

Лекція №5

Мова розмітки XML

Викл. Коваль В.В.

ОСІТ

2021р.



Основи web-розробки



Мова розмітки
XML

XML 1.1 (*Extensible Markup Language*)

XML – це мова розмітки, що описує цілий клас об'єктів даних, які називаються XML-документами.

Виділяють сім основних характеристик мови XML:

- XML пропонує метод структуризації файлу у вигляді текстового файлу.
- XML схожий на HTML.
- XML зрозумілий як комп'ютеру так і людині.
- XML утворює ціле сімейство технологій.
- XML достатньо гнучкий.
- XML достатньо новий, але у нього глибоке коріння.
- XML вільний від ліцензійних відрахувань, платформенно-незалежний, має широку підтримку.

Використання XML

- Інтеграція даних з різних джерел
- Локальна обробка даних на клієнті.
- Перегляд і маніпулювання даними в різних форматах.
- Можливість часткового оновлення даних.

Well-formed XML

- Кожен відкриваючий тег, що визначає деяку область даних у документі, обов'язково повинен мати свого закриваючого "напарника".
- У XML враховується регістр символів.
- Всі значення атрибутів у визначенні тегів повинні бути в лапках.
- Вся інформація, що розташовується між початковим і кінцевими тегами, розглядається в XML як дані і тому враховуються всі символи форматування (тобто пробіли, переноси рядків, табуляції не ігноруються, як в HTML).
- Документ повинен мати тільки один елемент верхнього рівня. Всі інші елементи повинні бути вкладені в елемент верхнього рівня.

XML. Конструкції мови

Коментарі

<! - коментар ->

Спеціальні символи

< , > " или $(десятичная форма записи),

Атрибути

- - ім'я має починатися з букви або символу підкреслення (_), після чого можуть слідувати або не слідувати інші літери, цифри, символи точки (.), Тире (-) або підкреслення;
- - кожне ім'я атрибута може тільки один раз бути присутнім в одному і тому ж початковому тезі або в тезі порожнього елемента.
- Наприклад:
- <LIST _1stPlace = "Sam"> - правильно
- <LIST 1stPlace = "Sam"> - неправильно

XML. Атрибути

- рядок може бути як в одинарних ('), так і в подвійних лапках (");
- рядок не може містити в собі той же символ лапок, якими він обмежений;
- рядок може містити посилання на символ або посилання на внутрішні примітиви загального призначення;
- рядок не може містити символ <(Синтаксичний аналізатор може сприйняти цей символ як початок опису XML-розмітки.)
- рядок не може містити символ &, якщо це не посилання на символ або примітив.

Створення екземплярів

<ім'я_елемента

ім'я_атрибута= “значення”

ім'я_атрибута= “значення”

ім'я_атрибута=“значення”>

Зміст елемента відповідного елемента

</ім'я_атрибута>

<account type= “checking” currency= “Gryvnja”>

<name>Івченко</name>

<balance>18623,12</balance></account>

The XML page cannot be displayed
Cannot view XML input using style sheet. Please
correct the error and then click the Refresh button, or try
again later.

**A string literal was expected, but no opening quote
character was found. Error processing resource**

file:///D:/WORK/Web-pro...

<?xml version= 1.0">

-----^

Рисунок 1 - Результат відображення неправильно
оформленого атрибута

Порожні елементи

`<ім'я _елемента />`

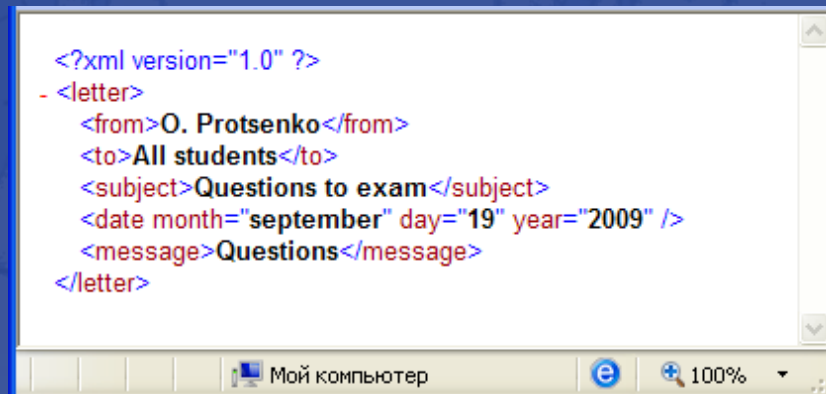
Наприклад,

`<date month="September" day="19" year="2010" />`

Приклад,

`<?xml version="1.0" encoding="UTF-8"?>`

- `<letter>`
- `<from> O. Protsenko</from>`
- `<to> All students </to>`
- `<subject> Questions to exam</subject>`
- `<date month="September" day="19"`
- `<message> Questions </message>`
- `</letter>`

A screenshot of a text editor window showing an XML document. The XML content is:

```
<?xml version="1.0" ?>
<letter>
  <from>O. Protsenko</from>
  <to>All students</to>
  <subject>Questions to exam</subject>
  <date month="september" day="19" year="2009" />
  <message>Questions</message>
</letter>
```

The text is color-coded: blue for tags, red for attribute values, and black for text. The editor window has a title bar and a taskbar at the bottom showing 'Мой компьютер' and a 100% zoom level.

XML. Конструкції мови

Директиви аналізатора

<? тип інструкція?>

CDATA <![CDATA [...]]>

Необхідна щоб задати область документа, яку при розборі аналізатор буде розглядати як простий текст, ігноруючи будь-які інструкції та спеціальні символи, але, на відміну від коментарів, мати можливість використовувати їх в додатку.

DTD (Document Type Definition) <! DOCTYPE ...>

- Містить правила, що описують структуру документа
- Документ автоматично перевіряється на відповідність цим правилам
- Описує дочірні елементи і атрибути для кожного елемента

Визначення типу документа

Схема DTD надає шаблон розмітки документа, в якому вказується наявність, порядок проходження і розміщення елементів і їх атрибутів у документі.

Оголошення змісту елемента або атрибута в схемі DTD називається *моделлю змісту для цього елемента або атрибута*.

<! ELEMENT ім'я_елемента (модель)>

Valid XML

Document Type Definition може бути описаний в документі або зовнішньому файлі:

- `<!DOCTYPE document-root [...]>`
- `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">`
- `<!DOCTYPE people_list SYSTEM "example.dtd">`

```
<!DOCTYPE COLLECTION [  
  <!ENTITY greeting "helloworld">  
  <!ELEMENT tag_name1 EMPTY>  
  <!ELEMENT tag_name2 (#PCDATA|ANY|(tag,tag2)*,tag3?)>  
  <!ATTLIST payment type CDATA #REQUIRED>  
>
```

Внутрішні схеми DTD

```
<?xml version="1.0" encoding="windows-1251"
    ?>
```

```
<!DOCTYPE note [
    <!ELEMENT note (#PCDATA)>
]>
```

```
<note> Завтра о 12.45 відбудеться лекція з
Web-дизайну</note>
```

Елемент, що містить дочірній елемент

- 1: <?xml version="1.0" encoding="windows-1251" ?>
- 2: <!DOCTYPE note [
- 3: <!ELEMENT note (text)>
- 4: <!ELEMENT text (#PCDATA)>
- 5:]>
- 6: <note>
- 7: <text>
- 8: Завтра о 12.45 відбудеться лекція з Web-дизайну
- 9: </text>
- 10: </note>

Оголошення порожнього елемента

<! ELEMENT і'мя_елемента EMPTY>

1:<?xml version="1.0" encoding="windows-1251" ?>

- 2: <!DOCTYPE note [
• 3: <!ELEMENT note (time,datetext)>
• 4: <!ELEMENT time EMPTY>
• 5: <!ELEMENT date EMPTY>
• 6: <!ELEMENT text (#PCDATA)>
• 7:]>
• 8: <note>
• 9: <time />
• 10: <date />
• 11: <text>
• 12: Завтра о 12.45 відбудеться лекція з Web-дизайну
• 13: </text>
• 14: </note>

Використання ключового слова ANY

- 1: <?xml version="1.0" encoding="windows-1251" ?>
- 2: <!DOCTYPE note [
- 3: <!ELEMENT note ANY>
- 4: <!ELEMENT time EMPTY>
- 5: <!ELEMENT date EMPTY>
- 6: <!ELEMENT text (#PCDATA)>
- 7:]>
- 8: <note>
- 9: <time />
- 10: <date />
- 11: <text>
- 12: Завтра о 12.45 відбудеться лекція з Web-дизайну
- 13: </text>
- 14: </note>

Змішаний вміст елементів

- 1: <?xml version= "1.0" encoding= "windows-1251"?>
- 2: <!DOCTYPE note [
• 3: <! ELEMENT note (#PCDATA | time | date | text)*>
• 4: <!ELEMENT time EMPTY>
• 5: <!ELEMENT date EMPTY>
• 6: <!ELEMENT text (#PCDATA)>
• 7:]>
- 8: <note>Увага!!!
- 9: <time />
- 10: <date />
- 11: <text>
- 12: Завтра о 12.45 відбудеться лекція з Web-дизайну
- 13: </text>
- 14: </note>

Використання атрибутів

```
<! ATTLIST ім'я_елемента  
ім'я_атрибута1 (тип) значення за замовченням  
ім'я_атрибута1 (тип) значення за замовченням >
```

Фундаментальні типи атрибутів DTD схем:

- Рядки, що зазначаються за допомогою ключового слова CDATA;
- Марковані атрибути, що зазначаються за допомогою визначених раніше маркерів;
- Атрибути з переліченням, що пропонують цілий ряд значень.

Стандартні значення атрибутів:

- **#REQUIRED** – вказує на те, що атрибут має бути вказаний;
- **#FIXED** – вказує на фіксоване значення атрибута. Якщо значення атрибута відрізняється від оголошеного, документ не вважається правильним.
- **#IMPLIED** – атрибут необов'язковий. Це означає, що при обробці елемента парсер може використовувати будь-яке значення, якщо в цьому є необхідність.

Складемо DTD схему для прикладу, де позначимо, що атрибути задаються рядками.

```
<?xml version="1.0" encoding="windows-1251" ?>
<!DOCTYPE note [
  <!ELEMENT note    (#PCDATA | time | date | text)* >
  <!ELEMENT time    EMPTY>
  <!ELEMENT date    EMPTY>
  <!ATTLIST date
    day  CDATA #REQUIRED
          month CDATA #REQUIRED
          year CDATA #FIXED "2009">
  <!ATTLIST time
    hh  CDATA #IMPLIED
          mm  CDATA #IMPLIED
          ss  CDATA #IMPLIED>
  <!ELEMENT text    (#PCDATA)>
    ]>
  <note>Увага!!!<time hh="13" mm="00"/>
  <date day="19" month="10" year="2009"/>
    <text>Завтра о 12.45 відбудеться лекція з Web-дизайну
    </text>
  </note>
```


Визначення атрибутів маркованого типу

Розглядають *4 різних типи маркованих атрибутів*:

- **ID** – унікальним чином ідентифікує об'єкт;
- **IDREF** – вказує на елементи, що містять атрибути ID;
- **ENTITIES** – посилання на зовнішній елемент;
- **NMTOKEN** – містить букви, цифри, крапки, знаки підкреслення, переноси і двокрапки, але не пропуски.

```
1:<?xml version="1.0" encoding= "windows-1251" ?>
2:  <!DOCTYPE note [
3:    <!ELEMENT note (date*,time+, text+, comment+)*>
4:    <!ELEMENT date EMPTY>
5:    <!ATTLIST date
6:      day  CDATA #REQUIRED
7:      month CDATA #REQUIRED
8:      year CDATA #FIXED "2009">
9:
10:  <!ELEMENT time EMPTY>
11:  <!ATTLIST time
12:    hh CDATA #REQUIRED
13:    mm CDATA #REQUIRED
14:    ss CDATA#IMPLIED>
15:
16:  <!ELEMENT text (#PCDATA)>
17:  <!ATTLIST text
18:    number ID #REQUIRED
19:    from CDATA #REQUIRED>
20:
21:  <!ELEMENT comment (#PCDATA)>
22:  <!ATTLIST comment
23:    txt IDREF #REQUIRED>
24:
25:  ]>
```

```
26:<note>
27: <date day="29" month="10" year="2009"/>
28:
29: <time hh= "08" mm= "15" ss= "17" />
30: <text number="n1" from="Проценко О.Б.">
31: Завтра відбудеться лекція з Web-дизану о 13.25
32: </text>
33: <comment txt= "n1">
34:     Не забути конспект
35: </comment>
36:
37: <time hh= "13" mm= "15" />
38: <text number= "n2" from= "Керівник відділу">
39: Терміново здати звіт. Завтра останній термін
40: </text>
41:     <comment txt= "n2">
42: Що важливіше лекція або звіт?
43: </comment>
44:</note>
```


Використання зовнішніх схем DTD

```
<!DOCTYPE кореневий_елемент SYSTEM  
    "my_dtd.dtd">
```

СХЕМИ XDR

Схеми XDR визначають елементи, подані в екземплярі XML, а також будь-які атрибути, пов'язані з цими елементами, крім того в схемах XDR визначаються взаємозв'язки елементів XML-документа.

<? xml version= “1.0”?>

1: <Schema

2: name=“ім’я_схеми”

3: xmlns=“urn:schemas-microsoft-com: xml-data”

4: xmlns:dt=“urn:schemas-microsoft-com:
datatypes”>

5: <!--Оголошення інших елементів -->

6: <ElementType />

7: <!--кінець списку оголошення елементів-->

8: </Schema>

Опис елементів в XDR-схемах має такий синтаксис:

```
<ElementType  
  name= "idref"  
  content= "{empty | textOnly | eltOnly | mixed}"  
  dt:type= "datatype"  
  model= "{open | closed}"  
  order= "{one | seq | many}" />
```

name — задає ім'я елемента

content— вказує на те, що містить описуваний елемент. Допустимими значеннями цього атрибута є:

empty — порожній елемент;

eltOnly — означає, що елемент може бути тільки контейнером для інших елементів;

textOnly — може містити тільки текст;

mixed — змішані дані (стандартне значення).

dt:type — оголошує тип даних елемента. Префікс дозволяє вказати простір імен для URI адрес.

model — дозволяє (open) або забороняє (closed) використовувати елементи, не визначені в схемі XDR.

order — оголошує порядок проходження дочірніх елементів екземпляра XML

one — припускає наявність одного елемента;

seq — вказує елементи в строгому порядку;

many — припускає наявність будь-якої кількості елементів.

1: <?xml version= “1.0”>

2: <Schema

3: name= “example”

4: xmlns=“urn:schemas-microsoft-com: xml-data”

5: xmlns:dt=“urn:schemas-microsoft-com:
datatypes”>

6: <ElementType name=“note” model= “closed”
content = “textOnly” dt:type= “string”/>

7: </Schema>

ОГЛОШЕННЯ ElementType ВКЛАДЕНИХ ЕЛЕМЕНТІВ

```
< ElementType name= ‘.’ >
```

```
<element type= ‘ім'я вкладеного елемента’>
```

```
</ElementType >
```

```
</ElementType name= ‘ім'я вкладеного елемента’ / >
```

Наприклад,

example 6.xml

```
<?xml version=“1.0” encoding= “windows-1251”?>
```

```
<note xmlns=‘x-shemas:example_6.xdr’>
```

```
<text>Завтра о 12.45 лекція з Web-дизайну</text></note>
```

1: <?xml version= ‘1.0’?>

2: <Schema

3: name= ‘example_6’

4: xmlns= ‘urn shemas-microsoft-com: xml – data’

5: xmlns:dt=‘urn: shemas-microsoft-com: datatypes’>

6: <ElementType name= ‘note’ model= ‘closed’ content=‘eltOnly’>

7: <element name= ‘text’/>

8: </ElementType>

9: <ElementType name=‘text’ model= ‘closed’ content=‘textOnly’ dt:type=‘string’/>

10: </Schema>

Дочірні елементи для елемента `ElementType` можуть бути такі елементи:

- `Element` – оголошує дочірній елемент
- `Description` – забезпечує опис `ElementType` елемента
- `Datatype` – визначає тип даних `ElementType` елемента
- `Group` – визначає порядок проходження елемента
- `Attribute Type` – визначає атрибут
- `Attribute` – визначає відомості про дочірній елемент `Attribute Type`.

Декілька елементів

Приклад example.xml:

```
<?xml version= "1.0" encoding= "windows-1251"  
  ?>
```

```
<note xmlns= 'x-schemas:example.xdr'>
```

```
<time />
```

```
<date />
```

```
<text>Завтра о 12.45 лекція з Web-дизайну
```

```
</text>
```

```
</note>
```



```
1: <?xml version= '1.0'>
2: <Schema
3:   name= 'example'
4:   xmlns= 'urn: shemas-microsoft-com: xml – data'
5:   xmlns:dt= 'urn:shemas-microsoft-com: datatypes'>
6:   <ElementType name='note' model='closed' content='eltOnly'>
7:       <element name= 'time' />
8:       <elementname= 'date' />
9:       <element name= 'text' />
10: </ElementType>
11: <ElementType name= 'time' model= 'closed' content= 'empty' />
12: <ElementType name= 'data' model= 'closed' content= 'empty' />
13: <ElementType name= 'text' model= 'closed' content= 'textOnly'
    dt:type= 'string' />
14: </Schema>
```


ЕЛЕМЕНТ AttributeType

```
<AttributeType  
  default='default-value'  
  dt:type='primitiv-type'  
  dt:values='enumerated-values'  
  name='idref'  
  required='{yes | no}'>.
```

default — значення атрибута за замовчанням. Наприклад, якщо атрибут відноситься до переліченого типу, значення за замовчанням повинне вказуватися у списку;

dt:type — тип даних для атрибута певного типу: entity, entities, enumeration, id, idref, idrefs, nmtoken, nmtokens, notation, string. При обраному типі enumeration необхідно вказувати і атрибут dt:values;

dt:values — містить всі допустимі значення, якщо dt:type='enumeration';

name — ім'я типу атрибута. Цей атрибут обов'язковий;

required — указує на необов'язкову наявність атрибута в описі елемента.

```
<attribute  
default='default-value'  
type='attribute-type'  
required= '{yes | no}'>
```

- **default** – значення атрибута за замовчанням. Має перевагу перед будь-яким значенням за замовчанням, вказаним в елементі `AttributeType`;
- **type** – ім'я елемента `AttributeType`, яке визначене в даній схемі (або в іншій, вказане за допомогою відповідного простору імен). Вказане значення повинне відповідати значенню атрибута `name` в `AttributeType`;
- **required** – вказує на обов'язкову наявність атрибута в описі елемента. Необов'язковий, якщо необхідний атрибут присутній в `AttributeType`.

example.xml

1: <?xml version= “1.0” encoding= “windows-1251” ?>

2: <note xmlns= ‘x-shemas:example.xdr’

3: time= ‘12:54:13’

4: date= ‘17-10-2009’>

5: <text>

6: Завтра о 12.45 лекція з Web-дизайну

7: </text>

8: </note>

example.xdr

```
1:  <?xml version= '1.0' ?>
2:  <Schema
3:      name='example'
4:      xmlns= 'urn:shemas-microsoft-com:xml-data'
5:      xmlns:dt='urt:shemas-microsoft-com:datatype'>
6:  <ElementType name='note' content='eltOnly' model='close' >
7:      <AttributeType
8:          dt:type='time'
9:          name= 'time'
10:         required='yes'/>
11:      <attribute
12:          type= 'time'
13:          required='yes'/>
14:      <AttributeType
15:          dt:type='date'
16:          name= 'date'
17:          required= 'yes'/>
18:      <attribute
19:          type= 'date'
20:          required= 'yes'/>
21:      <elementname = 'text'/>
22:  </ElementType>
23:  <ElementType name='text content='textOnly' model='close'/>
24:  </Schema>
```


ТИПИ ДАНИХ У XDR-СХЕМАХ

bin.base64	– двійкові дані з використанням шифрування MIME;
bin.hex	– визначає двійкові дані в шістнадцятиричному форматі;
boolean	– 0 або 1;
char	– символ (рядок, що складається з одного символу);
date	– указує на дату без часу у форматі ISO-8601, наприклад, 2009-10-16. Проте правильність зазначення дати не перевіряється, тобто допустиме значення і дати 17-41-59;
dateTime	– вказує дату (вказівка часу необов'язкова) в підмножині формату ISO-8601, наприклад, 2009-10-16 T09:57:12;
dateTime.tz	– вказує дату (вказівка часу і часового поясу необов'язково) в підмножині формату ISO-8601. Наприклад, 2009-10-16 T09:57:12-02:00;
fixed.14.4	– визначає десяткове число, що містить не більше 14 знаків зліва і не більше 4 цифр праворуч від десяткової крапки;
float	– визначає дійсне число (без обмежень на кількість цифр), може містити знак дробу, а також показник ступеня, значення від 1.797693E+308 до 2.22507E-308;
int	– ціле число з необов'язковим знаком;
number	– визначає число без обмеження цифр, може містити знак дробу і ступінь значення від 1.797693E+308 до 2.22507E-308;
time	– вказує час без дати і часового поясу в підмножині формату ISO – 8601, наприклад, 10:14:51;
time.tz	– вказує час і часовий пояс, наприклад, 10:14:51+02:00;

ІНДИКАТОРИ ВХОДЖЕННЯ В XDR-СХЕМАХ

```
<element  
type= 'тип_елемента'  
    minOccurs = '{0|1}'  
    maxOccurs='{1|*}' />
```

- example_9.xml
- 1:<?xml version= "1.0" encoding= "windows-1251"?>
- 2: <note xmlns='x-schemas:example.xdr'
- 3: time= '12:54:13'
- 4: date= '17-10-2009'>
- 5: <text>
- 6:18.10.2009 у 12.45 лекція з Web-дизайну
- 7: </text>
- 8: <text>
- 9: 18.10.2009 о 13.25 практика з Web-дизайну
- 10: </text>
- 11: <text>
- 12:20.10.2009 знову о 12.45 лекція з Web-дизайну
- 13: </text>
- 14: </note>

- example_9.xdr
- 1: <?xml version='1.0' ?>
- 2: <Schema
- 3: name='example'
- 4: xmlns='urn:shemas-microsoft-com:xml-data'
- 5: xmlns:dt='urn:shemas-microsoft-com: datatype'>
- 6: <ElementType name='note' content='eltOnly' model='close' >
- 7: <AttributeType
- 8: dt:type='time'
- 9: name='time'
- 10: required='yes'/>
- 11: <Attribute
- 12: type='time'
- 13: required='yes'/>
- 14: <AttributeType
- 15: dt:type='date'
- 16: name='date'
- 17: required='yes'/>
- 18: <Attribute
- 19: type='date'
- 20: required='yes'/>
- 21: <element name='text' minOccurs='1' maxOccurs='*' />
- 22: </ElementType>
- 23: <ElementType name='text' content='textOnly' model='close'/>
- 24: </Schema>

МОВА ВИЗНАЧЕННЯ СХЕМ XML (XSD)

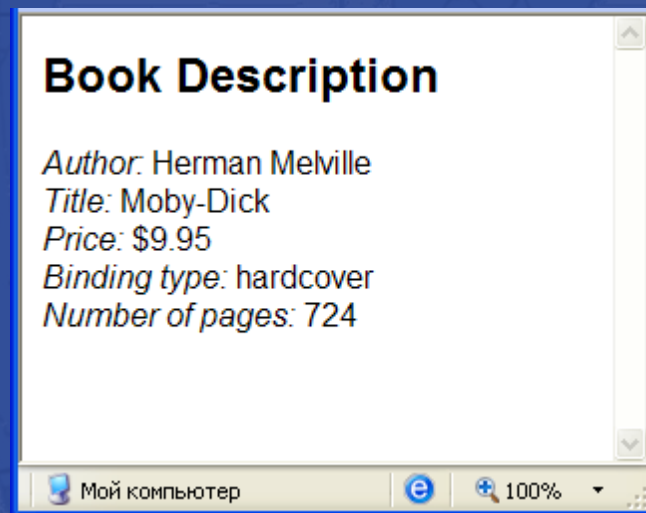
ОСНОВИ ВИКОРИСТАННЯ XSL-ТАБЛИЦЬ СТИЛІВ

example.xsl

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">
    <H2>Book Description</H2>
    <SPAN STYLE="font-style:italic">Author: </SPAN>
    <xsl:value-of select="BOOK/AUTHOR"/><BR/>
    <SPAN STYLE="font-style:italic">Title: </SPAN>
    <xsl:value-of select="BOOK/TITLE"/><BR/>
    <SPAN STYLE="font-style:italic">Price: </SPAN>
    <xsl:value-of select="BOOK/PRICE"/><BR/>
    <SPAN STYLE="font-style:italic">Binding type: </SPAN>
    <xsl:value-of select="BOOK/BINDING"/><BR/>
    <SPAN STYLE="font-style:italic">Number of pages: </SPAN>
    <xsl:value-of select="BOOK/PAGES"/>
  </xsl:template>
</xsl:stylesheet>
```



```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="example.xsl"?>
<BOOK>
  <TITLE>Moby-Dick</TITLE>
  <AUTHOR>
    <FIRSTNAME>Herman</FIRSTNAME>
    <LASTNAME>Melville</LASTNAME>
  </AUTHOR>
  <BINDING>hardcover</BINDING>
  <PAGES>724</PAGES>
  <PRICE>$9.95</PRICE>
</BOOK>
```



Відображення змінного числа елементів

```
<INVENTORY>
<BOOK>
<TITLE>The Adventures of Huckleberry Finn</TITLE>
<AUTHOR>
  <FIRSTNAME>Mark</FIRSTNAME>
  <LASTNAME>Twain</LASTNAME>
</AUTHOR>
<BINDING>mass market paperback</BINDING>
<PAGES>298</PAGES>
<PRICE>$5.49</PRICE>
</BOOK>
```

```
<BOOK>
<TITLE>The Adventures of Tom Sawyer</TITLE>
<AUTHOR>
  <FIRSTNAME>Mark</FIRSTNAME>
  <LASTNAME>Twain</LASTNAME>
</AUTHOR>
<BINDING>mass market paperback</BINDING>
<PAGES>205</PAGES>
<PRICE>$4.75</PRICE>
</BOOK>
```

```
<BOOK>
<TITLE>The Ambassadors</TITLE>
<AUTHOR>
  <FIRSTNAME>Henry</FIRSTNAME>
  <LASTNAME>James</LASTNAME>
</AUTHOR>
<BINDING>mass market paperback</BINDING>
<PAGES>305</PAGES>
<PRICE>$5.95</PRICE>
</BOOK>
</INVENTORY>
```

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">
    <H2>Book Description</H2>
    <SPAN STYLE="font-style:italic">Author: </SPAN>
    <xsl:value-of select="INVENTORY/BOOK/AUTHOR"/><BR/>
    <SPAN STYLE="font-style:italic">Title: </SPAN>
    <xsl:value-of select="INVENTORY/BOOK/TITLE"/><BR/>
    <SPAN STYLE="font-style:italic">Price: </SPAN>
    <xsl:value-of select="INVENTORY/BOOK/PRICE"/><BR/>
    <SPAN STYLE="font-style:italic">Binding type: </SPAN>
    <xsl:value-of select="INVENTORY/BOOK/BINDING"/><BR/>
    <SPAN STYLE="font-style:italic">Number of pages: </SPAN>
    <xsl:value-of select="INVENTORY/BOOK/PAGES"/>
  </xsl:template>
</xsl:stylesheet>
```


Book Description

Author: Mark Twain

Title: The Adventures of Huckleberry Finn

Price: \$5.49

Binding type: mass market paperback

Number of pages: 298



Мой компьютер



100%

```
<xsl:for-each select="INVENTORY/BOOK">
  <SPAN STYLE="font-style:italic">Title: </SPAN>
  <xsl:value-of select="TITLE"/><BR />
  <SPAN STYLE="font-style:italic">Author: </SPAN>
  <xsl:value-of select="AUTHOR"/><BR />
  <SPAN STYLE="font-style:italic">Binding type: </SPAN>
  <xsl:value-of select="BINDING"/><BR />
  <SPAN STYLE="font-style:italic">Number of pages:
  </SPAN>
  <xsl:value-of select="PAGES"/><BR />
  <SPAN STYLE="font-style:italic">Price: </SPAN>
  <xsl:value-of select="PRICE"/><P />
</xsl:for-each>
```

ФІЛЬТРАЦІЯ І СОРТУВАННЯ ДАНИХ XML

Фільтрація

```
<xsl:template match="BOOK [BINDING='trade paperback']">
```

```
match="ITEM[CD]"
```

```
match="SHIRT[COLOR='red']"
```

Сортування

```
<xsl:for-each select="INVENTORY/BOOK"
```

```
order-by="+AUTHOR/LASTNAME;  
+AUTHOR/FIRSTNAME">
```

```
<xsl:apply-templates select="INVENTORY/BOOK"  
order-by="-TITLE">
```


example.xsl

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">
    <H2>Book Inventory</H2>
    <xsl:for-each
      select="INVENTORY/BOOK[BINDING='trade paperback']"
      order-by="+AUTHOR/LASTNAME; +AUTHOR/FIRSTNAME">
      <SPAN STYLE="font-style:italic">Author: </SPAN>
      <xsl:value-of select="AUTHOR"/><BR />
      <SPAN STYLE="font-style:italic">Title: </SPAN>
      <xsl:value-of select="TITLE"/><BR />
      <SPAN STYLE="font-style:italic">Binding type </SPAN>
      <xsl:value-of select="BINDING"/><BR />
      <SPAN STYLE="font-style:italic">Number of pages: </SPAN>
      <xsl:value-of select="PAGES"/><BR />
      <SPAN STYLE="font-style:italic">Price: </SPAN>
      <xsl:value-of select="PRICE"/><P />
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>
```

ДОСТУП ДО АТРИБУТІВ XML

```
<xsl:for-each  
  select="INVENTORY/BOOK[@InStock='yes'  
]">
```

```
<xsl:value-of select="AUTHOR/@Born"/>
```

Використання каскадних таблиць стилів

example.css

BOOK

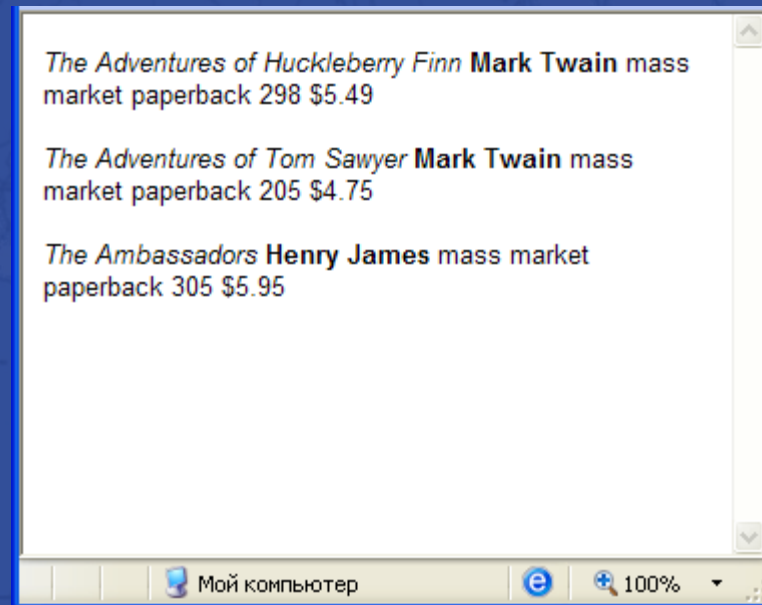
```
{display:block;  
margin-top:12pt;  
font-size:10pt}
```

TITLE

```
{font-style:italic}
```

AUTHOR

```
{font-weight:bold}
```



```
<?xml version="1.0"?>
```

```
<?xml-stylesheet type="text/css" href="example.css"?>
```

```
<INVENTORY>
```

```
</INVENTORY>
```


Розбір XML документів

XML парсери:

- DOM-парсер (Document Object Model) - попередньо аналізується XML-документ і зберігається дерево елементів XML в оперативній пам'яті. Вимогливий до ресурсів.
- SAX-парсер (Simple API for XML) - парсер, який базується на подіях (event-based). Швидкий, за рахунок розбору тільки конкретної частини документа. Займає мало пам'яті.



Дякую за увагу!

Подивитись
