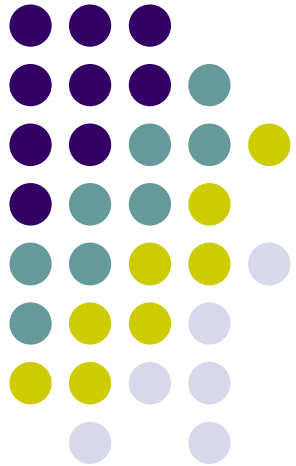
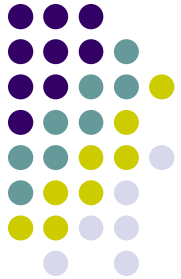
Мрежово програмиране

SGML, HTML и XML

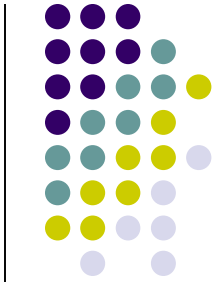




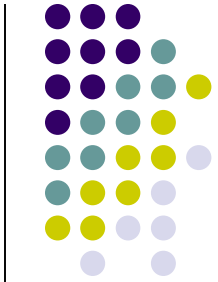
Стандартният обобщен маркиращ език (Structured Generalized Markup Language – SGML) представлява родоначалник на всички хипертекстови езици

- Езиците HTML и XML са получени от SGML, макар и по различен начин
  - XML е опростено подмножество на SGML
  - HTML първоначално е бил едно от SGML приложенията

SGML определя базовия синтаксис, но също така дава възможност да се създават собствени елементи



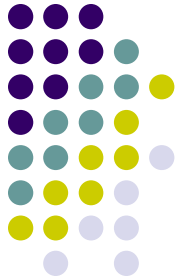
- За да се използва SGML за описание на определен документ, трябва да се обмислят съответстващия набор от елементи и структурата на документа.
- Например за да се опише една книга, трябва да се използват създадените от вас елементи с наименования BOOK, PART, CHAPTER, INTRODUCTION, A-SECTION, B-SECTION, C-SECTION и т.н.



Наборът от най-често използваните елементи за описание на документ от определен тип се нарича SGML приложение

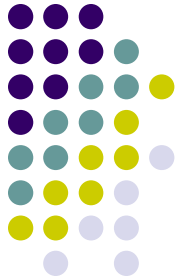
- SGML приложението включва в себе си правила, установяващи начините за организация на елементите, а също така и други особености на тяхното взаимодействие

Основни части на SGML документ



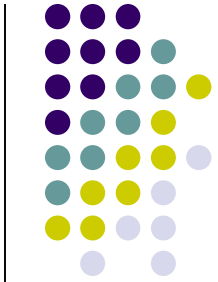
- SGML декларация, която определя какви символи и ограничители могат да се използват в приложението;
- Document Type Definition (DTD, определяне на типа на документа, схема) - дефинира синтаксиса на конструкциите за маркиране;
- Спецификацията на семантиката, свързана с препратките също така дава ограничения за синтаксиса, които не могат да бъдат изразени вътре в DTD;
- Съдържимото на SGML документа трябва да включва поне коренов елемент.

HTML притежава две основни характеристики:
- хипертекст, т.е. документът съдържа хипервръзки към други документи;
- универсалност, т.е. всеки HTML документ може да бъде разгледан във всеки уеб браузър в Интернет.



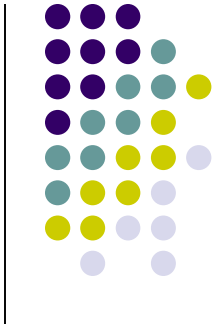
HTML е структурен език – форматирането се задава с поредица от инструкции, наричани тагове, които се изпълняват в момента на тяхното четене от браузъра, в резултат на което се изобразява и съответната информация на екрана.

HTML е описвал правилата, по които да се подготви информация за World Wide Web



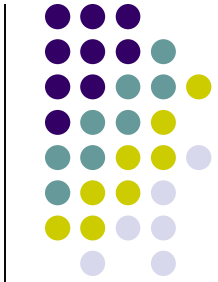
- HTML е набор от SGML правила, формулирани във вид за определяне на типа на документа (DTD), обясняващи, какво точно обозначават таговете и елементите.
- DTD схемата за HTML се съхранява в уеб браузъра.

```
<HTML>
<HEAD>
<TITLE>Home Page</TITLE>
</HEAD>
<BODY>
<H1>Home Page</H1>
<P><EM>Welcome to my Web site!</EM></P>
<H2>Web Site Contents</H2>
<P>Please choose one of the following topics:</P>
<UL>
 <LI><A Href="Writing.htm"><B>Writing</B></A></LI>
 <LI><A Href="Family.htm"><B>Family</B></A></LI>
 <LI><A Href="Photos.htm"><B>PhotoGallery</B></A></LI>
</UL>
<H2>Other Interesting Web Sites</H2>
<P>Click one of the following to explore another Web site:</P>
<UL>
  <LI>
   <A HREF=http://www.yahoo.com/>Yahoo Search Engine</A>
 </LI>
 <LI>
  <A HREF=http://www.amazon.com/>Amazon Bookstore</A>
 </LI>
  <LI>
   <A HREF=http://mspress.microsoft.com/>Microsoft Press</A>
  </LI>
</UL>
</BODY>
</HTML>
```

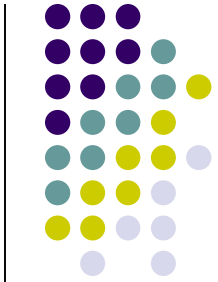



Home Page - Windows Internet Explorer
Google
Home Page
Home Page
Welcome to my Web site!
Web Site Contents
Please choose one of the following topics:
Writing
Family
Photo Gallery
Other Interesting Web Sites
Click one of the following to explore another Web site:
Yahoo Search Engine
Amazon Bookstore
Microsoft Press
100%

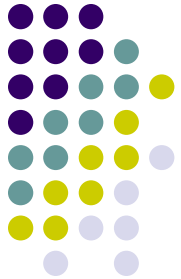
HTML елемент	Съставляваща на страницата
HTML	Цялата страница
HEAD	Информация за заглавието, например наименование на страницата
TITLE	Наименование на страницата, което се появява в реда заглавен прозорец на браузъра
BODY	Основен текст, изобразяван от браузера
H1	Заглавие от горно ниво
H2	Заглавие от второ ниво
P	Абзац на текста
UL	Маркиран списък (Unordered List)
LI	Отделен елемент в списъка (List Item)
IMG	Изображение
A	Връзка с друга страница или с друго място в дадената страница (елемент Anchor)
EM	Блок от текста в курсив (EMphasized)
B	Блок от текста с bold шрифт



Недостатъци на HTML

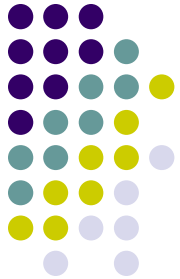


- HTML има фиксиран набор от тагове. Не е възможно създаването на собствени тагове.
- HTML е технология за представяне на данните. HTML не носи информация за значението на съдържанието, затворено в таговете.
- HTML е «плосък» език. Значимостта на таговете не е определена, затова няма как да се описват йерархични данни.
- В качеството на платформи за приложенията се използват браузъри. HTML не е достатъчно мощен за създаването на уеб приложения за съвременните уеб разработчици. Например не е възможно разработването на приложение за професионална обработка и търсене на документи.
- Големи обеми от трафик в мрежата. HTML документите, използвани като приложения, претоварват Интернет с големи обеми от трафик в системите клиент-сървър.

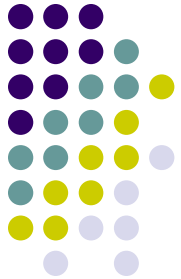


Примери за документи, които не могат да бъдат адекватно описани с помощта на езика HTML:

- Документ, който не съдържа типови компоненти (заглавия, абзаци, списъци, таблици и т.н.)
- База от данни (например списък с книги и тяхното описание)
- Документ, който трябва да се представи във вид на йерархична структура (книга, разбита на части, глави, раздели A, B, C и т.н.)

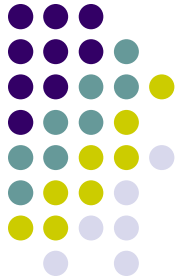


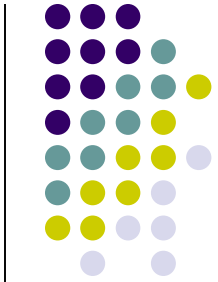
- За създаването на подмножеството на езика SGML, което да бъде прието от уеб обществото, е била организирана група от експерти по SGML, ръководена от Jon Bosak от компанията Sun Microsystems.
- Взето е решението да се премахнат множество от несъществените възможности на SGML. Полученият по този начин език получил името XML.
- Опростеният вариант се е оказал значително по-достъпен от оригинала. Неговата спецификация е заемала само 26 страници, докато спецификацията на SGML е повече от 500 страници.



- XML е препоръчан от *W3C* език (World Wide Web Consortium).
- XML е в текстов формат, предназначен за:
  - съхраняване на структурирани данни;
  - за обмен на информация между програми;
  - за създаване на специализирани маркиращи езици на негова основа.

```
<?xml version="1.0"?>
<INVENTORY>
 <BOOK>
 <HEADING>The Adventures of Huckleberry Finn</HEADING>
    <AUTHOR>Mark Twain</AUTHOR>
    <BINDING>mass market paperback</BINDING>
    <PAGES>298</PAGES>
    <PRICE>$5.49</PRICE>
 </BOOK>
 <BOOK>
  <HEADING>Moby-Dick</HEADING>
   <AUTHOR>Herman Melville</AUTHOR>
    <BINDING>trade paperback</BINDING>
   <PAGES>605</PAGES>
   <PRICE>$4.95</PRICE>
  </BOOK>
<BOOK>
   <HEADING>The Scarlet Letter</HEADING>
    <AUTHOR>Nathaniel Hawthorne</AUTHOR>
    <BINDING>trade paperback</BINDING>
    <PAGES>253</PAGES>
    <PRICE>$4.25</PRICE>
 </BOOK>
</INVENTORY>
```

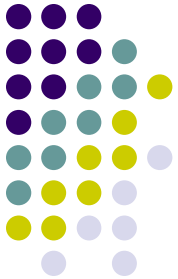




"Разширяемият тагов език Extensible Markup Language (XML) е съставна част на езика SGML… Той е предназначен за по-лесно използване на езика SGML в уеб и за изпълнение на задачи, които в настоящия момент се реализират с езика HTML. XML е разработен с цел усъвършенстване на използването и на взаимодействието на езиците SGML и HTML.“ (е записано в спецификацията на версия 1.0 XML)

- Защо е бил необходим друг нов език за уеб?
- Какви са неговите предимства и достойнства?
- Как той взаимодейства с HTML?
- Той ще замени ли HTML или само ще го усъвършенства?
- Какво представлява езикът SGML, част на който е XML, и защо не може да се използва за уеб страници SGML?

SGML позволява да се използва неправилен синтаксис



- Някои тагове не изискват затварящ таг, като <img>. Добавянето на затварящ таг води до грешка.
- Таговете могат да се слагат във всякакъв ред. Например:

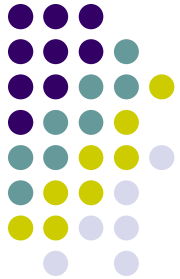
  Това е пробен низ,

  се третира като правилно разположение, въпреки, че разположението на таговете не е симетрично.
- Някои атрибути не изискват стойности, като nowrap в <td nowrap>.
- Атрибутите могат да бъдат дефинирани както с кавички, така и без. Това означава, че

  <img src=”picture.jpg”> и <img src=picture.jpg>

  се третират по един и същ начин.

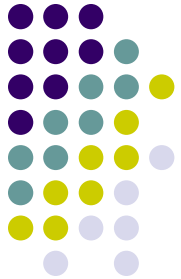
XML дефинира следните правила



- Всеки отварящ таг трябва задължително да има и затварящ.
- Ако за даден таг не е предвиден затварящ такъв, тогава в края на въпросния таг се добавя ”/”, като в <img />.
- Таговете трябва да се затварят в точният ред в който се отварят, като:

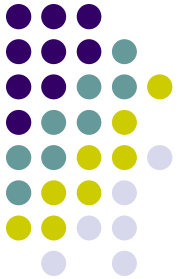
  Това е пробен низ.
- Всички атрибути задължително изискват стойности.
- Всички стойности на атрибути трябва задължително да са обградени в кавички.

XML има строго определен синтаксис



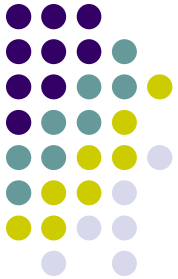
- Структурата на XML документите трябва да е разбираема за програмата, която обработва и изобразява информацията, съдържаща се в тези документи.
- Строгият синтаксис придава на XML документа предсказуема форма и подпомага написването на програми за обработка.
- Основното предназначение на езика XML е да се опрости работата с документите в уеб.

Начини за задаване на браузъра как да обработва и да изобразява всеки от създадените XML елементи

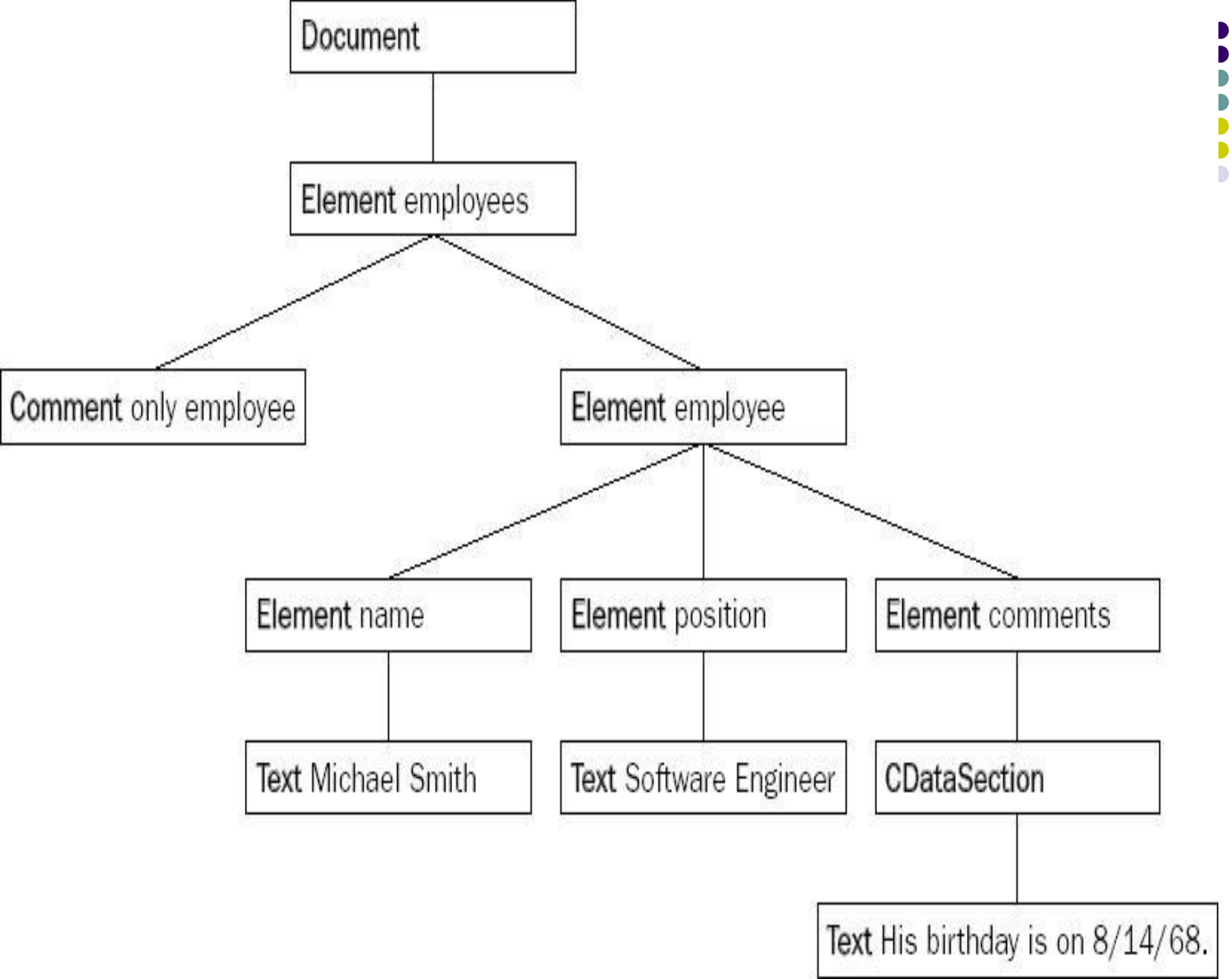


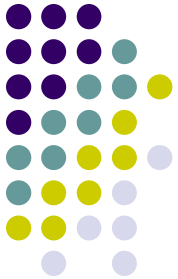
- Таблица със стилове
  - каскадна таблица със стилове (Cascading Style Sheet – CSS)
  - разширяема таблица във формата на езика на стиловите таблици (Extensible Stylesheet Language – XSL)
- Свързване на данните
  - Този метод изисква създаване на HTML страница, свързване към нея на XML документ и установяване на взаимодействието на стандартните HTML елементи на страницата
- Скриптове
  - Браузърът възприема XML документа като обектен модел на документа (Document Object Model – DOM), съставен от голям набор от обекти, свойства и команди. Написаният код позволява да се осъществява достъп, изобразяване и манипулиране с XML елементите

Този код може да бъде представен в DOM



```
<?xml version=”1.0”?>
    <employees>
      <!-- only employee -->
      <employee>
      <name>Michael Smith</name>
      <position>Software Engineer</position>
      <comments>
        <![CDATA[  His birthday is on 8/14/68.  ]]>
      </comments>
      </employee>
  </employees>
```


Document
Element employees
Comment only employee
Element employee
Element name
Element position
Element comments
Text Michael Smith
Text Software Engineer
CDataSection
Text His birthday is on 8/14/68.



Всеки XML документ започва с XML въведение (пролог), както е показано на първия ред в предишния код:

<?xml version=”1.0”?>

Този ред казва на различните програми, включително и на браузъра, че текущият документ трябва да бъде третиран като XML и че за него са валидни всички правила, дискутирани вече.
Следващият ред:

<books>

е елемент на документа (за елемент се счита съдържанието на всеки отварящ и затварящ таг).
Следващият ред:

<!-- begin list of books -->

е коментар и е едно от нещата, наследени от SGML.

<![CDATA[ ]]>

Това служи, за да се каже на програмата, която чете XML-ския файл, че текстът заключен в този таг не бива да се подлага на правилата на XML, а трябва да бъде третиран като обикновен текст.

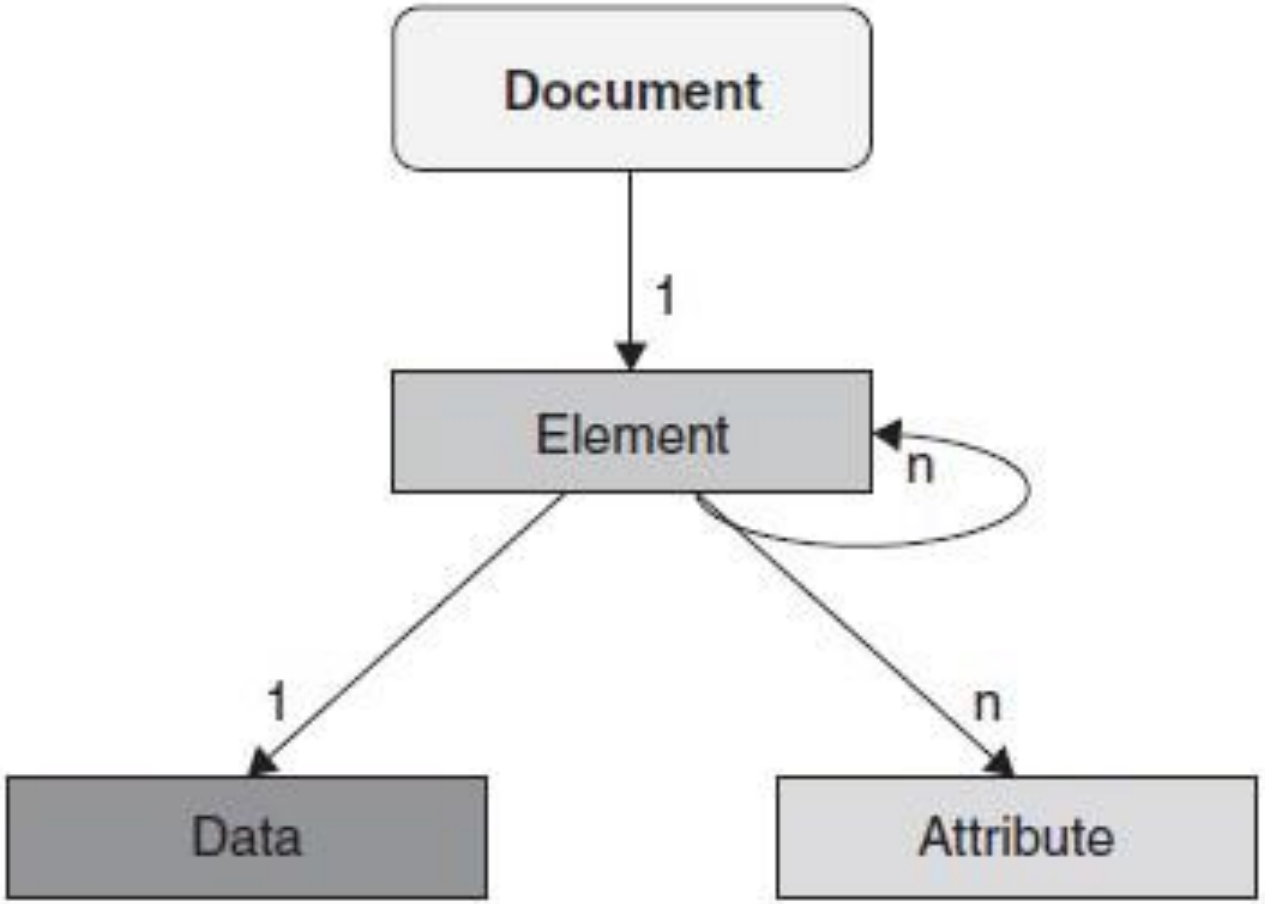
Ако например бяхме написали:

<description>If you’re a Java programmer working with XML, you probably already use some of the tools developed by the Apache Software Foundation. This book is a code-intensive guide to the Apache XML tools that are most relevant for Java developers, including Xerces, Xalan, FOP, Cocoon and Axis. <and Xindice>
</description>

То програмата щеше да определи, че <and Xindice> е таг и тъй като той няма затварящ, програмата щеше да прекъсне работа, казвайки, че този XML файл е невалиден. Затова се и използва

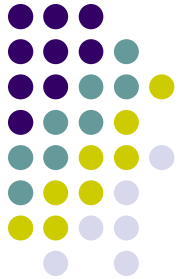
<![CDATA[ ]]> тага, за да се каже, че текстът там не трябва да се подлага на правилата на XML.

Структура на XML документ



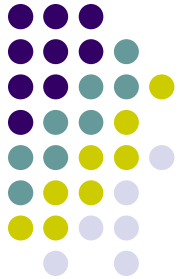
The general structure of an XML document

Йерархия от възли (нодове)



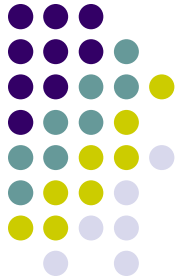
- Document – Най-високият нод в йерархията, към който се закачат всички други нодове.
- DocumentType – Обектното представяне на DTD използвайки <!DOCTYPE>, като в <!DOCTYPE HTML PUBLIC ”-//W3C//DTD HTML 4.0 Transitional//EN”>. Този нод няма дъщерни нодове.
- DocumentFragment – Използва се за временно съхраняване на нодове.
- Element – Представя съдържанието на отварящ и затварящ таг, като <tag></tag> или <tag />. Този нод е единственият, който може да съдържа атрибути или други дъщерни нодове.
- Text – Представя обикновен текст в XML документ, съдържащ се между отварящия и затварящия таг или в CData Секция. Този нод не може да има дъщерни нодове.
- CDataSection – обектното представяне на <![CDATA[ ]]>. Този нод може да има само текстови нодове, като дъщерни такива.
- Comment – представяне на XML коментар. Този нод няма дъщерни нодове.

Без да заменя HTML, XML се използва съвместно с него, като съществено разширява възможностите на уеб страниците за:



- виртуално представяне на документи от всякакъв тип;
- сортировки, филтрации, подреждане, търсене и манипулиране с информация с други методи;
- представяне на информацията в структуриран вид.

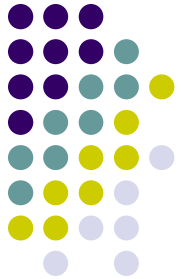
XML приложения, повишаващи качеството на XML документите



- Extensible Stylesheet Language (XSL) позволява създаването на мощни стилови таблици с използването на синтаксиса на XML
- XML Schema позволява разработването на подробни схеми за вашите XML документи с използването на стандартния XML синтаксис, което представлява по-мощна алтернатива на използването на DTD
- XML Linking Language (XLink) дава възможност за свързването на вашите XML документи. Той поддържа множествени целеви указатели и други полезни функции, осигуряващи голяма свобода в сравнение с механизма за организация на препратки в HTML
- XML Pointer Language (XPointer) позволява да се определят гъвкави целеви указатели

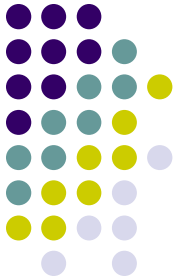
При съвместно използване на XPointer и XLink могат да се организират препратки към всяко едно място в целевия документ, а не само преходи към специално отделени пунктове

Реално използване на XML

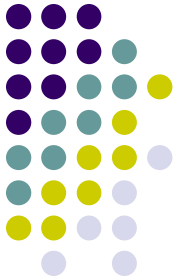


- Работа с бази данни
- Структуриране на документи
- Работа с векторна графика (VML – Vector Markup Language)
- Мултимедийни презентации (SMIL – Synchronized Multimedia Integration Language, HTML + TIME – HTML Timed Interactive Multimedia Extensions)
- Описание на канали (CDF – Channel Definition Format)
- Описание на програмни пакети и техните взаимни връзки. Такива описания осигуряват разпространението и обновлението на програмните продукти в Интернет (OSD – Open Software Description)

- Взаимодействие на приложенията чрез уеб с използването на XML съобщения (SOAP – Simple Object Access Protocol)
- Изпращане на електронни бизнес-карти по e-mail
- Изпращане на финансова информация (Quicken, Microsoft Money и OFX – Open Financial Exchange)
- Създаване, управление и използване на сложни цифрови форми за комерсиални Интернет транзакции (XFDL – Extensible Forms Description Language)
- Обмен на заявки за кандидатстване за работа и резюме (HRMML – Human Resource Management Markup Language)
- Форматиране на математически формули и на научна информация в уеб (MathML – Mathematical Markup Language)

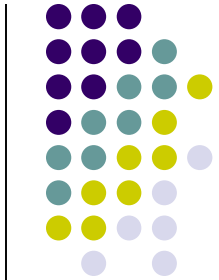


- Описание на молекулни структури (CML – Chemical Markup Language)
- Кодиране и изобразяване на информация за ДНК, РНК и вериги (BSML – Bioinformatic Sequence Markup Language)
- Кодиране на генеалогични данни (GeDML – Genealogical Data Markup Language)
- Обмен на астрономични данни (AML – Astronomical Markup Language)
- Създаване на музикални партитури (MusicML –Music Markup Language)
- Работа с гласови сценарии за доставка на информация по телефона. Гласовите сценарии могат да бъдат използвани например за генериране на гласови съобщения, справки за наличие на стоки и прогнози за времето (VoxML)

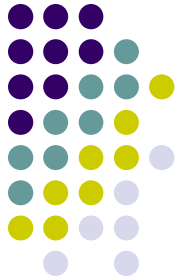


- Обработка и доставка на информация на курерски услуги. Federal Express например използва XML за тези цели
- Представяне на реклама в пресата в цифров формат (Ad Markup)
- Запълване на юридически документи и електронен обмен на юридическа информация (XCL – XML Court Interface)
- Кодиране на прогнозите за времето (OMF – Weather Observation Markup Format)
- Обмен на застрахователна информация
- Обмен на новини и информация с използване на отворени уеб стандарти (XMLNews)
- Предоставяне на религиозна информация (ThML – Theological Markup Language, LitML – Liturgical Markup Language)

Предимства на XML

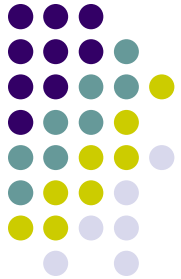


- Неговият формат на документа е разбираем както за компютъра, така и за потребителя;
- Поддържа Unicode;
- Във формата на XML могат да бъдат описани основните структури от данни, като записи, списъци и дървета;
- Това е самостоятелен документируем формат, който описва структурата и имената на полетата, а също и техните стойности;
- Притежава строго определен синтаксис и изисквания за анализ, което му позволява да бъде прост, ефективен и непротиворечив;

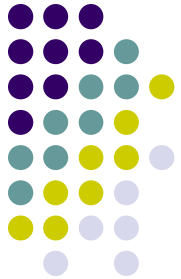


- Масово се използва за съхраняване и обработване на документи;
- Това е формат, основан на международни стандарти;
- Йерархичната структура на XML може да опише практически всякакъв тип документи;
- Представлява обикновен текст, свободен от лицензиране и някакви ограничения;
- Не зависи от платформата;
- Подмножество е на SGML, за който е натрупан огромен опит при разработването и създаването на специализирани приложения;

Недостатъци на XML



- Синтаксисът XML е с излишък.
  - Размерът на XML документа е съществено по-голям от бинарното представяне на същите данни (от порядъка на 10 пъти).
  - Размерът на XML документа е съществено по-голям от този, на документ в алтернативни текстови формати за предаване на данни (например JSON, YAML) и особено при форматите за данни, оптимизирани за конкретния случай на използване.
  - Излишъкът на XML може да повлияе на ефективността на приложението. Съответно расте и цената за съхраняване, обработка и предаване на данните.
  - За по-голямата част от задачите не е необходима цялата мощ на синтаксиса на XML и може да се използват значително по-прости и производителни решения.



- Пространството от XML имена е сложно за използване и за реализация на синтактичен анализатор.
- XML не съдържа вградена в езика поддръжка на типове от данни. Не съществуват понятията «цели числа», «низове», «дати», «булеви стойности» и т. н.
- Йерархичният модел от данни, предлаган от XML, е ограничен в сравнение с релационния модел и с обектно-ориентираните графове.