

## XML

EXtensible Markup Language :

- ✓ описва дървовидно структурирана информация;
- ✓ създаден за съхранение и пренос на данни;
- ✓ Markup Language като HTML;

доц. Стоянова

1

## XML

- ✓ XML е софтуерно и хардуерно независимо средство за пренос на данни.
- ✓ Няма предварително дефинирани тагове

доц. Стоянова

2

## XML

**Основа на XML**

Код:

```
<?xml version="1.0" encoding="windows-1251"?>
<student>
  <name>Иван</name>
  <egn>8201273840</egn>
  <adress>Младост 3</adress>
  <course>Информационни системи и технологии</course>
</student>
```

доц. Стоянова

3

## XML

Резултат:

```
<?xml version="1.0" encoding="windows-1251"?>
<student>
  <name>Иван</name>
  <egn>8201273840</egn>
  <adress>Младост 3</adress>
  <course>Информационни системи и технологии</course>
</student>
```

доц. Стоянова

4

## XML

**Всеки един XML документ трябва да отговаря на синтаксиса на XML.**

**Тагове**

Таговете - по двойки.

Една двойка определя един елемент от структурата, която XML документът описва.

доц. Стоянова

5

## XML

Пример:

```
<name>Иван</name>
```

Името на елемента може да съдържа всякакви букви (не символи) в Unicode формат, цифри, знакът точка ('.'), тире ('-') и долна черта ('\_').

доц. Стоянова

6

## XML

Името не може да започва с цифра, тире ('-') или точка ('.').

Имена, които започват с "xml" (без значение малки или големи букви), са резервирани !

Между символа '<' и името на тага не може да има други символи .

Отварящият таг завършва със символа '>'.

## XML

Пример:

`<name>, <first_name>, <I.name>, <_>, <last-name>`

`<name>, <first_name>, <name>, <user@name>, </name>`

## XML

Името на затварящия таг трябва да е същото като на отварящия.

Пример:

`<name>, <first_name>, </I.name>, <_>, </last-name>`

`<name>, <first_name>, <name>, <user@name>, </name>`

`<></>` - правилен, но не обяснява съдържанието!

## XML

**Описанието на един елемент съдържа:**

отварящ таг;  
затварящ таг;  
текст между таговете.

## XML

**Регистър (Case sensitive)**

`<Iname>`            `</IName>`

`<student>`        `</Student>`

## XML

**Влагане**

За реализиране на дървовидната структура на документа:

**елементите трябва да бъдат затваряни в обратен ред на тяхното отваряне.**

## XML

Пример:

```
<student>
  <name>Иван</name>
</student>
```

И

```
<student>
  <name>Иван</student>
</name>
```

доц. Стоянова

13

## XML

**Root елемент -**

```
<?xml version="1.0" encoding="windows-1251"?>
<students_list>
<student>
  <name>Иван</name>
  <egn>8201273840</egn>
  <address>Младост 3</address>
  <course>Информационни системи и технологии</course>
</student>
<student>
  <name>Петър</name>
  <egn>8203143840</egn>
  <address>Младост 1</address>
  <course>Информационни системи и технологии</course>
</student>
</students_list>
```

доц. Стоянова

14

## XML

Ако <students\_list> </students\_list> се премахнат, то документът няма да отговаря на изискванията.

?

Root element

доц. Стоянова

15

## XML

**Резултат:**

```
<?xml version="1.0" encoding="windows-1251" ?>
- <students_list>
- <student>
  <name>Иван</name>
  <egn>8201273840</egn>
  <address>Младост 3</address>
  <course>Информационни системи и технологии</course>
</student>
- <student>
  <name>Петър</name>
  <egn>8203143840</egn>
  <address>Младост 1</address>
  <course>Информационни системи и технологии</course>
</student>
</students_list>
```

доц. Стоянова

16

## XML

```
<?xml version="1.0" encoding="windows-1251" ?>
- <students_list>
  + <student>
  + <student>
</students_list>
```

доц. Стоянова

17

## XML

**Атрибути**

В отварящия таг, освен име на елемента могат да се дефинират и т.нар. **атрибути**.

Всеки атрибут - име, стойност - носи съпътстваща информация за елемента.

Атрибутите се изреждат след името на елемента, отделени с празно място.

доц. Стоянова

18

## XML

**Форматът им е име="стойност"**

името - същите изисквания, както и името на елемента  
стойността - в единични ('...') или двойни ("...").

Ако вътре в стойността трябва нещо да е също заградено в кавички – тогава цялата стойност се огражда в единия вид, а вътре използваме другия вид.

## XML

## Пример:

```
<file filename="data.txt">
<size>100</size>
<type>text</type>
</file>
```

## XML

## Пример:

```
<file>
<name>data.txt</name>
<size>100</size>
<type>text</type>
</file>
```

Информация - атрибут или елемент ?

## XML

Някои от проблемите при използване на атрибути:

- не могат да съдържат многобройни стойности;
- не могат да съдържат дървовидни структури;
- не могат лесно да се разширяват от гледна точка на бъдещи промени.

## XML

## Пример:

```
<file name="data.txt" size="100" type="text"></file>
```

Атрибутите са полезни, когато трябва да се предостави допълнителна информация на програмата, която ще обработва XML документа.

## Пример:

```
<?xml version="1.0" encoding="windows-1251"?>
<students_list>
<student id="1">
<name>Иван</name>
<egn>8201273840</egn>
<address>Младост 3</address>
<course>Информационни системи и технологии</course>
</student>
<student id="2">
<name>Петър</name>
<egn>820143840</egn>
<address>Младост 1</address>
<course>Информационни системи и технологии</course>
</student>
.....
.....
</students_list>
```

## XML

## Метаданни (данни за данните)

## Данни

доц. Стоянова

25

## XML

## Пространства от имена

Тъй като в XML значението на таговете не е предварително дефинирано, понякога може да се получи неяснота в какъв контекст да се разбира името на даден таг.

доц. Стоянова

26

## XML

Това може да е проблем, ако:

- два елемента с еднакво име, но различен смисъл, трябва да се използват в един XML документ;
- два елемента с еднакво име, но различен смисъл, трябва да се използват в два XML документа, които ще се използват заедно.

доц. Стоянова

27

## XML

Пример:

XML документ съдържащ HTML информация за таблица:

```
<table>
  <tr>
    <td>Apples</td>
    ...<td>Bananas</td>
  </tr>
</table>
```

доц. Стоянова

28

## XML

и в същото време съдържа описание на маса:

```
<table>
  <name>Coffee Table</name>
  <width>85</width>
  <length>110</length>
</table>
```

Елементите <table> - различно съдържание и значение, но еднакви имена!

доц. Стоянова

29

## XML

Конфликтите на имената на елементите - използване на представка (префикс).

Пример:

```
<h:table>
  <h:tr>
    <h:td>Apples</h:td>
    <h:td>Bananas</h:td>
  </h:tr>
</h:table>
<f:table>
  <f:name>Coffee Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>
```

доц. Стоянова

30

## XML

### Използването на представки – дефиниране на пространства от имена!

Пространствата от имена се дефинират чрез `xmlns` атрибута в началния таг на елемента.

Синтаксисът е `xmlns:prefix="URI"`.

Uniform Resource Identifier.

Най-използвания URI е URL - Uniform Resource Locator. URL идентифицира домейна.

доц. Стоянова

31

## XML

Когато `xmlns` атрибутът е сложен в началото->>

**всички наследници с еднакви представки са от същия Namespace.**

Адресът, който е в примера, не се използва от браузера, а служи за уникално име на Namespace.

доц. Стоянова

32

## XML

```
<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>Coffee Table</f:name>
<f:width>80</f:width>
<f:length>120</f:length>
</f:table>
</root>
```

доц. Стоянова

33

## XML

Атрибутът `xmlns` на `<table>` таговете дава `h:` и `f:` представки като квалифицирани пространства от имена.

Когато пространство от имена е дефинирано за един елемент, то **всички наследници на елемента със същия префикс** се асоциират със същото пространство от имена.

доц. Стоянова

34

## XML

Пространствата от имена могат да се декларират в елементите, където те се използват или в XML root елемента:

Пример:

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

доц. Стоянова

35

## XML

URI в пространствата от имена не се използват от парсера за търсене на информация.

Целта - уникални имена на пространствата от имена.

Стриктно и пълно описание - на сайта на w3c.

доц. Стоянова

36

## XML

### Дефиниране на документи

- Един документ е добре дефиниран, ако спазва гореописаните правила.
- Програмите, които четат XML документи, четат документи, които са добре дефинирани.
- При първото несъответствие със стандарта, четенето на документа приключва и не може да бъде довършено.

доц. Стоянова

37

## XML

- Програмите, които четат XML документи са изключително малки в сравнение с тези, които разпознават HTML, защото ....
- За да се нарече даден документ "XML документ", той задължително трябва да е добре дефиниран.

доц. Стоянова

38

## XML

### Пример:

```
<code>
int matchtwo(a,b){
    if (a < b && a < 0){
        return 1;
    } else {
        return 0;
    }
}
</code>
```

доц. Стоянова

39

## XML

### Решение: чрез добавяне на Escape символи;

Вместо '<' - '&lt;'; (за знак по-малко и не начало на таг).

Символът '&' обозначава началото на такава последователност!

'&lt;'; за '<'

'&gt;'; за '>'

'&apos; за апостроф

'&quot; за знака за кавички

'&amp;'; за '&'

Символите '<' и '&' задължително трябва да се заместят с техните escape последователности, докато за другите три това може и да не се прави.

доц. Стоянова

40

## XML

За правилно интерпретиране на текста, третият ред трябва да изглежда така:

```
if (a &lt; b &amp;&amp; a &lt; 0){
```

'&lt;'; за '<'  
'&gt;'; за '>'  
'&apos; за апостроф  
'&quot; за знака за кавички  
'&amp;'; за '&'

доц. Стоянова

41

## XML

Или със следния таг - CDATA.

```
<code>
<![CDATA[int matchwo(a,b){
if (a < b && a < 0){
    return 1;
} else {
    return 0;
}
}]]>
</code>
```

Няма нужда от затварящ таг!

доц. Стоянова

42

## XML

След срещането на последователността '<!CDATA[' всички следващи символи се интерпретират като обикновен текст, а не като XML, до първото срещане на ']]>':

CDATA тагът може да съдържа всякаква последователност от символи, с изключение на ']]>':

доц. Стоянова

43

## XML

### Коментари

Коментарите в XML документа също не се интерпретират като XML.

```
<!-- <tag not parsed> </no errors> -->
```

Коментарите могат да съдържат различни последователности от символи, с изключение на '-->':

доц. Стоянова

44

## XML

За разлика от ескапе символите и CDATA-данните, коментарът не е част от XML документа.

Пример:

```
<students_list>
  <!-- All the students should have excellent marks -->
</students_list >
```

доц. Стоянова

45

## XML

### Инструкции за интерпретация

В XML документите могат да се пишат специални тагове, които съдържат информация (инструкции) за правилното интерпретиране на данните.

Тези тагове се наричат *Processing Instructions (PI)*.

Те не са част от XML текста.

За разлика от коментарите, PI съдържат информация, полезна за програмата, която ще чете и интерпретира данните.

доц. Стоянова

46

## XML

Форматът им е следният - `<?target PIdata?>` - на мястото на `target` стои името на програмата, за която е предназначена информацията, а `PIdata` е самата информация.

Пример:

```
<students_list>
<?TUstudents primary key = [attribute, id]?>
<?ScholarshipList identification:attribute=id?>
<student id="1">...</student>
<student id="2">...</student>
...
</students_list >
```

доц. Стоянова

47

## XML

Една специална PI е "xml ...".

```
<?xml version="1.0" encoding="utf-8"
standalone="yes"?>
<?xml version="1.0" encoding="windows-1251"?>
```

Тази декларация трябва да присъства в началото на всеки XML документ.

В нея атрибутът `version` е задължителен

доц. Стоянова

48



## XML

### Използване на XML документ

#### XML парсъри

- проверяват дали е добре дефиниран;
- извличат информацията от него.

Съществуват множество XML парсъри. Някои от тях са с отворен код.

## XML

### Интернет браузъри

- o отваря
- o визуализира според структурата им

## XML

### Валиден документ

- ❖ добре дефиниран
- ❖ елементи с произволни имена, ясна структура.
- ❖ конкретна структура на документа, възможните атрибути, техните стойности и т.н.

## XML

### Двата най-известни вида документи са:

**Document Type Definition (DTD);**  
**XML Schema Definition (XSD).**

Валидирането на XML документи улеснява четенето им не само от хора, а и от програми.

## XML

### Document Type Definition (DTD)

#### Пример:

```
<?xml version="1.0" encoding="windows-1251"?>
<!DOCTYPE note [
  <!ELEMENT note (to,from,heading,body)>
  <!ELEMENT to (#CDATA)>
  <!ELEMENT from (#CDATA)>
  <!ELEMENT heading (#CDATA)>
  <!ELEMENT body (#CDATA)>
]>
<note>
  <to>Иван</to>
  <from>Петър</from>
  <heading>Поздравление</heading>
  <body>Честит имен ден</body>
</note>
```

## XML

### XML Schema Definition (XSD)

```
<?xml version="1.0" encoding="windows-1251"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="note">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="to" type="xs:string" />
        <xs:element name="body" type="xs:string" />
        <xs:element name="from" type="xs:string" />
        <xs:element name="heading" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

## XML

### Приложения на XML

**XSL** - описва структурата на данните, а не как те изглеждат.

е **XSL (XML Stylesheet Language)** - е език, който се състои  
всъщност от два независими езика:

- XSL Transformations (XSLT);
- XSL Formatting Objects (XSL-FO).

**XSLT** служи за преобразуване от един XML в друг документ с произволна структура (например HTML).

## XML

### SOAP

XML е полезен с универсалността си и с това, че чрез него различни компютърни системи могат да споделят данни.

**Simple Object Access Protocol (SOAP)** е XML базиран протокол, който предоставя достъп до отдалечени обекти (и техните методи), Web services и сървъри.

## XML

На основата на XML са например:

- Chemical Markup Language – език за описание на химически съединения,
- Mathematical Markup Language (MathML) – за описание и вграждане на математически символи и изрази в HTML,
- RSS – използва се при публикуване на данни, които ежедневно се допълват (статии и новини, обяви за публични събития и т.н.),
- Synchronized Multimedia Integration Language (SMIL) – език за мултимедийни презентации в Web пространството,
- Open Software Description (OSD) – формат за описание на софтуерни ъдждейти,
- Scalable Vector Graphics (SVG) – за описание на векторни изображения,
- MusicXML – за описание на ноти, петолиния, тактове, паузи и т.н.,
- Open Financial Exchange – за обмен на финансови отчети.

## XML

### Заклучение:

XML - по-често и широко използван.

Множество програми за работа с XML документи, а в базите данни има нов тип – xml или възможност за запис в xml формат.

[http://www.w3schools.com/w3c/w3c\\_xml.asp](http://www.w3schools.com/w3c/w3c_xml.asp) – пълна XML спецификация.

<http://www.w3schools.com/xml/> - описва синтаксиса на XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- xml schema: http://schemas.microsoft.com/office/docbook/2011-11-15/2836 -->
<Students>
  <ID>1</ID>
  <F_Name>Cristian</F_Name>
  <L_Name>Ivancu</L_Name>
  <F_NUM>121218451</F_NUM>
  <Date_Beg>2012-12-10T00:00:00</Date_Beg>
</Students>
<ID>2</ID>
<F_Name>Mircea</F_Name>
<L_Name>Marian</L_Name>
<F_NUM>22134856</F_NUM>
<Date_Beg>2013-03-10T00:00:00</Date_Beg>
</Students>
<ID>3</ID>
<F_Name>John</F_Name>
<L_Name>Chavez</L_Name>
<F_NUM>111144736</F_NUM>
<Date_Beg>2013-05-09T00:00:00</Date_Beg>
<Date_End>2013-01-10T00:00:00</Date_End>
</Students>
<ID>4</ID>
<F_Name>Klaus</F_Name>
<L_Name>Schneider</L_Name>
<F_NUM>76654413</F_NUM>
<Date_Beg>2013-12-10T00:00:00</Date_Beg>
</Students>
<ID>5</ID>
<F_Name>Mircea</F_Name>
<L_Name>Marian</L_Name>
<F_NUM>22135667</F_NUM>
<Date_Beg>2013-05-09T00:00:00</Date_Beg>
<Date_End>2013-01-10T00:00:00</Date_End>
</Students>
<ID>6</ID>
<F_Name>Mircea</F_Name>
<L_Name>Marian</L_Name>
<F_NUM>44221966</F_NUM>
<Date_Beg>2013-03-10T00:00:00</Date_Beg>
</Students>
</Students>
```

