

Поняття Web-технології

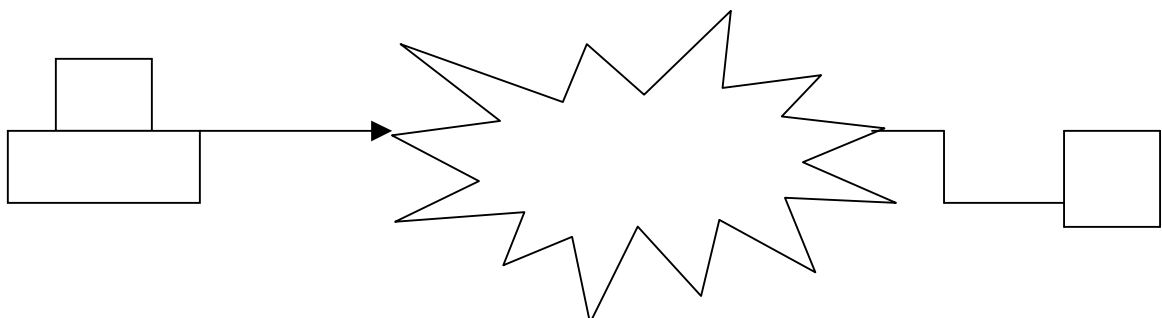
1. Основи Web-технології
2. Принципи взаємодії броузера та Web-сервера
3. Кодування тексту
 - 3.1 ASCII – кодування
 - 3.2 Однобайтове кодування
 - А) KOI – 8
 - Б) ISO – 8859-5
 - В) CP – 1251
 - Г) IBM - 866
 - Д) MAC – сур
 - 3.3 Двобайтове кодування
 - А) ISO – 10646
 - Б) VTF – 8
 - В) Vricode

1. Основи Web-технології

Web-технологія включає в себе взаємодію Основи Web – клієнта із Web-серверами по протоколу HTTP. Web – сервери містять на собі Web- content – будь яка інформація, що може бути прочитана із серверів – HTML і XML сторінки динамічно генеровані, сторінки при допомозі мов динамічного програмування (ASP, PHP, Gava) сторінки розділені на фрейми, розроблені при допомозі мов динамічного HTML Gava.

Web – клієнти мають підтримувати усі сучасні протоколи відображення Web- contenty.

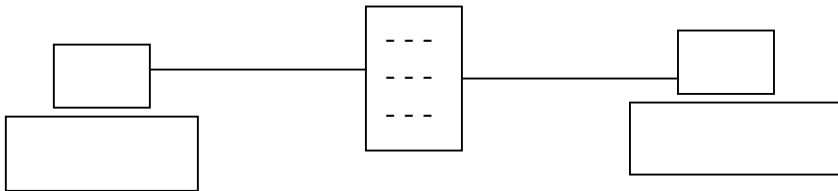
2 Принципи взаємодії броузера та Web-сервера



7 Прикладний рівень
6 Представницький
5 Сеансовий рівень
4 Транспортний рів
3 Мережений рівень
2 Канальний рівень
1 Фізичний рівень

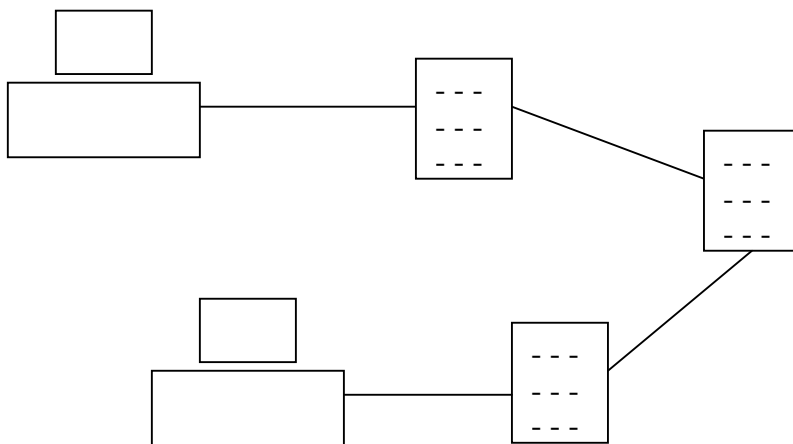
Модель взаємодії двох відкритих систем

1) Фізичний рівень



2) Канальний рівень поділений на байти тобто 0:1

3) Вони можуть взаємодіяти тільки на мережевому рівні



4) Взаємодія двох комп'ютерів



5) Сенсовий

6) За представлення інформації і шифрування інформації.

7) Протокол передачі інформації

3. Кодування тексту

Кодування або кодова таблиця – це однозначна відповідність між підмножиною цих чисел і деяких наборів символів. Символ може бути буквою може відповідати звуку – мови, може бути приставлений графічно, знаком, а може і не мати ніякого зображеного опису. Тобто, символ – це найменша не ділена частина інформації.

3.1 ASCII

Важливе місце в кодуванні символів займає кодування ASCII. Це кодування є 8-ми бітним, тобто займає один байт, тобто охоплює 256 символів тобто може описуватись 7-ма бітами. Найстарший біт 8 завжди дорівнює 0. Важливість цього кодування полягає в тому, що він включає в собі найважливіші символи які є універсальними для комп'ютерів будь-якої країни, або територіальної зони. Він включає латинський алфавіт, цифри, основні знаки пунктуації і основні символи позначення. Будь-яке кодування завжди сумісне з ASCII, тобто на своїх перших 128-ми місцях розміщують ті ж самі символи в тому ж порядку. Перші 32 позиції в кодуванні ASCII називаються керуючими символами і призначаються не для передачі текстової інформації, а для керування пристроєм що отримує по лінії зв'язку текстовий файл перевід каретки, перевід строки, табуляція, перевірка речення.

3.2 Однобайтні кодування

Якщо задіяти в однобайтне кодування старший біт то ми отримаємо додаткові ще 128 біт. Яких повинно хватити для кириличного алфавіту і набору

деяких спеціальних символів. Одним із перших стандартів кодування російських букв на комп'ютерах був KOI-8. Воно примінялося в доісторичні Радянські часи. Всередині 80-х з'явилися перші русифіковані системи UNIX вони унаслідували це кодування від своїх пращурів. Поки у 86-х роках не з'явилося славнозвісна компанія Microsoft з своєю ОС MS-DOS і ПК. Також крім Microsoft перша фірма яка випустила ОС була фірма Apple. І само собою „Макінтоші”. Організація ISO випустила ISO 88-59-5. ISO 88595 символи йдуть підряд.

Хоча в нашій країні комп'ютери цієї марки так і не набули популярності, порівняно з їх популярністю на заході.

Міжнародна організація по стандартизації (ISO) спробувала навести порядок у 8-ми бітних кодуваннях, створивши серію ISO 88-59, що розширюють таблицю ASCII.

Якщо кодування ISO 8859-5 для кирилиці так і не прижилась, перша із цієї серії – кодування ISO 8859-1 відома під назвою Latin-1, - змогла стати загально прийнятим стандартом для кодування „розширеної” латинської. В це кодування включені майже всі символи, які використовуються на письмі західноєвропейських мов – французького, німецького, іспанського.

ISO 8859-1 - латинське

ISO 8859-5 - кирилиця

ISO 8859-6 - арабське

ISO 8859-7 - грецьке

Але нажаль кириличне кодування ISO не набуло значного поширення, хоча кодування ISO 8859-1 (латинське) стало стандартом ДЕФАК.

По аналогії з ASCII перші 32 позиції в другій половині кодування в серії ISO 8859 оголошені „невикористовуючими”.

3.3 Двохбайтове кодування.

Мови з ієрархічною писемністю користуються змішаними кодуваннями в яких є ієрогліфи, а їх є десятки тисяч, представлені двохбайтовими кодами, а вставки на латиниці кодуються по однобайтовій таблиці. Переключення між

двохбайтовими і однобайтовими режимами робиться спеціально зарезервованими керуючими символами.

В 1991 році була спроба створити єдине двохбайтове універсальне кодування, що охоплювало всі алфавіти і ієрогліфи. Результатом став стандарт під назвою UNICODE, що охоплював не тільки всіх живих і більшості мертвих мов світу, але й більшість музикальних, математичних, хімічних символів. Хоча масове використання в документах UNICODE і програмами залишається справою майбутнього. Для Web-дизайнера воно має велике значення, адже воно оголошене стандартом документа в HTML починаючи з 4-ї версії.

ISO вже представила стандарт 4-байтове сумісне з UNICODE кодування під назвою ISO 10646. В ближній час починає відігравати все більшу роль ISO 10646 під назвою UTF 8.

Тема №2 Основи стандарту HTML та ієрархічні CSS специфікації.

1. SGML стандарти
2. Історії та версії HTML.
3. Синтаксис та текстова розмітка.
4. Посилання та прив'язки.
5. Форми.
6. Зображення та об'єкти.
7. Таблиці.
8. Фрейми.
9. Національні особливості побудови Web-сайтів.
10. Середовище і доступність.
11. Мета дані та пошук.
12. Основні принципи CSS.
13. Можливості CSS.
14. Модульний HTML.

1. В 1986 році набула статусу міжнародного стандарту HTML, яка стала носити назву SGML.(Standart Gimeralize Marca Language). Ця узагальнена метамова призначена для побудови систем логічної і структурної розмітки будь-яких різновидності текстів. Структурна розмітка означає, що керуючі коди, що вносяться в текст не несуть ніякої інформації про кодування елемента, а лише вказують границі і підпорядкування основних його частин, тобто задають його структуру. Засновники SGML намагалися повністю абстрагуватися від проблем представлення тексту в різних програмах, на різних комп'ютерних платформах. Хоча формально ніщо не мішає записати засобами SGML любую інформацію про елементи документа – в тому числі і параметри його форматування, - ідеологія цього языка потребує обмежитися вказуванням на рівень заголовка і його місце в ієрархічній структурі документа. Всі решта мають бути винесені в так називаючі стильові специфікації – абсолютно ідеальний і, як прийнято казати, ортогональний по відношенню до структурної основи інформаційній „слой”. Завдяки цим обмеженням виділений текст зможе без ускладнення інтерпретувати любую програму, яка працює з будь-яким думачим засобом виводу. Сам по собі SGML є не готова система розмітки тексту, а лише зручна метамова, яка дозволяє будувати такі системи для конкретних обставин. Стандарт SGML встановлює лише синтаксис запису елементів розмітки, а також правила вибору нових елементів і вказування структурних відносин між ними. Для практичної розмітки документів треба додатка SGML – набір вибраних елементів, які представляють собою, по суті, формальне описання структури документа.

2. Сама система SGML не отримала дуже великого поширення до тих пір поки 1991 році співробітники європейського інституту фізики, що займалися створенням системи передачі інформації через Інтернет не обрала SGML в якості основи для нової мови розмітки гіпертексту в документі. Ця мова найвідоміша з додатків SGML була названа HTML. На початку HTML як SGML додаток дотримувався всіх особливостей структурної розмітки тексту. Із 40 видів HTML версії 1.2. (1993) лише 3 мали функції фізичних параметрів

представлення документу, але не були рекомендовані до використання. Вся розмітка була чисто логічною і лише в описовій частині стандарту, що супроводжувала формальне виведення тегів було описано частково виведення шрифтів у графічних броузерах. Першим графічним броузером була програма Mozaik, що була розроблена як і сам WWW в національному центрі комп'ютерних приладних програм США. В цей золотий вік не існувало ніяких протиріч між стандартами і HTML розвивався в рамках структурної розмітки. Першо-початковий варіант стандарту включав в себе багато цінних нововведень: теги для створення таблиць, розмітки для математичних формул, вставки обрамлених текстових малюнків, примітками та іншим. Але саме головне, третя версія була спробою вирішити протиріччя між ідеологією структурної розмітки і потребами користувачів, зацікавлених у гнучких можливостях візуального представлення. Протиріччя було вирішене додаванням до стандарту HTML ієрархічних стильових специфікацій. Останній стандарт HTML версія 4.0 в грудні 1997. В тексті стандарту чітко проведено розділення логічних і візуальних тегів. Останні є не рекомендовані.

3. Вивчення будь-якої комп'ютерної мови починається з знайомства з його основними будівельними блоками – операторами, висловами, змінами. Мова HTML є досить простою і крім звичайного тексту має лише один тип керуючих інструкцій, які називають тегами. Необхідно розуміти різницю між тегами (одинацями розмітки) і елементами (складові частини документа). Теги розділяють вихідний неформатований текст документа на елементи і по-друге створюють нові елементи, яким нічого не відповідало в тексті. Наприклад графічні вставки Java applet. Теги є двох видів: парні і непарні. Парні охоплюють фрагмент тексту або інші теги. Парні теги повинні вкладатися одні в одні без пересікання, тобто якщо в області дії тега А відкрився тег В, то він повинен закритися до того, як закриється тег А. Більшість тегів, як парні так і непарні мають атрибути, що змінюють і уточнюють дію тега, а атрибути відповідно можуть приймати відповідні значення. Регістр букв не відіграє значення для позначення тегів і назв атрибутів, але мають впливати на значення

атрибутів. Існують атрибути, що не потребують присвоєння значення, бо сам факт присутності атрибутів включає відповідний режим роботи даного тегу. Згідно стандарта ланки навколо значення атрибута обов'язкові в тих випадках, коли це значення містить символи крім букв, цифр, точки і дефіса. Щоб ввести в документ символи, які відсутні на клавіатурі або мають син таксі HTML спеціальне значення приміняють підстановки. Підстановки є: 1. мнемонічні (<-<) ; 2. числові.

Набір мнемонічних кодів визначених в стандарті HTML включає в себе всі символи кодування LATIN 1, в тому числі і пробіл, а починаючи з версії HTML 4.0 включає символи UNICODE. В числових підстановках замість мнемонічного використовують десятковий числовий код з додаванням перед символом дієзу (# 160). Деякі Web-програмісти використовують методику використання сторінок переводячи свої тексти в мнемонічні або числові підстановки. Два обов'язкові теги, які мають бути в мінімальному HTML документі. <TYPE> </TYPE>

<'DOCTYPE HTML PUBLIC' - // W3C// DTD HTML 4.0 //EN>

При текстовій розмітці стандартом рекомендується використання мінімум структурних систем. Даний підхід є виходом для тих хто не потребує якогось серйозного дизайну або не має коштів на нього.

4. Можливість зв'язувати документи павутиною взаємних зсилок є головною відмінністю інтернета від всіх інших засобів розповсюдження інформації, яке відображається в назві мови розмітки гіпертексту. Зсилки в HTML реалізує тег A, який сам по собі є досить простим і обмеженим по можливостях. Будь-яка зсилка має два обов'язкових елемента: джерело і пункт призначення. Джерело – це зображення або фрагмент тексту, який знаходиться в середині тега A, а пункт призначення – це URL адрес документа на який веде зсилка. Адрес призначення може вказувати не тільки на весь документ в цілому, але й на елемент цього документа. Для цього необхідно, щоб пункт призначення був помічений атрибутом name того ж тега A. В свою чергу в документі джерелі ця мітка приписується до адреса призначення через символ

дієз. Для документів, що генерує у відповідь на питання програми на сервері, прямо в адресі можна передавати параметри виводу; звичайно такі параметри, записані у вигляді змінна = значення, відрізняються від URL визиваючі програми символом.

5. Іншою типовою відмінністю HTML документів від паперових є форми або бланки призначені для зворотнього зв'язку, тобто від ??шини інформації до користувача на сервер. Форми включають: 1) поля для вводу тексту; 2) прапорці; 3) випадаючі списки; 4) списки з прокруткою 5) кнопки

Автори HTML можуть вибирати для своїх форм будь-які елементи, вимагаючи їх розмір і зміст. Власне тег форм об'єднує групу зв'язаних по змісту елементів і вказує адрес програми, що буде обробляти ці форми на сервері. HTML сторінка може міститися будь-яка кількість незалежно одна від одної форм, в кожній з яких має бути присутня „пускова кнопка”, яка відправляє дані на сервер. Ця кнопка обов'язково має бути стандартною інтерфейсною кнопкою. Її роль має бути зображення. Кнопка створена при допомозі тега `input` з атрибутом `TYPE=SUBMIT`. Також для відправлення даних на сервер може використовуватись кнопка `Enter`.

6. Тег `img` призначений для вставки зображення і створює нові елементи документа, що відсутні в вихідному тексті. Цей тег посилається на зображення, що зберігається у файлі формату `gif` або `gr`. Цей графічний файл може розміщуватись в тому ж каталозі, що HTML файл сторінки або лежати в іншому каталозі або в іншому сервері. Більшість атрибутів цього тега керують форматуванням зображення, встановлюють розмір поля та інше. В останніх версіях HTML, досить імовірно, що буде прийнята попитка перейти на використання ОБ'ЄКТ для ставки будь-яких зовнішніх по відношенню до документу об'єктів або даних, в тому числі зображення.

7. До другої версії HTML не було ніяких засобів для створення таблиць, якщо не враховувати фрагментів збереження всіх пробілів табуляцій і переносу стрічок. Тег `table` часто використовують для візуального форматування сторінки, а не для прямого представлення табличного матеріалу. Алгоритм верет

таблиць, якому необхідно враховувати багато суперечливих один одному повідомлень достатньо складний і погано заду коментований, і через це в деяких деталях несумісний в броузерах.

8. Фрейми – це можливість поділити вікно броузера на частини, загрузивши в точне з вікон окремих HTML файл. Це дає можливість не тільки гнучкого розвитку інтерфейсу, а також і те, що HTML файл дає можливість посилатись на інші, але URL може зовсім не співпадати с тим адресом, який відображений в URL броузера. Це досить цікаво, якщо просто вставити вміст одного файла в інший неможливо. Сайт із фрейтами часто критикують за незручність і нелогічність навігацій. Більш серйозними є доступність фреймів для графічного середовища і автоматичних накопичувачів інформації.

9. Стандартом кодуванням HTML документа починаючи з версії 4.0 оголошено кодування UNICODE, однак це не означає, що всі HTML документи повинні створюватися, зберігатися і передаватися в цьому документі. Стандарт допускає використання автором будь-яких інших кодувань. Однак броузер всередині себе розбираючи синтаксис документа повинен перекодувати його з зовнішньо заданого автором кодування в UNICODE. Для дизайнера ця вимога має один практичний наслідок – числові підстановки повинні зсилатися на UNICODE – коди потрібних символів, а не на їх коди в поточному кодуванні документа. Це зроблено для того, щоб не тільки мнемонічні, а і числові підстановки не залежали від кодування документа, а також давали можливість зсилатися на символи за межами поточного кодування. На жаль, наскільки багато версій броузерів до цієї пори не підозрюють про цю потребу стандарту, з відомою гарантією можна користуватися тільки числовими або мнемонічними кодами із репертуару Latin – 1.

10. Забезпечення доступу до сайту для будь-яких платформ і будь-яких середовищ (графічний, текстовий, звуковий) є важливим аспектом Web-дизайном. Не потрібно вважати, що загальноприйнятими графічними броузерами не може користуватися тільки малий % чудаків, інвалідів і власників текстових терміналів. Все більше стає користувачів, які з різних

причин надають перевагу прослуховуванню вмісту Web- сторінок використовують програму, яка читає текст вголос або власників переносних пристроїв для доступу до мережі з чорно-білим екраном. Також необхідно врахувати, що сторінку будуть читати не тільки люди, а й роботи і автоматичні накопичувачі інформації. Доступність інформації в Інтернет спирається на два принципи: 1) переваги використання структурної, а не тільки візуальної розмітки; 2) доступності – підтримка текстового еквіваленту для всієї інформації сторінки.

Але це не означає, що нам потрібно робити копії всіх HTML файлів з тим же текстом, але без графіків і таблиць. Для цього HTML дозволяє зберігати в тому ж файлі текстовий транс-крипт для будь-яких текстових елементів, насамперед всього зображення. Звичайний текст – самий універсальний комп'ютерний формат даних з самими широкими можливостями перетвореннями і відображеннями, і тому підтримка текстового плану веб – сторінки ліпший спосіб забезпечити її доступність для любых користувачів інформації.

11. Один з найменш вивчених аспектів Web – дизайну це необхідність врахування не тільки естетичного інформаційного представлення, а й особливості прийняття автоматичних збирачів інформації. Зараз до цього класу належать виключно роботи пошукових систем, що збирають текстові бази доступних в мережі документів і надаючи зсилки до них по ключовим словам. В майбутньому число роботів Інтернета буде збільшуватись і можливо появиться інтелектуальні екземпляри, сприйняття інформації до якого буде наближено до людського. Але на даний час Web – дизайнеру необхідно враховувати можливості примітивних текстових пошукувачів. Як наслідок дотримуватись нескладних правил в цій області може принести вигоду для потрібного сайту більше половини відвідувачів, якого попали на нього через пошукові системи. Для досягнення цієї мети необхідно використовувати тег meta в ключових сторінках сайта, яка має містити анотації і списки ключових слів. В тегу meta атрибути name, content.

12. Від версії системи CSS дуже сильно залежить, що з її допомогою можна добитися. Перша версія специфікація (CSS level 1 або просто CSS1), стала офіційним стандартом в кінці 1996 року, по суті, лиш пропонувала CSS – запис для тих параметрів форматування, котрі і без того вже, будь це законно або законно було доступним HTML- документам в тодішніх графічних броузерах. Властивості CSS включали в себе вибір шрифту, параметри форматування тексту, установку фонового коліру або зображення, ширину полів трохи другорядних параметрів та ін.

13. Не можна сказати, що доступна на сьогодні веб – дизайнерам технологія текстової розмітки – HTML з невеликою домішкою CSS – була зовсім залишена здібностей до поділу аспектів вмісту і представлення. Опит, вроджена акуратність і відповідальне відношення до матеріалу, з яким приходиться працювати, дозволяє окремим дизайнерам практикувати в HTML стиль, який відповідає потребам ідеології SGML. Багатьом дизайнерам зовсім не просто перебудуватись на „ортогональний стиль” розмітки. Якщо навіть примітивні „іменовані стилі” в текстових процесорах рахується прерогативою „професіональних користувачів ”, що говорить про більш послідовні системи ортогональної розмітки.

Тема. Стандарт мови розмітки XML

1. Синтаксис.
2. Рівні відповідності.
3. Конверсія.
4. Розширення мови посилання XML.
5. Розширена мова стильових специфікацій XSL.

1. Модульний підхід дозволяє досягнути в HTML деякої відповідності структури і представлення. Звичайно набагато зручніше зберігати блоки візуального кода, що повторюється в окремому зальному для всього Web – сайта стильовому файлі. При цьому документи розмічати лише посиланнями на той чи інший блок, тобто по суті тегами логічної розмітки, що говорить лише про те, що стоїть в даному місті документа, а не про те, як воно виглядає. Саме

таке природне розділення аспектів змісту і представлення надає мова XML. XML – це компактне спрощення під множини мов SGML, яке розроблено консорціумом W3 для розрахунку на поступове витіснення з Інтернету мови HTML. XML – освоюється провідними провідниками програм причому не лише броузерів. XML – з часом з'явиться в більшості текстових процесорів, баз даних, систем підготовки документації, а деякі дослідники вбачають в будову мови в ОС.

Зовні XML дуже схожі на HTML, він використовує ті самі кутові дужки, що відкривають і закривають теги, атрибути і підстановки. Але якщо в HTML всі допустимі теги жорстко задані стандартом, то XML документ може використовувати будь-які назви тегів, які придумуються на ходу автором. Це пояснюється різним статусом цих мов розробки Web – сайту.

HTML є одним із додатків SGML і його породженням, а XML підмножина SGML, яка має лише трохи менші можливості, але повністю придатна для створення фіксованих систем розмітки документів. Такі системи на основі XML досить бурхливо створюють в останній час від складної мови MAS ML для розмітки математичних текстів до простих наборів з декількох десятків тегів, для збирання кулінарних рецептів або текстів церковних проповідей.

Вся специфіка HTML як одного з додатків SGML виражається у формальній конструкції, яка називає визначення типу документа. В ідеалі DTD – це найвищий авторитет в усьому, що стосується синтаксиса тої чи іншої версії HTML.

Оскільки DTD для кожної версії HTML зафіксовано в офіційній специфікації мови, то в самому документі наводити не потрібно, однак будь-який HTML документ повинен посилатися на своє DTD за допомогою тега.

DTD зберігає таку інформацію:

- 1) повний список основних документів з вказуванням на обов'язок для кожного з них відкриваючих і закриваючих тегів.

- 2) повний список атрибутів для кожного елемента з інформацією про результати.
- 3) Ієрархічна структура документа у вигляді інформації про те, які інші елементи в якому порядку і в яких поєднаннях можуть зустрічатися в середині кожного елемента.

Наприклад в DTD для HTML 4.0 вказано, що в елементі HTML можна опускати як відкриваючі так і закриваючі теги. Елемент OL повинен мати відкриваючі і закриваючі теги, а вміст його повинен складатися з одного або декількох стоячих один за одним елементів DTD. Всі елементи XML повинні мати як відкриваючі так і закриваючі теги.

2. Якщо SGML кожен документ повинен мати своє DTD, в HTML є одне DTD на всіх, то XML представляє собою компроміс. В першому випадку документ може мати і посилатися на DTD а може обходитись без нього. В останньому випадку кожен новий тег і атрибут позначається самим фактом свого вжитку.

Таким чином для XML документів існує два рівні відповідності стандарту:

- 1) Web-formed
- 2) Wolid

Перший рівень – це документи, що не мають DTD, але задовольняють всім іншим вимогам.

Другий рівень – це документи, що мають в своєму складі DTD або посилаються на зовнішнє DTD.

Правильно структуровані документи хоча і поступаються по правильності документа Wolid, але підходять для більшості практичних випадків. Це означає, що ви можете зразу описувати структуру вашого документа на майже людській мові, придумуючи теги на ходу піклуючись лише про правильне вкладення тегів одних в одних. Ще одною суттєвою відмінністю є те, що в XML теги можуть бути українською мовою.

< речення >
< підмет >
< іменник > мама < / іменник >
< / підмет >
< присудок тип = „просте” >
< дієслово > мила < / дієслово >
< / присудок >
< додаток тип = „пряме” >
< іменник > раму < / іменник >
< / додаток >
< / речення >

Розробники XML дозволяють використовувати в іменах тегів і атрибутів не лише латинські букви, але й кирилицю, ієрогліфи і будь-які символи, які являються буквами хоча б в одній мові. Така розмітка дозволяє інтерпретатору XML рубати документи на куски у відповідності його теговою структурою. Після того, як весь документ розділений на окремі елементи в дію вступає інший додаток, задачею якого може бути індексування документа, занесення його у базу даних, або форматування у відповідності з прикладеною до нього стильовою специфікацією. Однак важливо розуміти, що всі ці задачі лежать уже за межами особистої мови XML, - який вільний від догляду за візуальним представленням документу і дозволяє зфокусувати на його логічній структурі.

3. Важливість використання будь-які теги означає, що будь-який HTML документ дуже легко перетворюється в XML. Зміни, що потрібні для цього перетворення досить формальні:

- 1) всі значення атрибутів повинні бути в лапках.
- 2) регістр букв у відкриваючих і закриваючих тегах повинен співпадати.
- 3) всі елементи повинні мати відкриваючий і закриваючий тег.

Це відноситься не лише до елементів з факультативними тегами, а і до пустих елементів, які в HTML мають лише відкриваючий тег. Наприклад тег потрібно буде закрити:

```
<IMG_ALT = " " _SRC = " e. gif " > < / IMG >
```

Крім цього XML допускає особливо скорочену форму запису для пустих елементів.

```
< IMG_ALT = " " SRC = "e. gif" / >
```

Існують утиліти, які переводять HTML в XML, тег в тег з дотриманням всіх правил. Результат такого переводу буде правильно структурованим документом з точки зору інтерпретатора XML, хоча розмітка не стане ні на грам більш структурованою. Тому для більш ефективного перетворення необхідно замінювати на відповідно логічні теги уніфіковані HTML, блоки, які мають разом з функцією форматування ще й структурну функцію, які дозволяють отримати на виході змістовний XML код, який містить структуровану основу документа і здатний працювати з будь-якою підключеною стильовою класифікацією.

4. Майже одночасно з самим XML консорціумом W3 був стандартизований XLL – механізм створення гіпертекстових зсилок в XML – документах. Цей аспект мови значно удосконалений порівняно з HTML. Це основні риси текстової моделі XML:

1) XML – зсилки реалізовані не на рівні тегів, а з допомогою зарезервованих імен атрибутів.

2) Для XML – зсилки можна вказати, чи буде вона звичайною зсилкою, або броузер, зустрівши в документі цю зсилку, мусить активізуватися не чекаючи іншої команди користувача.

3) Для зсилки можна вказати результат її активації а саме: чи вивести документ, на який вона зсилається або вміст нового документа треба вставити в середину текучого документа.

4) Важливі вдосконалення внесені в синтаксис URL – адрес, які використовуються у зсилках.

5. Одним з прототипів XSL – створений вже дуже давно для використання разом з SGML мова DSSSL. Так як і DSSSL, XSL пропонує два поступових етапів при обробці документа. На першому етапі ієрархічне дерево

елементів виходячого документа перетвориться в друге дерево, яке в принципі може не мати з вихідним майже нічого спільного. На другому етапі в справу вступає форматировщик інтерпретуючий туди приобразованого на першому етапі документа і виводячи його на екран, на принтер, або на будь-який другий пристрій виводу.

Якщо врахувати той факт, що словник візуального форматування XSL мусить ще пройти довгий процес реалізації і налагодження в броузерах, на даний момент більш реалістичний здається другий підхід до використання XSL.