occurs (that marks string endings in C), e.g., in UCS-4:

$$A \mapsto 65 \mapsto (0, 0, 0, 65)$$

- uniform blow-up of storage space, but most texts mostly use ASCII or ISO-8859-1.

- error-prone, as if one byte is lost, all following data will be decoded incorrectly.

Unicode Transformation Formats (UTF)

Unicode Transformation Formats (UTF) use a variable number of bytes for coding a character.
UTF-8:

`0x00–0x7f` (bit sequences 0.....) code ASCII characters directly,

`0xc0–0xfd` (bit sequences 11.....) mark the start of a multi-byte character representation (and code its length and leading bits of its code),

`0x80–0xbf` (bit sequences 10.....) code continuations of multi-byte character representations,

`0xfe, 0xff` (bit sequences 1111111.) are not used.

bit sequence	*bytes*	*free bits*	*character codes*
0.....	1	7	0x00–
110..... 10.....	2	$5+6=11$	0x80–
1110.... 10..... 10.....	3	$4+2\cdot 6=16$	0x800–
11110... 10..... 10..... 10.....	4	$3+3\cdot 6=21$	0x10000–
111110.. 10..... 10..... 10..... 10.....	5	$2+4\cdot 6=26$	0x200000– 0x
1111110. 10..... 10..... 10..... 10..... 10.....	6	$1+5\cdot 6=31$	0x4000000–0x

II. XML / 1. Unicode, URIs, and XML Syntax

Uniform Resource Identifiers (URIs)

URIs are used to identify resources.

Example:

http://www.ismll.uni-hildesheim.de/lehre/xml-09s/index.html

URIs are defined in RFC 3986 (01/2005).

Generic URI syntax

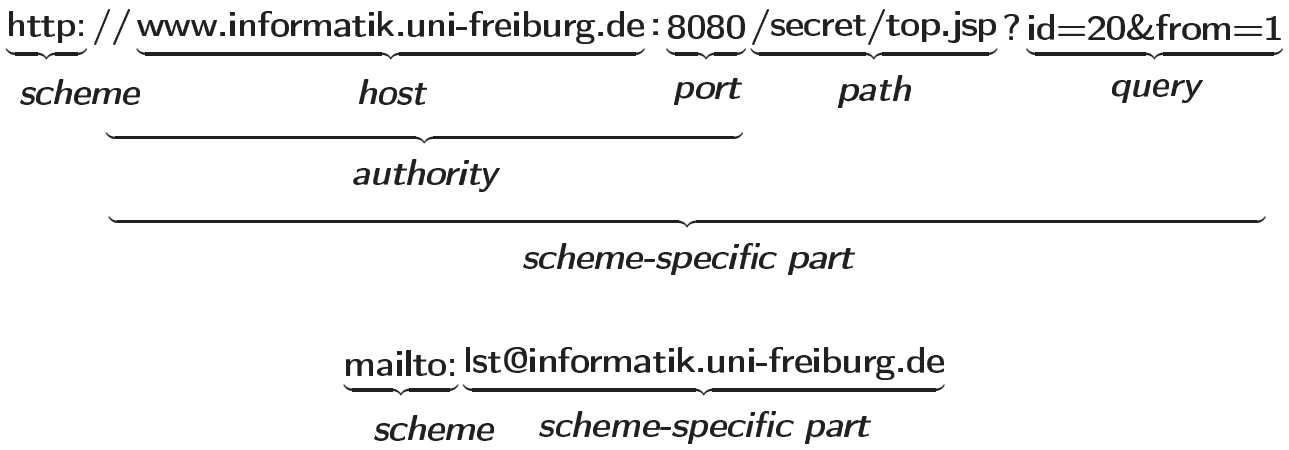


Figure 6: Typical parts of URIs.

Generic URI syntax:

⟨*URI*⟩ := ⟨*scheme*⟩ : ⟨*scheme-specific-part*⟩

Hierarchical URIs

An URI is called **hierarchical** iff

⟨*scheme-specific-part*⟩ := (// ⟨*authority*⟩ [⟨*path*⟩]
| ⟨*path*⟩) [? ⟨*query*⟩] [# ⟨*fragment*⟩]

⟨*path*⟩ := (/ ⟨*path-segment*⟩)+

otherwise its called **opaque**.

The path-segments `.` and `..` have special meaning: context path and parent path.

A hierarchical URI is called **server-based** iff

⟨*authority*⟩ := [⟨*userinfo*⟩ @] ⟨*host*⟩ [: ⟨*port*⟩]

otherwise it is called **registry-based**.

Fragment identifiers

Fragment identifiers are used to identify **parts of the resource** identified by an URI.

Example:

http://www.informatik.uni-freiburg.de/xml/books.html#R03

```
1 <html>
2 <body>
3   <li><a name="EE04">Rainer Eckstein, Silke Eckstein:
4     <em>XML und Datenmodellierung</em>, 2004.</a></li>
5   <li><a name="R03">Erik T. Ray:
6     <em>Learning XML</em>, 2003.</a></li>
7 </body>
8 </html>
```

Figure 7: HTML document at http://www.informatik.uni-freiburg.de/xml/books.html.

Relative (hierarchical) URIs

A relative URI is defined as:

⟨*relativeURI*⟩ ::= (// ⟨*authority*⟩ [⟨*path*⟩]
| ⟨*path*⟩
| ⟨*relativePath*⟩) [? ⟨*query*⟩]

⟨*relativePath*⟩ := ⟨*path-segment*⟩ (/ ⟨*path-segment*⟩)*

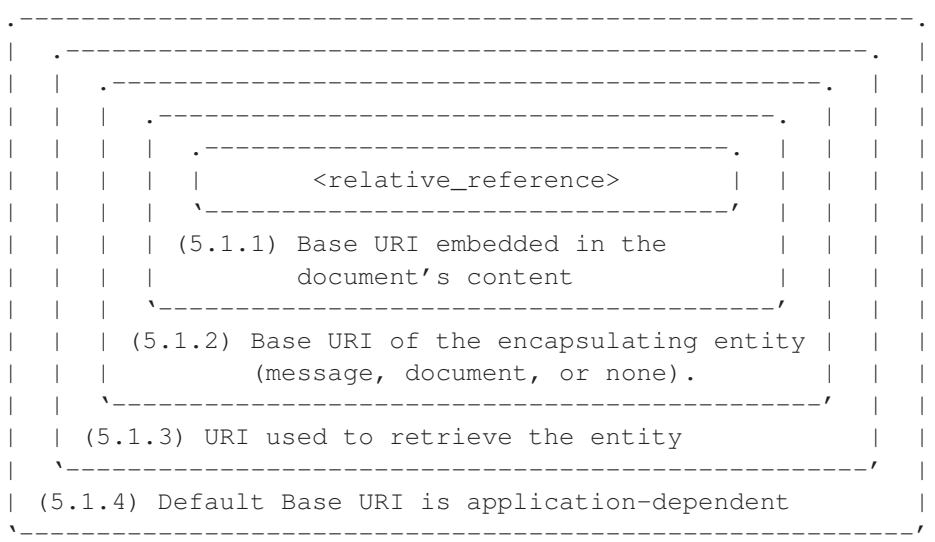


Figure 8: A **Base URI** is the context for resolving relative URIs [RFC 2396].

URI schemes

URI schemes are managed by Internet Assigned Numbers Authority (IANA).

Scheme Name	Description	Reference	Type
ftp	File Transfer Protocol	RFC 1738	server-based
http	Hypertext Transfer Protocol	RFC 2616	server-based
mailto	Electronic mail address	RFC 2368	opaque
file	Host-specific file names	RFC 1738	server-based
pop	Post Office Protocol v3	RFC 2384	server-based
dav	dav	RFC 2518	server-based
tel	telephone	RFC 2806	opaque
https	Hypertext Transfer Protocol Secure	RFC 2818	server-based
urn	Uniform Resource Names	RFC 2141	opaque
⋮	⋮	⋮	⋮

66 URI schemes (as of 2009-04-06; http://www.iana.org/assignments/uri-schemes.h
URI registrations are regulated by RFC 4396 (2/2006).

Example:

tel:+(49)-761-203-8164

URI types by URI semantics

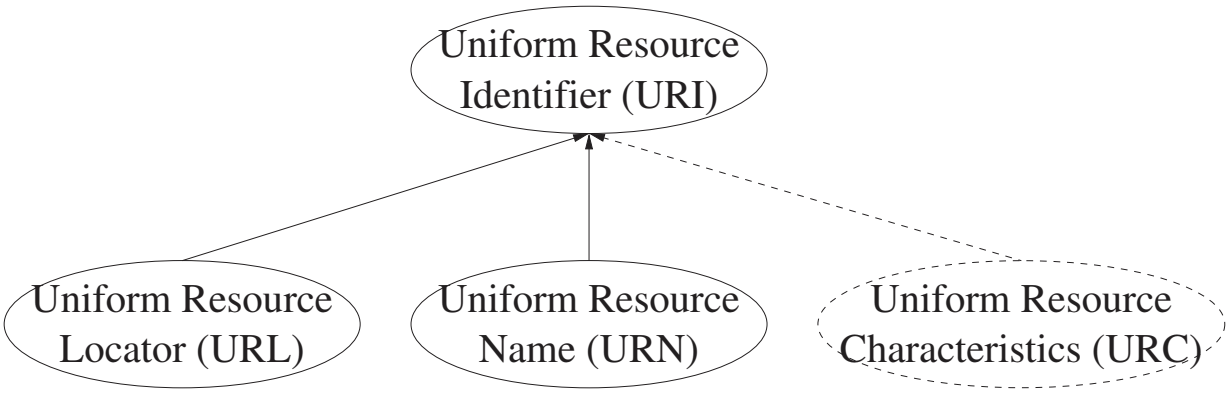


Figure 9: URI types.

Uniform Resource Names (URNs)

URNs are special kinds of URIs that

- **map other namespaces** into URN-space,
- are required to remain **globally unique and persistent** (even when the resource ceases to exist or becomes unavailable).
- have scheme `urn`.

⟨*URN*⟩ := `urn:` ⟨*namespace*⟩ `:` ⟨*namespace-specific-part*⟩

Examples:

urn:isbn:0-395-36341-1
urn:newsml:reuters.com:20000206:IIMFFH05643_2000-02-06_17-54-01_L0615

A book or a news item (identified by an URN) may be retrieved from different locations (URLs).

URN Namespaces	Value	Reference
ietf	1	RFC 2648
pin	2	RFC 3043
issn	3	RFC 3044
oid	4	RFC 3061
newsml	5	RFC 3085
oasis	6	RFC 3121
xmlorg	7	RFC 3120
publicid	8	RFC 3151
isbn	9	RFC 3187
nbn	10	RFC 3188
web3d	11	RFC 3541
mpeg	12	RFC 3614
mace	13	RFC 3613
fipa	14	RFC 3616
swift	15	RFC 3615
⋮	⋮	⋮

40 URN namespaces (as of 2008-12-09;
http://www.iana.org/assignments/urn-namespaces)

Characters Allowed in URIs

In URIs only some characters may be used literally in non-syntactic parts ("data").

All others have to be escaped using their code (in some character encoding):

⟨*dataChars*⟩ := ⟨*alphanum*⟩ | − | _ | . | ! | ~ | * | ′ | (|)
⟨*escapedChar*⟩ := % ⟨*hexDigit*⟩ ⟨*hexDigit*⟩

Codes have been interpreted as codes in different character encodings, depending on the URI scheme.

UTF-8 is recommended by RFC 2718 and already used by some schemes (e.g., urn, imap, pop).

Example:

http://www.informatik.uni-freiburg.de/login.jsp?name=Hans%20Meyer

Internationalized Resource Identifiers (IRIs)

IRIs allow more characters to be used literally (RFC 3987; 01/2005).

In IRIs only

- data characters that can be misinterpreted as syntactic characters and
- some bidirectional formatting characters

have to be escaped.

All other data characters are used literally
(in some character encoding, e.g., UTF-8).

Example:

http://www.informatik.uni-freiburg.de/login.jsp?name=Hans%20Müller

Schemes still are restricted to US ASCII characters.

II. XML / 1. Unicode, URIs, and XML Syntax

1. Unicode

2. Uniform Resource Identifiers (URIs)

3. XML Syntax

W3C development process

W3C specifications are called **Recommendations**.

Stages of W3C recommendations:

stage	completion date XML 1.0	XML 1.1
Working Draft	1996/11/14	2001/12/13
	1997/11/17	
Last Call Working Draft		2002/04/25
Candidate Recommendation		2002/10/15
Proposed Recommendation	1997/12/08	2003/11/05
Recommendation	1998/02/10	2004/04/15
Working Draft	2000/08/14	
Recommendation (2nd edition)	2000/10/06	2006/08/16
Proposed Edited Recommendation	2003/10/30	
Recommendation (3rd edition)	2004/02/04	
Recommendation (4th edition)	2006/08/16	
Recommendation (5th edition)	2008/11/26	

Every XML document consists of a **prolog** and a single element, called **root element**.

⟨*document*⟩ := ⟨*prolog*⟩ ⟨*element*⟩ (⟨*Comment*⟩ | ⟨*PI*⟩ | ⟨*S*⟩)*

⟨*prolog*⟩ := `<?xml` ⟨*S*⟩ `version = "1.1"`
(⟨*S*⟩ `encoding = "` ⟨*encoding*⟩ `"`)?
(⟨*S*⟩ `standalone =` (`"yes"` | `"no"`))?
⟨*S*⟩* `?>`
(⟨*Comment*⟩ | ⟨*PI*⟩ | ⟨*S*⟩)*
(⟨*DoctypeDecl*⟩ (⟨*Comment*⟩ | ⟨*PI*⟩ | ⟨*S*⟩)*)?

In all productions

- matching `"` can be replaced by `'`.
- `=` may be surrounded by spaces (i.e., match ⟨*S*⟩? `=` ⟨*S*⟩?).

⟨*S*⟩ := (#x20 | #x9 | #xD | #xA)+

A minimal XML document

```
1 <?xml version="1.1"?>
2 <page/>
```

Figure 10: A minimal XML document with root element "page".

In XML 1.1 the version attribute is mandatory.

If the version attribute is missing, version 1.0 is assumed.

Elements and Attributes

⟨*element*⟩ := ⟨*emptyElementTag*⟩
| ⟨*STag*⟩ ⟨*content*⟩ ⟨*ETag*⟩

⟨*emptyElementTag*⟩ := < ⟨*Name*⟩ (⟨*S*⟩ ⟨*Name*⟩ = " ⟨*AttValue*⟩ ")* ⟨*S*⟩? />

⟨*STag*⟩ := < ⟨*Name*⟩ (⟨*S*⟩ ⟨*Name*⟩ = " ⟨*AttValue*⟩ ")* ⟨*S*⟩? >

⟨*ETag*⟩ := </ ⟨*Name*⟩ ⟨*S*⟩? >

⟨*Name*⟩s

- start with a unicode letter or _
  (: is also allowed, but used for namespaces).
- may contain unicode letters, unicode digits, −, ., or ·.

A **wellformed document** requires,

- that start and end tag of each element match,
- that for each tag the same attribute never occurs twice.

Not-wellformed Documents (1/2)

```
1 <?xml version="1.1"?>
2 <book>
3   <author><fn>Rainer</fn><sn>Eckstein</sn></author>
4   <author><fn>Silke</fn><sn>Eckstein</sn></author>
5   <title>XML und Datenmodellierung</title>
6   <year>2004</year>
7 </book>
8 <book>
9   <author><fn>Erik T.</fn><sn>Ray</sn></author>
10  <title>Learning XML</title>
11  <year edition="2">2003</year>
12 </book>
```

Figure 11:

Not-wellformed Documents (2/2)

```
1 <?xml version="1.1"?>
2 <book>
3   <author><fn>Erik T.</fn><sn>Ray</author></sn>
4   <title>Learning XML</title>
5   <year edition="2">2003</year>
6 </book>
```

Figure 12:

```
1 <?xml version="1.1"?>
2 <book author="Rainer Eckstein" author="Silke Eckstein">
3   <title>XML und Datenmodellierung</title>
4   <year>2004</year>
5 </book>
```

Figure 13:

Element content

The contents of an element can be made up from 6 different things:

1. other elements,
2. Character data,
3. References,
4. CDATA sections,
5. Processing instructions, and
6. comments.

$$\langle content \rangle := \langle CharData \rangle? \; (\; (\langle element \rangle \mid \langle Reference \rangle \mid \langle CDSect \rangle \mid \langle PI \rangle \mid \langle Comment \rangle \;) \; \langle CharData \rangle? \;)^*$$

Character data

⟨*CharData*⟩ may contain any characters except

`<`, `&`, or the sequence `>]]`

Attribute values may not contain

- `"`, if delimited by `"`,
- `'`, if delimited by `'`,

These characters can be expressed by references.

Character data

```
1 <?xml version="1.1"?>
2 <abstract>
3   x^2 = y has no real solution for y < 0.
4   But there are solutions for y = 0 & for y > 0.
5 </abstract>
```

Figure 14: Forbidden characters in character data.

Character data

```
1 <?xml version="1.1"?>
2 <abstract>
3   x^2 = y has no real solution for y &lt; 0.
4   But there are solutions for y = 0 &amp;amp; for y > 0.
5 </abstract>
```

Figure 15: Using references in character data.

References

⟨*Reference*⟩ := ⟨*EntityRef*⟩ | ⟨*CharRef*⟩

⟨*CharRef*⟩ := `&#` [0-9]+ `;`
| `&#x` [0-9a-fA-F]+ `;`

⟨*EntityRef*⟩ := `&` ⟨*Name*⟩ `;`

There are five predefined entity references:

`<`	`>`	`&`	`&apos;`	`&quot;`
<	>	&	'	"

All other entities known from HTML (as `&auml;`) are **not** predefined in XML.

Custom entities can be defined in the document type declaration.

CDATA sections

CDATA sections allow the literal usage of all characters (except the sequence `]]>`).

⟨*CDSect*⟩ := `<![CDATA[` ⟨*CData*⟩ `]]>`

CDATA sections are typically used for longer text containing `<` or `&`.

CDATA sections are flat, i.e., there is no possibility to structure them with elements (as `<` or `&` are interpreted literally).

Character data and CDATA sections

```
1 <?xml version="1.1"?>
2 <abstract>
3   x^2 = y has no real solution for y &#3c; 0.
4   But there are solutions for y = 0 &#26; for y &#3e; 0.
5 </abstract>
```

Figure 16: Using numeric character references.

```
1 <?xml version="1.1"?>
2 <abstract><![CDATA[
3   x^2 = y has no real solution for y < 0.
4   But there are solutions for y = 0 & for y > 0.
5 ]]></abstract>
```

Figure 17: Using a CDATA-section.

Attribute values

```
1 <?xml version="1.1"?>
2 <book abstract="Discusses meaning of "wellformed"">
3   <author>John Doe</author>
4   <title>About wellformedness</title>
5 </book>
```

Figure 18: Literal usage of attribute delimiter.

```
1 <?xml version="1.1"?>
2 <book abstract='Discusses meaning of "wellformed"'>
3   <author>John Doe</author>
4   <title>About wellformedness</title>
5 </book>
```

Figure 19: Using different attribute delimiters.

```
1 <?xml version="1.1"?>
2 <book abstract="Discusses meaning of &quot;wellformed&quot;">
3   <author>John Doe</author>
4   <title>About wellformedness</title>
5 </book>
```

Figure 20: Using references in attribute values.

Comments

Comments can occur in the prolog, in the contents of elements and after the root element.

Comments are not allowed to contain the character sequence $--$.

$\langle Comment \rangle$:= $<!--\ \langle Char \rangle^*\ -->$

Comments

```
1 <?xml version="1.1"?>
2 <!-- list is not complete yet ! -->
3 <books>
4   <!-- yet to be ordered -->
5   <book>
6     <author><fn>Rainer</fn><sn>Eckstein</sn></author>
7     <author><fn>Silke</fn><sn>Eckstein</sn></author>
8     <title>XML und Datenmodellierung</title>
9     <year><!-- look up year of publication --></year>
10  </book>
11 </books>
12 <!-- eof -->
```

Figure 21: Comments in the prolog and in the contents of elements.


```
1 <?xml version="1.1"?>
2 <book>
3   <author><fn>Rainer</fn><sn>Eckstein</sn></author>
4   <author><fn>Silke</fn><sn>Eckstein</sn></author>
5   <title>XML und Datenmodellierung</title>
6   <year <!-- edition="1" -->>2004</year>
7 </book>
```

Figure 22: Comments in tags are not allowed.

```
1 <?xml version="1.1"?>
2 <books>
3   <!-- 2004 ------------------------------------ -->
4   <book>
5     <author><fn>Rainer</fn><sn>Eckstein</sn></author>
6     <author><fn>Silke</fn><sn>Eckstein</sn></author>
7     <title>XML und Datenmodellierung</title>
8     <year>2004</year>
9   </book>
10 </books>
```

Figure 23: -- is not allowed in comments.

Processing Instructions

Processing instructions (PIs) allow documents to contain instructions for applications.

⟨*PI*⟩ := `<?` ⟨*Name*⟩ (⟨*S*⟩ ⟨*Char*⟩*)? `?>`

The name of a PI must be different from `xml`.

Character encoding schemata

Character encoding schemata are specified by the name they are registered with at IANA (http://www.iana.org/assignments/character-sets), e.g.,

US-ASCII

ISO-8859-1

ISO-10646-UCS-2 or **csUnicode** (UCS2)

ISO-10646-UCS-4 or **csUCS4** (UCS4)

UTF-8

UTF-16

...

If no encoding is specified in the XML declaration, UTF-8 is assumed.

Character encoding schemata

```
1 <?xml version="1.1"?>
2 <page>
3   Grüß Gott !
4 </page>
```

Figure 24: Non-wellformed document (assumed that the file is ISO-8859-1 coded).

```
1 <?xml version="1.1" encoding="ISO-8859-1" ?>
2 <page>
3   Grüß Gott !
4 </page>
```

Figure 25: XML document coded in ISO-8859-1.

Language and Whitespaces

There are two predefined attributes,

- `xml:lang`

  and

- `xml:space`,

that can be used with any element.

`xml:lang` specifies the language of the character contents of elements and attributes with (RFC 3066)

- an ISO language code
  (http://www.loc.gov/standards/iso639-2/langcodes.html)

  or

- an IANA language code
  (http://www.iana.org/assignments/language-tags).

Language Attribute

Example ISO and IANA language codes:

language code	meaning	source
de	ISO	German
de-CH	ISO	German, Swiss variant
de-DE	ISO	German, German variant
en	ISO	English
en-US	ISO	US English
en-GB	ISO	Britain English
tlh	ISO	Klingon
de-1901	IANA	German, traditional orthography
de-1996	IANA	German, orthography of 1996
⋮	⋮	⋮

Language Attribute

```
1 <?xml version="1.1"?>
2 <page>
3   <p xml:lang="de">Guten <s>Morgen</s>!</p>
4   <p xml:lang="en">Good <s>morning</s>!</p>
5   <table>
6     <tr><td>USD</td><td>0</td><td>1</td><td>...</td></tr>
7     <tr><td>EUR</td><td>0</td><td>0.839818</td><td>...</td></tr>
8   </table>
9 </page>
```

Figure 26: Language attribute.

References