

Детальніше (основна офіційна інформація): <https://www.w3.org/XML/>
Навчальна сторінка w3c: http://www.w3schools.com/xml/xml_what_is.asp

XML (Extensible Markup Language) — це стандарт побудови мов розмітки ієрархічно структурованих даних. Є підмножиною мови SGML.

XML документи мають деревоподібну структуру, що складається із кореневого та дочірніх елементів.

Розглянемо приклад (http://www.w3schools.com/xml/xml_tree.asp):

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="web">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

Даному прикладові відповідає структура, зображена на рис. 1.

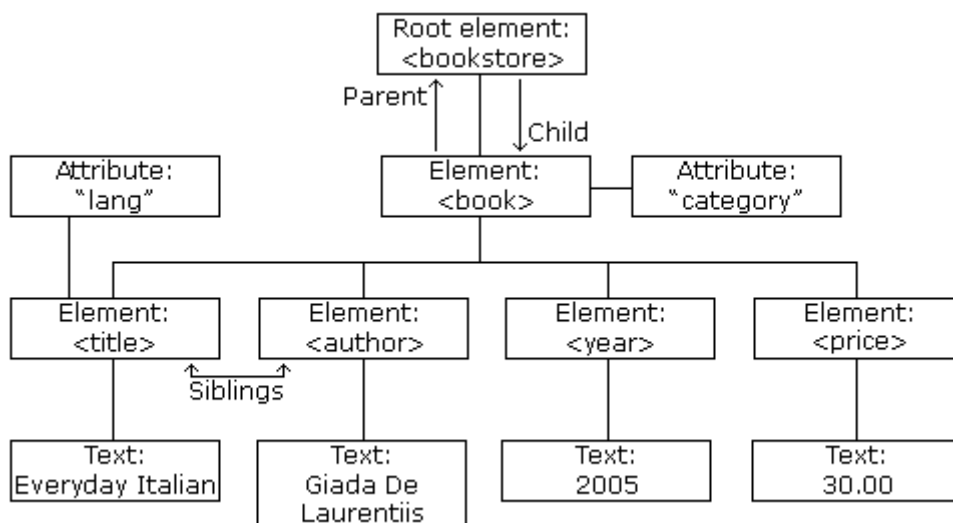


Рисунок 1 -Деревоподібна структура XML документа

Рядок `<?xml version="1.0" encoding="UTF-8"?>` називають прологом. Пролог є опційним. Якщо пролог існує, він повинен знаходитися на початкові XML-документа. Кодування XML-документів за замовчуванням — UTF-8, при цьому файл повинен бути збережений з кодуванням UTF-8. У разі, якщо кодування відрізняється від UTF-8, його слід явно вказати у прологові, скориставшись властивістю `encoding`. UTF-8 також є дефолтним кодуванням для HTML5, CSS, JavaScript, PHP та SQL.

Структура XML-документа є простою:

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

XML-документ є деревом елементів.

Складові XML-документа

З точки зору DTD, усі XML документи складаються з наступних блоків:

- елементи (elements);
- атрибути (attributes);
- сутності (entities);
- PCDATA;
- CDATA.

Документ обов'язково повинен мати кореневий елемент. У наведеному вище документі кореневим є елемент `<bookstore>`.

Дочірні елементи: `<book>`. У свою чергу, елемент `<book>` має власні дочірні елементи: `<title>`, `<author>`, `<year>`, `<price>`.

Атрибути: `category`, `lang`.

Елементи

Елементи — основні складові XML документів. Елементи можуть містити текст, інші елементи або бути порожніми.

Елементом є усе, що починається з відкриваючого (включно) тегу і закінчується закриваючим (включно) тегом, наприклад:

```
<price>29.99</price>
```

Елемент може містити:

- текст;
- інші елементи;
- атрибути;

Кожен XML елемент повинен мати заключний тег. Пролог не має заключного тегу, оскільки не є елементом XML.

XML елементи повинні бути коректно вкладені:

`<a>hello` — правильно;
`<a>hello` — неправильно.

Елемент, що не має вмісту, називають пустим:

`<element></element>` або `<element />`

Порожні елементи можуть мати атрибути.

Особливості формування імена елементів:

- імена елементів — регістрозалежні;
 - ім'я елемента повинно починатися з літери або символу підкреслювання;
 - ім'я елемента не може починатися із будь якої комбінації літер x, m та l з довільними регістрами (XML, xMl, XMl тощо);
 - ім'я елемента може містити літери, цифри, дефіси, символи підкреслювання та крапки;
 - ім'я елемента не може містити пробілів.
- Елемент може мати довільне ім'я (крім xml).

Імена елементів

Давайте елементам імена-описи: `<person>`, `<firstname>`, `<lastname>`.

Створюйте короткі та прості імена, наприклад: `<book_title>`, а не `<the_title_of_the_book>`.

Уникайте "-" (деякі програми знак мінус можуть інтерпретувати як віднімання одного значення від іншого).

Уникайте "." (деякі програми можуть інтерпретувати "first.name" як "name" - властивість об'єкта "first").

Уникайте ":". Двокрапка зарезервована для просторів імен.

Стилі імен елементів

Не існує визначених стилів імен XML елементів. Існує ряд стилів, які використовують на практиці (http://www.w3schools.com/xml/xml_elements.asp):

Стиль	Приклад	Опис
Нижній регістр	<code><firstname></code>	Усі символи — у нижньому регістрі
Верхній регістр	<code><FIRSTNAME></code>	Усі символи — у верхньому регістрі
Підкреслювання	<code><first_name></code>	Між словами — символ підкреслювання
Pascal регістр	<code><FirstName></code>	Перша літера кожного слова — у верхньому регістрі

Стиль	Приклад	Опис
Верблюжий регістр	<firstName>	Перша літера кожного, крім першого, слова — у верхньому регістрі

XML елементи — розширювані

XML елементи можуть бути розширені за потреби містити додаткову інформацію. Наприклад, нехай існує елемент:

```
<note>
  <to>Tove</to>
  <from>Jani</from>
  <body>Don't forget me this weekend!</body>
</note>
```

Нехай виникла потреба додати до елемента <note> деяку інформацію, після чого елемент набув вигляду:

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

Перевага XML у тому, що мова — розширювана. Парсер, орієнтований на обробку старого формату, без шкоди для програми працюватиме із новим форматом (звичай, що у цьому випадкові нові елементи будуть проігноровані).

Атрибути

Детальніше:

<https://www.w3.org/TR/REC-xml/#attdecls>,
http://www.w3schools.com/xml/xml_attributes.asp

XML елементи можуть мати атрибути у вигляді пар ім'я/значення. Призначення атрибутів — містити дані, що стосуються елемента.

Елементи можуть мати атрибути. Значення повинні бути взяті у лапки. Можна використовувати як одинарні, так і подвійні лапки.

```
<book category="web">
```

Якщо атрибут містить подвійні лапки, можна скористатися одинарними, наприклад:

```
<gangster name='George "Shotgun" Ziegler'>
```

Також можна використати сутність:

```
<gangster name="George &quot;Shotgun&quot; Ziegler">
```

XML елементи vs атрибути

Розглянемо приклади:

```
<person gender="female">  
  <firstname>Anna</firstname>  
  <lastname>Smith</lastname>  
</person>
```

```
<person>  
  <gender>female</gender>  
  <firstname>Anna</firstname>  
  <lastname>Smith</lastname>  
</person>
```

У першому випадкові `gender` є атрибутом. У другому — елементом. Обидва приклади надають однакову інформацію. Не існує правил щодо того, коли використовувати атрибути, а коли — елементи.

Наступні три документи містять одну і ту ж інформацію. У першому `date` — атрибут, у другому — елемент, у третьому — елемент подано у розширеному вигляді.

Атрибут:

```
<note date="2008-01-10">  
  <to>Tove</to>  
  <from>Jani</from>  
</note>
```

Елемент:

```
<note>
  <date>2008-01-10</date>
  <to>Tove</to>
  <from>Jani</from>
</note>
```

Розширений елемент:

```
<note>
  <date>
    <year>2008</year>
    <month>01</month>
    <day>10</day>
  </date>
  <to>Tove</to>
  <from>Jani</from>
</note>
```

Третій приклад є найбільш зручним з точки зору обробки.

Чи уникати атрибутів XML?

Візьміть до уваги деякі особливості використання атрибутів:

- елементи можуть володіти дочірніми елементами, атрибути — ні;
- атрибути не можуть описувати деревоподібні структури, елементи — можуть;
- елементи легко розширюються для використання у майбутньому, атрибути — ні.

Не створюйте елементів, подібних до наступного:

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

XML атрибути метаданих

Іноді елементи володіють ID посиланнями. Такі ID використовують для ідентифікації

(як у HTML). Наприклад:

```
<messages>
  <note id="501">
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
  </note>
  <note id="502">
    <to>Jani</to>
    <from>Tove</from>
    <heading>Re: Reminder</heading>
    <body>I will not</body>
  </note>
</messages>
```

Атрибути id використовують для ідентифікації елементів note. Id не є частиною елемента note. Тобто, метадані (дані про дані) слушно представляти як атрибути замість зберігання їх у елементах.

Сутності. Екранування символів

Сутності (Entities) використовують для створення скорочень для спеціальних символів.

Сутність складається з трьох частин: символа “&”, імені сутності та символа “;”.

Включення керуючих символів у тіло xml може призвести до помилок у роботі парсера. Розв'язком є екранування таких символів. Парсер інтерпретуватиме їх як дані і не плутатиме з розміткою:

Символ	Екранування	Пояснення
<	<	позначає початок тегу; екранування потрібне завжди
&	&	позначає початок посилання на сутність; екранування потрібне завжди
>	>	екранувати, чи ні – залежить від контексту; варто екранувати завжди
'	'	обов'язково екранувати у атрибутах, визначених за допомогою одинарних лапок; варто екранувати завжди
"	"	обов'язково екранувати у атрибутах, визначених за допомогою подвійних лапок; варто екранувати завжди

Варто завжди екранувати вказані символи, хоча це не завжди необхідно.

Коментарі у XML

Синтаксис коментарів у XML подібний до HTML:

```
<!-- This is a comment -->
```

Два послідовних дефіси усередині коментаря — заборонені.

```
<!-- This is a -- comment -->
```

New Line у XML

Windows-додатки зберігають новий рядок як CR + LF (carriage return +line feed);
Unix та Mac OSX: LF;
Old Mac: CR;
XML зберігає New Line як LF.

Пробіли

Послідовні пробіли у XML, на відміну від HTML, зберігаються. HTML відсікає пробільні символи і залишає тільки один.

Well Formed XML

XML документ, який відповідає синтаксичним правилам, називають добре сформованим "Well Formed" XML документом.

Що таке простір імен?

Згідно із <https://www.w3.org/TR/REC-xml-names/> простір імен — це набір елементів та атрибутів, який ідентифікують за допомогою IRI (Uniform Resource Identifier, <http://www.rfc-editor.org/rfc/rfc3986.txt>). Такий набір часом називають словником.

Синтаксис XML дозволяє змішувати вміст різних мов розмітки. Нехай існує елемент <table>, що містить дані таблиці. Припустимо, що існує інший елемент з таким же ідентифікатором — <table>, який містить інформацію про стіл (з атрибутами - довжиною, шириною).

Оскільки елементи володіють різними списками атрибутів та мають однакові імена, парсер стикнеться із конфліктом імен.

Уникнути конфлікту можна за допомогою просторів імен. Простір імен можна уявляти як сферу, усередині якої певний елемент має певну значення (семантику).

Позначають простір імен за допомогою спеціального ідентифікатора, який є нічим іншим, як символічний рядок. Зазвичай як ідентифікатор використовують HTTP URL, хоча жодного стосунку до мережі він не має і не інтерпретується як адреса ресурсу. Ідея проста:

HTTP URL — унікальні, тому можуть бути використані як ідентифікатори.

Ім'я елемента чи атрибута складається з двох частин: простору імен та локального імені. Повне ім'я має вигляд: {namespace URI}local name.

Оголошення простору імен

Розглянемо, як використати простір імен у XML. Простір імен оголошують за допомогою атрибута `xmlns`.

Оголосити простір імен можна двома способами: прив'язуючи простір імен до префікса або оголосити його таким, що має значення за замовчуванням (дефолтний).

Перший спосіб виглядає наступним чином `xmlns:prefix = "namespace URI"`, наприклад:

```
<prefix:foo xmlns:prefix = "http://prefix.com/">
  <prefix:bar />
</prefix:foo>
```

Простір імен за замовчуванням оголошують наступним чином:

```
<foo xmlns = "http://prefix.com/">
  <bar />
</foo>
```

У першому випадкові простір імен "<http://prefix.com/>" пов'язано із префіксом `prefix`. При цьому обидва імені семантично рівноправні. Вони не означають нічого. Можна, скажімо, замінити префікс “`prefix`” на “`bar`”. Це не викличе жодних семантичних змін. Дочірній елемент `<bar>` належить просторові імен "<http://prefix.com/>", про що свідчить відповідний префікс.

У другому випадкові простір імен оголошено за замовчуванням, область видимості його визначається елементом, у якому оголошено простір імен, та його вмістом. Усі елементи, що знаходяться в області видимості і не мають власного префікса, належать просторові імен за замовчуванням.

У XML атрибут, що не має префікса, не належить жодному з просторів імен. Простір імен за замовчуванням не поширюється на атрибути.

Випадок, коли простір імен оголошено за замовчуванням, видається зручнішим. У такому разі для чого існують обидва випадки оголошення?

Простори імен зазвичай прив'язують до префіксів тоді, коли їх — більше одного. Наприклад:

```
<foo xmlns = "http://prefix.com/">
  <a:bar xmlns:a = "http://prefix.com/a" />
</foo>
```

У даному прикладі елемент `foo` належить просторові імен "<http://prefix.com/>", а його дочірній елемент `bar` — просторові імен "<http://prefix.com/a>".

Простір імен можна оголосити усередині довільного елемента.

Оголошення можуть перевизначати одне одного:

```
<foo xmlns = "http://prefix.com/">
  <bar />
  <a:bar xmlns:a = "http://prefix.com/a" xmlns = "http://anotherbar.com/" />
    <bar />
  </a:bar>
</foo>
```

Тут: перший елемент <bar> належить просторові імен "<http://prefix.com/>", другий — "<http://prefix.com/a>", а третій — "<http://anotherbar.com/>".