1 історія HTML

2 Семантичні вимоги HTML

3 Складові елементи HTML-документа

Коротка історія HTML

Початок історії HTML можна віднести до 1986 року, коли Міжнародна організація по стандартизації (ISO) прийняла стандарт ISO-8879, що має назву "Standard Generalized Markup Language (SGML)". Цей стандарт описував узагальнений метамова, що дозволяє будувати системи структурної розмітки будь-яких різновидів текстів. Керуючі елементи (так звані теги), що вносяться до тексту при такій розмітці, не несли ніякої інформації про зовнішній вигляд документа, а лише задавали його логічну структуру, тобто вказували кордону і супідрядність складових частин документа. Розмічений таким чином текст могла інтерпретувати будь-яка програма, що працює на будь-якій комп'ютерній платформі з будь-яким пристроєм виведення.

Незважаючи на всю значущість принципів, що лежать в основі мови SGML, він не мав помітного поширення до тих пір, поки в 1991 р співробітник Європейського інституту фізики елементарних частинок Тім Бернерс-Лі не вибрав його в якості основи для нової мови розмітки гіпертекстових документів. Ця мова, що став найвідомішим і широко використовуваним додатком SGML, був названий HTML - HyperText Markup Language, що перекладається як мову розмітки гіпертексту.

Перші версії HTML поділяли всі особливості ідеології свого прабатька. Вся розмітка була чисто логічної, а з більш ніж сорока тегів HTML версії 1.2, що вийшла в світ в червні 1993 року, тільки три тега відповідали за фізичні параметри представлення документа.

У вересні 1993 року група програмістів Національного центру суперкомп'ютерних додатків США (NCSA) випустила перший (і на довгий час єдиний) графічний браузер Mosaic, завдяки якому мова HTML набув широкого поширення. Браузер підкорив віртуальний простір з вражаючою швидкістю - усього за рік близько двох мільйонів користувачів встановили Mosaic на свої комп'ютери.

У квітні 1994 р під егідою створеного в тому ж році Консорціуму Всесвітньої павутини (World Wide Web Consortium, W3C) почалася підготовка нової версії мови HTML 2.0, що стала офіційною рекомендацією W3C лише у вересні 1995 р Стандарт HTML 2.0 увібрав в себе всю ситуацію до 1994 році практику створення веб-сайтів.

У березні 1995 почалася робота над проектом HTML 3. До цього часу вже досить очевидно стало протиріччя між ідеологій чисто логічної розмітки існували версій HTML і потребами користувачів, зацікавлених в розширенні засобів візуального оформлення. Щоб вирішити це протиріччя, не відмовляючись від парадигми структурної розмітки, автори HTML 3 ввели підтримку нового засобу - так званих ієрархічних стильових специфікацій (Cascading Style Sheets, CSS). Стильові специфікації представляли собою окремий по відношенню до структурної розмітки "інформаційний шар" і були призначені тільки для візуального форматування структурних елементів документа. На жаль, робота над цією версією була перервана у зв'язку з відсутністю підтримки з боку виробників браузерів. Прийнята трохи пізніше рекомендація HTML 3.2 втратила багато нових властивостей 3.0, однак закріпила розробки популярних в той час браузерів Netscape Navigator і InternetExplorer.

У 1997 році консорціум W3C опублікував версію HTML 4.0 в якості рекомендації, яка включила додаткові спеціальні розширення браузерів, але спробувала також раціоналізувати і очистити HTML. Багато елементів і атрибути (в основному стосуються візуального оформлення) були позначені як що не рекомендуються. Це повинно було стимулювати більш "правильне" використання HTML.

Версія HTML 4.01 була опублікована в 1999 р Це найостанніша версія HTML, хоча W3C вже опублікував чорновий варіант специфікації п'ятої версії мови HTML. Робота над чорнової версією специфікації HTML 5 почалася в березні 2008 року. Для цього була сформована спеціальна група, яка об'єднала близько п'ятисот учасників, серед яких фахівці таких компаній, як Apple, Google, IBM, Microsoft, Mozilla, Nokia, Opera, BEA Systems, Cisco, France Telecom і Hewlett-Packard.

Серед найбільш помітних і важливих нововведень в HTML 5 Консорціум W3C виділяє програмні інтерфейси для роботи з двовимірної графікою, засоби впровадження в веб-сторінки відео- і аудіоматеріалів, а також інструменти, що дозволяють відвідувачам самостійно редагувати сайти. Ознайомитися з чорновим варіантом специфікації HTML 5 можна на офіційному сайті Консорціуму (http://www.w3.org/TR/2008/WD-html5-20080122/).

Семантичні вимоги HTML

Ключовими моментами в Специфікації HTML є синтаксичні правила HTML і визначення семантики його елементів.

Перш ніж дати визначення терміна "семантика" стосовно мови гіпертекстової розмітки, необхідно розібратися з поняттям структурної розмітки.

У загальному випадку структура документа являє собою певне розташування і зв'язок його частин, що складають єдине ціле. Згідно з ідеологією HTML, що вносяться до документ керуючі конструкції не повинні нести будь-якої інформації про візуальному представленні документа. Вони призначені тільки для позначення меж і підпорядкування складових частин HTML-документа. Іншими словами, при розмітці структури документа розробник повинен повністю абстрагуватися від питань подання та всього лише створити "каркас" майбутнього HTML-документа.

Природно, що для правильного опису структури документа необхідно чітко уявляти собі призначення кожного конкретного елемента HTML, що як раз і є семантикою елемента. Таким чином, семантична верстка являє собою вибір елементів HTML виходячи з їх смислового призначення, а не на основі того, як він виглядає в браузері.

Яскравим прикладом порушення правил семантичної верстки є так звана таблична верстка, яка представляє собою метод верстки документів, при якому структурний елемент TABLE використовується для управління візуальним розташуванням інших елементів, а не для представлення табличних даних, що є його смисловим призначенням. Тому використання елемента TABLE для подання нетаблічних даних є порушенням його семантики.

Семантична верстка невіддільна від концепції поділу структури та подання, згідно з якою мова розмітки гіпертексту HTML повинен використовуватися тільки для опису структури документа (тобто його змісту), а для візуального представлення цієї структури (тобто його оформлення) пропонується інший офіційно затверджений W3C стандарт - каскадні таблиці стилів CSS (Cascading Style Sheets).

Складові елементи HTML-документа

елементи

Будь HTML-документ являє собою набір елементів, що описують окремі складові документа, такі як заголовки, списки, абзаци тексту, таблиці та ін. Імена більшості елементів являють собою загальновживані слова англійської мови, зрозумілі скорочення і позначення.

Найчастіше елемент розмітки складається з трьох частин: початкового та кінцевого компонентів, між якими розміщуються текст або інші елементи документа. Ці компоненти являють собою спеціальні керуючі конструкції для розмітки вмісту HTML-документа і називаються тегами. Початковий тег складається з імені елемента, укладеного в кутові дужки (<і>). Кінцевий тег відрізняється від початкового тим, що перед ім'ям елемента в ньому ставиться коса риска (/). Наприклад, елемент EM, який використовується для акцентування тексту, буде мати наступний вигляд:

<EM> акцентіруемий текст </ EM>

Слід також зауважити, що імена елементів не чутливі до регістру, тобто записи <EM> і <em> рівнозначні.

Елементи повинні або слідувати один за одним, або бути вкладені один в інший. Так, якщо початковий тег <STRONG> розташований всередині елемента <P> ... </ P>, то і кінцевий тег </ STRONG> повинен бути розташований всередині цього елемента.

Кінцеві теги деяких елементів в документі можна опускати, тому що більшість браузерів влаштовано так, що при обробці тексту документа початковий тег сприймається як кінцевий тег попереднього. Наприклад, кінцевий тег елемента LI, використовуваного для створення пункту списку, не обов'язковий, оскільки початок чергового пункту списку означає кінець попереднього. Є й інші кінцеві теги, без яких браузери відмінно працюють, наприклад, кінцевий тег </ HTML>. Проте, рекомендується не опускати кінцеві теги, щоб уникнути плутанини і помилок при відтворенні документа. Крім того, деякі елементи, такі, як, наприклад, елемент IMG, який використовується для вставки в HTML-документ зображення, не мають кінцевого тега, оскільки не мають вмісту.

Всі елементи HTML діляться на дві категорії: блокові елементи і строкові елементи. Блоковий елемент зазвичай інформує про структуру документа. За замовчуванням блоки починаються з нового рядка і відокремлюються від попередніх і наступних блоку символом нового рядка певної ширини. Блоки можуть бути вкладені одна в одну, а блокові елементи можуть містити рядкові елементи. До основних блоковим елементам ставляться параграфи, заголовки, попередньо форматований текст, цитати, роздільник, таблиці і списки.

Строкові елементи містяться всередині структурних блокових елементів і охоплюють не цілі галузі, а тільки частини тексту документа. Строкові елементи зазвичай з'являються в параграфах тексту і не призводять до появи в документі нового рядка. Поширеними строковими елементами є гіпертекстові посилання, зображення, виділені слова або фрази і короткі цитати.

атрибути

Елементи можуть також мати атрибути, які визначають додаткову інформацію про них. Багато атрибути в HTML є загальними для всіх елементів, проте більшість з них є специфічними для даного елемента або групи елементів.

Атрибути розміщуються в початковому тегу після імені елемента і відокремлюються один від одного одним або декількома пропусками, причому порядок проходження атрибутів у тегу значення не має. Значення атрибута (якщо таке є) слід за знаком рівності, хто стоїть після імені атрибута. Значення повинне існувати в одиночні або подвійні лапки. Незважаючи на те, що в деяких ситуаціях лапки можна опустити, Специфікація HTML рекомендує завжди укладати значення атрибутів в лапки. Імена атрибутів можуть бути набрані в будь-якому регістрі, проте їх значення можуть залежати від регістру.

Всі атрибути і їх можливі значення визначаються в основному Специфікаціями HTML (http://www.w3.org/TR/html401/index/attributes.html). Користувач не може створювати свої власні атрибути або використовувати значення, які не визначені Специфікацією, так як це може викликати проблеми правильної інтерпретації веб-сторінки.

Як було відзначено, ряд атрибутів можна застосувати практично до всіх елементів HTML. Найбільш часто використовуваними атрибутами є базові атрибути (class, id, style і title), що визначають загальні властивості елементів, і локалізують атрибути (dir і lang), які вказують на властивості мови написання вмісту елемента. Коротко розглянемо ці атрибути.

Атрибут id, званий також ідентифікатором елемента, привласнює елементу ім'я, унікальне в межах даного документа, тобто ніякі два елементи не можуть мати однакових значень id. В основному ідентифікатор застосовується в якості селектора стилів окремих елементів, як закладки для гіперпосилань, а також для вказівки на конкретний елемент зі сценарію. У наступному прикладі атрибут id використаний для ідентифікації параграфа:

<P id = "footer"> Знання - сила! </ P>

Атрибут class задає приналежність елемента певного класу. На відміну від атрибута id, до одного класу може ставитися будь-яку кількість елементів. Крім того, елемент може належати до декількох класів: в цьому випадку значенням атрибута є список імен класів, розділених пробілами. Однак більшість браузерів не підтримують списки класів. Атрибут class в основному використовується в якості селектора стилів. У наступному прикладі створений параграф включається в клас елементів з ім'ям header:

<P class = "header"> Заголовний текст </ P>

Атрибут style дозволяє задати стиль елемента. Однак для використання цього атрибута в заголовок документа потрібно включити метаопісатель. Наступний приклад демонструє спосіб зміни кольору шрифту для заданого параграфа:

<! DOCTYPE HTML PUBLIC "- // W3C // DTD HTML 4.01 // EN"

"Http://www.w3.org/TR/html4/strict.dtd">

<HTML>

<HEAD>

<META http-equiv = "Content-Style-Type" content = "text / css">

...

</ HEAD>

<BODY>

...

<P style = "color: red"> Мир, труд, май! </ P>

...

</ BODY>

</ HTML>

У більшості випадків вживання атрибутів class та id краще, тому що вони забезпечують поділ вмісту документа і стилю його відображення, що більшою мірою узгоджується з концепцією поділу структури та подання.

Всі три описаних вище атрибута можна застосувати до всіх елементів, крім елементів BASE, HEAD, HTML, META, SCRIPT, STYLEі TITLE.

Атрибут title визначає заголовок елемента і часто використовується оглядачами як підказка, яка виводиться на екран, коли курсор поміщається на даний елемент. Наведений нижче приклад демонструє використання атрибута titleсовместно з елементом A, що створює посилання на веб-сторінку:

<A href="http://www.microsoft.com"

title="Официальный сайт Microsoft">Microsoft</A>

Атрибут lang визначає мову, на якому написані значення інших атрибутів даного елемента і його вміст, а також всіх вкладених елементів, які не мають аналогічного атрибута. Використання атрибуту дозволяє авторам змінювати стиль тексту в залежності від мови. Наприклад, текст двомовного документа може бути оформлений наступним чином:

...

<P lang = "en"> This paragraph is in English. </ P>

<P lang = "ru"> Цей абзац російською мовою. </ P>

...

Атрибут dir визначає напрямок виведення тексту: зліва направо (dir = "ltr", за замовчуванням) або справа наліво (dir = "rtl"). Для всіх символів в кодуванні Unicode з метою правильного відображення тексту визначено напрямок. Так, латинські і російські літери виводяться зліва направо, а єврейські і арабські - справа наліво.

Стандарт Unicode визначає двонаправлений алгоритм, який повинен застосовуватися щоразу, коли документ містить символи, що виводяться справа наліво. Хоча зазвичай цей алгоритм дає правильне зображення тексту, існують ситуації, коли напрямок виведення тексту доводиться ставити явно за допомогою атрибута dir.

Спеціальні символи і посилання-мнемоніки

У документ HTML також можуть бути включені посилання-мнемоніки (або символічні посилання). Дані посилання призначені для введення в документ нестандартних і спеціальних символів. До нестандартних відносяться символи, які важко або неможливо ввести за допомогою клавіатури або в кодуванні конкретного документа. До спеціальних символів відносяться символи, які починають і закінчують частини документа HTML, а не уявляють відповідні символи (<,>, & і ").

Посилання-мнемоніки починаються знаком & і закінчуються крапкою з комою (;). Однак багато браузери можуть бути досить лояльні до помилок HTML, таким як відсутність крапки з комою.

Посилання можуть бути або числами (числові посилання), або скороченими словами (об'єктні посилання). Наприклад, амперсанд може бути введений в документ як & amp ;, що є об'єктної посиланням символу, або як &, що є числовий посиланням. Повну таблицю символьних посилань можна знайти на сайті evolt.org (http://www.evolt.org/article/A\_Simple\_Character\_Entity\_Chart/17/21234).

Коментарі в HTML

HTML-документи можуть містити коментарі, які є зручним засобом для опису коду, пошуку розділів документа або пояснення мотивів, якими керувався розробник коду. Коментарі не впливають на відображення документа, а тільки пояснюють його вміст при перегляді HTML-коду. Текст коментарів розташовується між двома групами символів: відкриває обмежувачем (<! -) і закриває обмежувачем (->). У тексті коментарів забороняється використовувати два або більше поспіль дефіса.

Коментарі в HTML можуть виглядати наступним чином:

<! - це коментар ->

<! - це коментар,

     що займає два рядки ->

Структура HTML-документа

Приклад HTML-документа виглядає наступним чином:

<! DOCTYPE HTML PUBLIC "- // W3C // DTD HTML 4.01 // EN"

 "Http://www.w3.org/TR/html4/strict.dtd">

<HTML>

<HEAD>

    <TITLE> Приклад простого HTML-документа </ TITLE>

</ HEAD>

<BODY>

    <P> Це найпростіший HTML-документ. </ P>

</ BODY>

</ HTML>

Приклад виконання цього коду браузером представлений на малюнку 4.1.



Мал. 4.1. Вид в браузері простого HTML-документа

Кожен документ рекомендується починати з елемента декларації типу документа, або DOCTYPE, який описує використовуваний тип і версію HTML. Дана інформація призначена насамперед для браузерів і програм-валідаторів розмітки. DOCTYPEуказивает браузеру основні синтаксичні елементи для кожної версії HTML, а також режим відтворення. Валідатори розмітки переглядають DOCTYPE, щоб визначити, згідно з якими правилами вони повинні перевіряти даний документ.

За декларацією типу документа слід елемент <HTML> ... </ HTML>, який вказує, що наступний документ є HTML-документом і включає в себе все інше вміст документа. Даний елемент називається кореневим елементом документа. Він може містити тільки елементи HEAD і BODY, причому кожен з цих елементів може зустрічатися всередині елемента HTML не більше одного разу. Даний елемент має два необов'язкових атрибути: lang і dir.

Усередині елемента HTML розташовується заголовок. Заголовок документа, що створюється елементом <HEAD> ... </ HEAD>, містить інформацію про загальні властивості документа. Даний елемент в обов'язковому порядку повинен містити елемент TITLE і може містити необов'язкові елементи BASE, SCRIPT, STYLE, META, LINK і OBJECT. Елемент HEAD має три необов'язкових атрибути: lang, dir та profile; останній вказує місцезнаходження словника метаданих. Передбачається, що такий словник повинен містити імена метапеременних, значення яких визначаються елементами META і LINK в заголовку документа.

Після елемента HEAD слід елемент <BODY> ... </ BODY>, який є оболонкою, яка містить реальний контент веб-сторінки. Елемент BODY обов'язково повинен містити хоча б один блоковий елемент: в даному випадку це елемент P, створює параграф, який містить текст "Це найпростіший HTML-документ.". Елемент BODY має досить багато атрибутів, проте рекомендованими до застосування є тільки id, class, style, title, lang і dir.

Редактори для верстки веб-сторінок

Редактор для верстки веб-сторінок або HTML-редактор - це комп'ютерна програма, що дозволяє складати і редагувати документи в форматі HTML. Сьогодні на запит в мережі Інтернет за ключовим словосполученням "HTML-редактор" можна отримати перелік із сотні різноманітних програм. Однак все редактори діляться на два класи: візуальні і текстові.

Візуальні редактори не вимагають від розробника знань HTML, CSS і інших технологій розмітки документів. Користувач просто має різні елементи майбутньої сторінки в вікні редагування, а редактор сам генерує відповідний код. Саме тому візуальні редактори ще називають WYSIWYG-редакторами, що означає What You See Is What You Get - що бачиш, те й отримуєш.

Візуальні редактори зазвичай розташовують масою функцій, покликаних полегшити роботу розробнику. Так, хороший редактор повинен володіти підсвічуванням синтаксису, можливістю попереднього перегляду створеного документа, менеджером проектів, бібліотеками різних елементів і ін.

До популярних візуальних редакторам відносяться продукти компанії Adobe - GoLive і Dreamweaver, а також Microsoft FrontPage і Microsoft Expression Web. Дані редактори є пропрієтарними, хоча виробники пропонують всім бажаючим ознайомчі версії даних програм. Так, ознайомчу версію Microsoft Expression Web можна отримати, перейшовши за посиланням: //www.microsoft.com/downloads/details.aspx? Displaylang = en & FamilyID = 0a73a3a7-3e06-4125-b3c6-f9c10387e9cc. Серед вільно розповсюджуваних редакторів варто виділити редактор Nvu, який хоч і поступається у функціональності згаданих вище редакторам, однак безкоштовне поширення робить його привабливим для багатьох розробників.

Однак жоден візуальний редактор не досконалий, і всі вони так чи інакше обмежені в своїх можливостях. Тому професійні розробники часто користуються невеликими текстовими редакторами. Текстові редактори, як правило, містять набір функцій, що полегшують розробнику написання коду. До найбільш поширених функцій відносяться підсвічування коду, різні гарячі клавіші і т.д. До найбільш популярним текстовим редакторам відносяться Macromedia HomeSite і HTML Pad. Однак лідером серед текстових редакторів є Notepad (він же Блокнот). У цій програмі немає ніяких спеціальних функцій, але зате вона є у кожного користувача Windows, тому що входить в число стандартних. З неї можна почати свої перші кроки в написанні коду, а потім вже змінити на більш сподобався редактор.