XML - въпроси и отговори
Първа част

Напоследък в новините за разработвани нови технологии и софтуер все по-често се появява абревиатурата XML. Сега ще свалим мистерията около трибуквеното съчетание и ще разберем какво всъщност е XML и как можете да го използвате.

1. Какво е XML?

XML е съкращение на Extensible Markup Language. Езикът е официално специфициран от World Wide Web Consortium (W3C) през февруари 1998 г. Спорно е дали XML представлява език или система от правила за създаване на нови езици. Казано на български, означава Разширяем език за маркиране. От самото му наименование е видно сходството с масово разпостранения HTML. Разликите между тях обаче е огромна.

Самата организация W3C предпочита да нарича XML "общ синтаксис за изразяване на структурата на информацията". И именно тук е голямата разлика с така познатия ви HTML. Той носи информация за това как трябва да се изобрази страницата. Кой текст е заглавие, кое изречение трябва да се подчертае, къде да се наблегне на отделни думи, къде да се изобрази списък с отделни елементи. Тази информация обаче е специфична за отделната страница и не носи никаква информация за съдържанието й. Браузърите, които имат в себе си информация как да тълкуват отделните тагове ще разчетат правилно кода ви, но той ще остане неразбран за многобройните други приложения, като WAP браузърите или програмите на различните PDA.

2. Каква е ползата от XML?

С все по- бурното равитие на комуникациите разработчиците на различни информационни канали се изправят пред необхдимостта да структорират информацията така, че тя да бъде достъпна за публикуване в множество формати. Така веднъж написан текстът би могъл да се публикува на хартия, в Интернет, да се използва от различен тип софтуер, да може да се конвертира в глас, да е достъпен за WAP телефони, дори за устройствата от ново поколение. При това без да е необходимо

да се променя самото съдържание. Всяка медия ще може да го интерпретира по своему и да извлича необходимата й информация.

Представете си, че като автори на web или друго съдържание имате свободата сами да измисляте таговете които използвате. Така че те да са максимално удобни и описателни за съответната информация. Добавянето на множество специфични тагове към един език само ще го направи по тежък за обработване и сложен за изучаване и приложение.

Отговор на тези изисквания дава именно XML. Неговата употреба може да се разглежда в два аспекта - за публикуване (предоставяне на информацията на потребителите в някакъв формат) или за обмен на данни между две машини. Във втория случай XML представлява идеално средство за подготовката на информацията Програмите, които обработват XML информацията се наричат парсери, а процесът - парсване. Спорно е дали в българския език има дума или израз, които да ги заменят. На първо време можем да използваме тези понятия.

3. Каква е връзката между XML и HTML?

Всъщност HTML и XML са братя. Те произлизат от Standard Generalized Markup Language (SGML), но докато HTML е негово приложение, XML е негово подразделение. Разликата е малко неясна на пръв поглед, но когато навлезете в особеностите на XML със сигурност ще я усетите. HTML, SGML и XML имат своето приложение и ще продължат да се използват там, където са най-подходящи.

4. Как развитието на XML подобрява възможностите за е-бизнеса?

Най-голямо е приложението на XML за момента в електронния бизнес. Способността му да представи информация в стандартен текстов формат и да осигури стабилна синхронзация между различни фирми, сайтове и приложения обясняват големия шум, който се вдига около него. Наскоро бяха преброени над 250 разновидности на XML, което говори не само за неговата популярност. Ако скоро не бъдат поставени основите на конкретни спецификации и стандарти, то XML няма да може да изпълни основното си предназначение- да бъде единна среда за описание на информацията.

Все повече фирми започват да използват XML разновидности за да осигурят реалното представяне на бизнеса си в Интернет. Една от най-разпостранените разновидности е Common Business Language (CBL), който се използва за описание на отделните продукти и услуги, бизнес ситемите и правилата. CBL произлиза от Electronic Data Interchange (EDI), но разширява многократно възможностите на бизнес технологията. Една от стандартизираните спецификации на CBL е Product Information Exchange (PIX) - отнасяща се до изготвянето на каталози.

Освен всичко останало, XML дава много по-добра възможност за сортиране и класифициране на данните от различни източници. Представете си търсеща машина, която трябва не просто да гадае за съдържанието на сайтовете, които посещава, разчитайки на мета таговете в тях. Ако има по - добри опорни точки при сортирането, търсачката ще даде и по-точни резултати на посетителите си. Например тя ще може с точност да ви каже в кои сайтове можете да си купите GSM, например. И ще пропусне тези, в които само се правят ревюта на отделни модели.

5. Как се пише XML?

Нека си представим, че пишем автобиография. Ако използваме HTML ще форматираме отделните и части така, че те да се изобразят в смислови цялости, но това ще касае само представянето в браузър. Всъщност в този случай ще укажем как желаем да изглежда докумнта ни в тези програми. И ако по някаква причина пожелаем да променим стила на показването или да премахнем част от информацията, това трябва да стане с промяна на самия документ. Много по-голяма свобода ще ни даде XML. Ако решим да го използваме в написването на автобиографията, то тогава бихме постигнали по-голяма гъвкавост.

А/ Уточняване на структурата

Първото, което трябва да направим е да определим йерархията на информацията в документа. Нека включим в него лични данни, обучение, опит и езиците, които владеем. Виждаме че се оформят 4 основни блока, които от своя страна ще съдържат по-малки късове от информация. Например в личните данни ще включим името си, което от своя страна ще се състои от три отделни къса информация - лично, бащино и фамилно.

Блокът с образованието също ще се раздели на по-малки смислови структури - учебно заведение, специалност, придобита степен, начало и край на обучението и т.н.

След като изясним за себе си всички нива на информацията можем да съставим схема на документа си. Разбира се това не е задължително, но поне в началото ще ви помогне много в писането на кода. Ето я структурата на примерната автобиография :

- Автобиография

-Лични данни

-Име

-Собствено

-Бащино

-Фамилно

-Дата на раждане

-Място на раждане

-Адрес

-Държава

-Окръг,област

-Град (име и пощенски код)

-Квартал, район (не е задължително да присъства)

-Улица

-Номер

-Етаж(не е задължително да присъства)

-Апартамент(не е задължително да присъства)

-Телефонен номер (код и номер)

-E-mail

-Личен сайт

-Образование

-Учебно заведение

-Тип

-Име

-Град

-Специалност

-Придобита степен, звание

-Срок на обучението

-Начало

-Край

-Опит

-Работно място

-Име (име и място)

-Длъжност

-Задължения

- Срок

-Начало

-Край (не е задължително)

Както можете да забележите, информацията е структурирана в отделни смислови цялости, които от своя страна също може да се раздробят.Ако трябваше да прдствим данните в HTML формат най-вероятно щяхме да използваме форматране, което да подчертае отделните единици. Но най-вероятно щяхме да изобразим началните и крайните дати на обучението и работата (например) по един и същ начин, така че разлика между тях не би могла да се направи. В XML всяка част от информацията е уникална и стои на полагащото й се място.

Б/ Създаване на кода

Нека сега попълним този скелет и преобразуваме логическите единици в ЕЛЕМЕНТИ. Елементът е основната градивна единица на XML документа. Той би могъл да се състои от информация, от други елементи или да е комбинация между двете. Елементът винаги се ограничава След малко ще изясним това. Нека първо погледнем примерен вариант на автобиографията, написан на XML:

Иван

Георгиев

Петров

20.01.1970

София

България

София

Sofia

Лозенец

Христо Смирненски

1

11

9630886

СОУ

Вапцаров

София

Обща

Средно образование

1984

1988

СУ

Климент Охридски

София

Българска филология

Магистър

1988

1993

IDG - България

Редактор

Работа в списание PC World

1999

(Копие на този код можете да откриете в приложение към списаните диск или в сайта на PCWorld – http://www.pcworld.bg/network/cv.xml.)

Сигурно ви направи впечатление, че използвах кирилица при въвеждането на данните. Повечето български самоучители, посветени на XML заобикалят този въпрос и оставят погрешното впечатление, че може да се използва само латиница. Важното е да укажете точно енкодинга, след това грижата за правилната обработка остава на съответните приложения. Все пак остава добра идея да пишете на латиница в таговете.

B/ Основни положения

Имайки предвид примерния код можем да покажем основните правила на XML синтаксиса. Те са абсолютно заължителни и най-малкото несъобразяване с тях ще доведе до грешка в обработката на целия код!

1. Всеки документ трябва да започва с указване на типа му : .

2. Всеки документ трябва да има един основен "коренен" (root) елемент, който включва всички останали (в случая - "cv").

3. На всеки отварящ таг трябва да съответства затварящ!

4. Таговете не могат да се кръстосват. Не можеше да отворя елемента за месторабота преди да съм затворил този за образованието.

5. Таговете не могат да започват с цифри или низ "xml" и не могат да съдържат интервали і двуеточия (освен ако не използваме namespaces).

6 Прави се разлика между малки и големи букви. Елементите cv и Cv са различни. Отварящият таг не съответства на затварящ .

Г/ Опасности

Най-често срещаните грешки се пораждат от липса на затварящи тагове или от проблеми с малки и големи букви.HTML разглези повечето разработчици, понеже позволява тези две правила да се нарушават. Голяма част от кода на браузърите служи за откриване и отстраняване на грешките на авторите на web страници. В XML нарушаването на тези изисквания е фатално.

Празните пространства имат значение и не се игнорират както в HTML. Имайте това предвид, когато въвеждате информацията. Все пак пространствата между различните тагове не са фатални и можете спокойно да ги използвате.

Д/ Елементи и атрибути

Ако разгледаме внимателно кода ще открием, че коренният елемент на документа е "cv". В него са включени други елементи, като например "personal_data". От своя страна той също е съставен от други елементи. Тоест, елементът "personal_data" е дъщерен на коренния, но родителски за "ime", например.

Ако обърнем внимание на тага ще установим, че той всъщност затваря сам себе си, посредством наклонената черта преди дясната скоба. Това са така наречените "празни тагове", които не съдържат информация в себе си. Данните там са представени в така наречените "атрибути". Атрибутите са начин да включите добавъчни данни към елемента, без да е необходимо да създавате нов. Важно е да не забравяте, че празните елементи трябва да се отбелязват със самозатварящи се тагове.

Можете да използвате атрибути за всички елементи, както направихме в примера с 1. Така указахме, че става дума за отделен вход на сградата. Не пропускайте да оградите стойностите на атрибутите с кавички, без значение дали са единични или двойни. Все пак затварящите кавички трябва да са като отварящите.

Същинското съдържание на елемента се нарича PCDATA (Parsed Character Data). То се

обработва от парсерите в съответствие с маркерите, които го заобикалят. Друга част от съдържанието на документа - тази, която не е част маркирането, но не е и в прякото съдържание на елемнта се нарича CDATA (Character Data). Тук влизат атрибутите, отправките към външни файлове и т.н.

Има някои символи, които не можете да използвате в съдържанието на документа. Като например & и ъглови скоби - . Трябва да ги заменяте съответно с & < и >. Например:

Това е < текст &amp още нещо &gt тук.

Специален метод за допълване на силата на елементите са така наречените "пространства от имена" - Namespaces. С тяхна помощ се избягва употрбата на едни и същи елементи с различни цели в един или няколко документа. Но тяхната концепция е прекалено обширна за да се вмести в настоящия текст.

6 .Какво означава добре оформен и валиден документ?

Както вече стана дума, синтактичните изисквания на XML са строги и не търпят изключения. Въведен е специален термин за документите, които отговарят на всички правила и са абсолютно правилно написани. Те се наричат добре оформени (well-formed) документи. Тяхната обработка преминава гладко и не би трябвало да създаде грешки. Винаги е полезно да проверите дали документите ви са добре формирани, преди да ги включите в употреба.

Възможноста всеки сам да създава таговете, които използва, би могла да доведе до абсолютен хаос, в който всеки сайт или програма говорят на някакъв свой си език и не могат да се разберат едни с други. За това, когато се създава един XML документ, използваните в него елементи следва да се опишат. Така обработката на документите става по-ясна. Описанието на елементите се извършва в така наречената Document Type Definition (DTD). То би могло да се съдържа във външен файл, но може и да е в началната част на XML документа. Документи, които отговарят точно на дефинициите си се наричат валидни (valid).

Тези определения маркират двете нива на адекватност, която би трябвало да има един документ. Не винагио е необходимо той да бъде валиден. В много случаи парсерът не се интересува от това какво е съдържанието. По тази линия различаваме два типа парсери - валидиращи и невалидиращи. Ако използвате Internet Explorer за да изобразите XML съдържание, то в този случай няма да се направи проверка дали елементите в съдържанието съответстват на дефинираните в DTD.

Най-важното и задължително условие за един XML документ е той да е добре оформен.

В следващата част на “XML - въпроси и отговори” ще обясним какво е и как се използва DTD, как различните браузъри третират кода, ще се запознаем подробно с популярния формат за синдикация RSS и други. На диска към списанието ще откриете примерния код на автобиографията, както и този текст във XML формат.