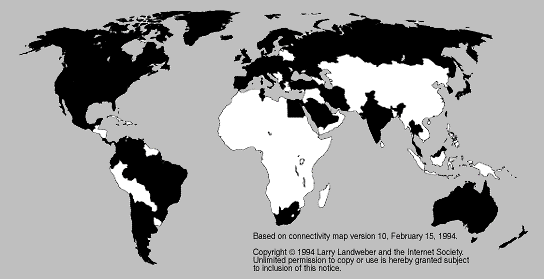
**World Wide Web i HTML**

Розділ I.

Вступ

World Wide Web

Internet, як ми вже знаємо, - це найбільша світова комп'ютерна мережа. Тепер Internet має приблизно 150 мільйонів користувачів більш ніж в 50 країнах. WWW доступний в основному через Internet; але кажучи WWW і Internet ми маємо не одне і те ж. WWW можна віднести до внутрішнього змісту, тобто це якийсь абстрактний мир знань, в той час як Internet є зовнішньою стороною глобальної мережі у вигляді величезної кількості кабелів і комп'ютерів.



Малюнок 1. Країни, позначені чорним кольором, мають зв'язок по Internet. Кількість людей, що мають такий доступ, в цих країнах стає все більше і більше. Країни, позначені білим кольором можуть працювати по e-mail, в локальних мережах або не мають нічого подібного взагалі.

1.1. Трохи історії

Історія WWW - Світової Інформаційної Павутини почалася в Березні 1989, коли Tim Berners-Lee з Європейської Лабораторії Фізики Елементарних Часток (відомої як CERN), де працював колектив дослідників-фізиків, запропонував новий спосіб обміну результатами досліджень і ідеями між організаціями. Такий обмін був дуже необхідний, тому що члени цієї організації працювали в різних країнах.

Пропонувалося використати просто систему гіпертекста для передачі документів і встановлення зв'язку між членами суспільства фізиків - ядерників. Тоді ще не було планів застосування звуку або відео, і не розглядалася можливість передачі зображення.

До кінця 1990, уперше програмне забезпечення WWW було встановлене на ЕОМ NeXT. Тепер можна було переглядати і передавати документи гіпертекста іншим людям через Internet, а також з'явилася можливість редагувати документи гіпертекста прямо на екрані комп'ютера. Програма спочатку демонструвалася перед членами суспільства і на семінарах в CERN'е, а потім була показана на конференції “Гіпертекст'91”.

До 1992 Tim продовжував виступати зі своїм проектом, доти, поки не з'явилися бажаючі продовжити роботу над цією проблемою.

Сотні людей з всього світу взяли участь в розробці цього проекту, одні писали програми і документи для WWW, інші просто розказували людям про WWW. Група піонерів-проектувальників WWW навіть не могла передбачати тоді, що почата ними справа досягне таких масштабів. Тільки за перших чотири місяці 1994 об WWW говорили і писали CNN, “Wall Street Journal”, “Economist”, “Fortune”, “New York Times” та інші видання.

1.2. Що таке WWW?

Так що ж таке World Wide Web, або, як кажуть в просторіччі, WWW, the Web, або ще простіше - 3W? WWW - це розподілена інформаційна система мультимедіа, заснована на гіпертексті. Давайте розберемо це визначення по порядку.

*Розподілена інформаційна система:* інформація зберігається на величезній безлічі так званих WWW-серверів (servers). Тобто комп'ютерів, на яких встановлене спеціальне програмне забезпечення і які об'єднані в мережу Internet. Користувачі, що мають доступ до мережі, отримують цю інформацію за допомогою програм-клієнтів, програм перегляду WWW-документів. При цьому програма перегляду посилає по комп'ютерній мережі запит серверу, що зберігає файл з необхідним документом. У відповідь на запит сервер висилає програмі перегляду цей необхідний файл або повідомлення про відмову, якщо файл по тих або інших причинах недоступний. Взаємодія клієнт-сервер відбувається за певними правилами, або, як кажуть інакше, протоколу. Протокол, прийнятий в WWW, називається HyperText Transfer Protocol, скорочено - HTTP.

*Мультимедіа:* інформація включає в себе не тільки текст, але і двох- і трьохмірну графіку, відео і звук.

*Гіпертекст:* інформація в WWW представляється у вигляді документів, кожний з яких може містити як внутрішні перехресні посилання, так і посилання на інші документи, що зберігаються на тому ж самому або на будь-якому іншому сервері.

Такі посилання називають гіперпосиланнями або гіперзв'язками. На екрані комп'ютера у вікні програми перегляду посилання виглядають як виділені яким-небудь чином (наприклад, іншим кольором і/або підкресленням) дільниці тексту або графіки. Вибираючи гіперпосилання, користувач програми перегляду може швидко переміщатися від однієї частини документа до іншої, або ж від одного документа до іншого. При необхідності програма перегляду автоматично зв'язується з відповідним сервером в мережі і запитує документ, на який зроблене посилання. До речі, ідея гіпертекстового представлення інформації повинна вже бути добре знайома користувачам різних версій системи Microsoft Windows. Саме за цим принципом побудована в Windows система підказок (Help), з тією лише різницею, що гіпертекстова система підказок Windows не є розподіленою.

Отже, WWW-документ може містити стилізований і форматований текст, графіку і гіперзв'язки з різними ресурсами Internet. Щоб реалізувати всі ці можливості, була розроблена спеціальна мова, названа HyperText Markup Language (HTML), тобто, Мова Розмітки Гіпертекста. Документ, написаний на HTML, являє собою текстовий файл, що містить власне текст, несучий інформацію читачеві, і прапори розмітки. Останні являють собою певні послідовності символів, що є інструкціями для програми перегляду; згідно з цими інструкціями програма розташовує текст на екрані, включає в нього малюнки, що зберігаються в окремих графічних файлах, і формує гіперзв'язки з іншими документами або ресурсами Internet. Таким чином, файл на мові HTML набуває вигляд WWW-доку-мента тільки тоді, коли він інтерпретується програмою перегляду. Про мову HTML буде детально розказано у відповідному розділі, оскільки без знання основ цієї мови неможливо створити сторінку (тут і далі в тексті мається на увазі WWW-сторінка) для публікації в WWW.

1.3. Як в WWW задається місцезнаходження документа?

Віртуальний світ WWW населений мільйонами документів, що мешкають на сотнях тисяч серверів. Задача WWW-мандрівника - знайти серед них той, який містить необхідну інформацію, і прочитати його за допомогою програми перегляду, - а для цього програма перегляду повинна знати точне місцезнаходження даного документа. Воно однозначно визначається адресою сервера, номером порту, ім'ям директорії і ім'ям файла з цим документом. Для успішного завантаження необхідного документа сервера програмі перегляду також треба указати протокол, що використовується цим сервером. Це пов'язано з тим, що крім WWW-серверів, програми перегляду можуть звертатися до серверів пересилки файлів (так званим FTP-серверам), серверів телеконференцій, серверів електронної пошти і деяких інших типів серверів. Комбінація протоколу, адреси сервера, номера порту, імені директорії і імені файла отримала назву Uniform Resource Locator, або скорочено URL. Синтаксис URL показаний нижче:

протокол://адреса\_сервера:номер\_порта/ім'я\_директорії/ім'я\_файла

Протокол, як вже було сказано вище, це сукупність правил, по яких відбувається взаємодія клієнт-сервер. Ми вже знаємо, що “рідний” протокол WWW називається http. Крім http, WWW-клієнти можуть спілкуватися з серверами, підтримуючими протоколи ftp, gopher і деякими іншими, на яких я тут зупинятися не буду, а розкажу про них далі.

Назва протоколу відділяється від іншої частини URL двокрапкою. У назві протоколу великі латинські букви і відповідні ним малі букви еквівалентні.

Початок адреси відмічається двома косими // Він складається з декількох частин - так званих піддоменів. Піддомени, що являють собою поєднання букв і цифр, розділяються точками, причому, “уточнення координат адресата” від більш великих до більш дрібних піддоменів відбувається з права на ліво. Так само як і в назві протоколу, в адресі сервера великі латинські букви і відповідні ним малі букви еквівалентні.

Номер порту виражається цілим позитивним числом і відділяється від адреси двокрапкою. Порт - це неначе “двері”, через які можна зайти в сервер. Сервер може мати декілька доступних портів; якщо номер порту WWW-сервера рівний 80, то в URL його можна не вказувати.

Файли з документами, що публікуються в WWW, часто мають суфікси. html або. htm. Ці суфікси є абревіатурою від назви мови, на якій пишуть WWW-документи, HyperText Markup Language, про яке буде детально розказано далі.

Якщо ім'я директорії починається з символа ~ (тильда), це означає, що це - так звана домашня директорія (home directory) користувача даної комп'ютерної системи. Наприклад, на комп'ютерах з операційною системою UNIX кожному користувачеві привласнюється ім'я користувача, під яким він реєструється в системі, і виділяється власна “домашня” директорія, призначена для зберігання файлів, що належать йому. Ім'я домашньої директорії співпадає з ім'ям користувача. Замість абсолютного імені домашньої директорії з вказівкою всіх її піддиректорія можна використати ім'я користувача, якому передує символ ~ (тильда).

Треба звернути увагу на наступні особливості завдання імен директорій і файлів в URL:

Директорія, вказана в URL, як правило, не співпадає з істинною директорією в файловій системі на тому сервері, де знаходиться необхідний файл. Перетворення директорій, заданих в URL, в істинні директорії в файловій системі здійснюється WWW-сервером згідно з правилами, заданими при його установці і настройці.

Якщо в URL вказане тільки ім'я директорії, а ім'я файла не задане, то сервер на такий запит вишле файл з певним ім'ям, що залежить від настройки сервера. Якщо ж такого файла в заданій директорії немає, то більшість серверів автоматично вишлють вам список всіх файлів в даній директорії.

1.4. Програми перегляду

Ми вже знаємо, що, програма, з якою безпосередньо спілкується користувач при роботі з ресурсами WWW, називається програмою перегляду WWW-документів. Нагадуємо, що робота такої програми полягає в посилці запиту до заданого сервера, інтерпретації отриманої інформації і представлення її користувачеві. Крім того, вона виконує ряд допоміжних задач, наприклад, підтримує список прочитаних документів, щоб користувач міг до них повернутися надалі.

У цей час набули поширення більш десяти програм перегляду WWW-документів. З графічних програм перегляду найбільшої уваги, на наш погляд, заслуговують NCSA Mosaic, Netscape Navigator і Microsoft Internet Explorer.

Програма NCSA Mosaic була створена в Національному Центрі Суперкомп'ютерних Додатків Університету штату Іллінойс в Урбана-Шампейн на початку 1993 року групою програмістів, очолюваних студентом на ім'я Марк Андреєссен. У той час Mosaic була єдиною програмою перегляду, написаною на професійному рівні, дружньою по відношенню до користувача і працюючою на багатьох типах комп'ютерів. Не викликає сумнівів, що саме Mosaic відкрила World Wide Web для більшості користувачів. У цей час в NCSA група розробників (в основному, студентів старших курсів) продовжує трудитися над новими версіями Mosaic, незважаючи на те, що по своїх можливостях ця програма вже істотно поступається більш новим програмам. Mosaic є некомерційним програмним продуктом, доступним для користувачів безкоштовно. Початковий код програми Mosaic відкритий і активно використовується розробниками багатьох інших програм перегляду.

Netscape Navigator (скорочено - Netscape) - це в даний момент, мабуть, найбільш популярна програма перегляду, яка працює на тих же типах комп'ютерів, що і Mosaic. Netscape - програма компанії Netscape Communications Corporation, заснованої на початку 1994 року Джимом Кларком, який в свій час заснував велику компанію Silicon Graphics, знамениту своїми графічними станціями. Джиму Кларку вдалося переманити в Netscape Communications Corporation первинних розробників програми Mosaic з Іллінойса, в тому числі і самого Марка Андреєссена. Netscape - це комерційний продукт, за користування яким треба платити. Однак, для користувачів в університетах Netscape доступна безкоштовно. Крім версій, робота над якими завершена, компанія періодично випускає відлагоджувальні версії Netscape (так звані бета-версії), якими можна користуватися безкоштовно до настання певної дати, після якої програма перестає функціонувати.

Microsoft Internet Explorer (скорочено - IE) - витвір фірми Microsoft Corporation. По функціональних можливостях і зручності використання IE приблизно еквівалентний згаданій вище програмі Netscape Navigator. На відміну від останньої, IE працює на значно меншій кількості комп'ютерних платформ. IE розповсюджується безкоштовно для всіх користувачів.

Крім Mosaic, Netscape і IE, варто згадати такі програми як Lynx, IBM Web Explorer і AOL Browser.

Аналіз доступу на сервер Engineering показує, що найбільш широко використовується Netscape Navigator. У цей час між Microsoft Corporation і Netscape Corporation йде найгостріша боротьба за першість на ринку програмного забезпечення для Internet, і певно, що Microsoft незабаром істотно потіснить Netscape і інших своїх конкурентів.

1.5. Основні команди програм перегляду

Віконний інтерфейс графічних програм перегляду простий і інтуїтивно зрозумілий всім користувачам, знайомим з Microsoft Windows. У смузі заголовка вікна, яка розташовується вздовж верхнього кордону вікна, виводиться назва програми перегляду - Netscape - і назва поточного документа. Далі у верхній частині вікна розташовані елементи управління програмою: меню і екранні кнопки. Велику частину вікна займає власне область, в якій виводиться зміст WWW-документів, що переглядаються. У нижній частині вікна розташований рядок стану. У цьому рядку Netscape відображає таку інформацію, як, наприклад, процентну частку від повного розміру файла, що завантажується в даний момент або URL документа, на який вказує гіпертекст, якщо курсор миші вмістити на відповідну дільницю документа.

У таблиці нижче приведені команди, необхідні для виконання найпростіших дій при перегляді документів в WWW при допомозі Netscape Navigator 3.0. Система команд і меню в інших графічних програмах не набагато відрізняється від прийнятої в Netscape. Потрібно врахувати, що автори програм перегляду з якихось таємничих спонук використовують різні терміни для позначення одних і тих же понять.

|  |  |
| --- | --- |
| Дія | Команда в Netscape 3.0 |
| Відкрити документ з відомим URL на WWW-сервері. | Меню: File ¦ Open Location  Потім ввести URL в діалоговому вікні, що з'явилося і “натиснути” екранну кнопку OK.  Або: ввести URL в полі “Location:”, розташоване у верхній частині вікна, і натиснути клавішу Enter. |
| Відкрити домашню сторінку (homepage). | “Натиснути” екранну кнопку із зображенням будиночка Home |
| Перервати завантаження документа. | “Натиснути” екранну кнопку Stop |
| Відкрити документ, що знаходиться в файлі на локальному комп'ютері. | Меню: File ¦ Open File |
| Перезавантажити поточний документ. | “Натиснути” екранну кнопку Reload. |
| Зберегти поточний документ в файлі. | Меню: File ¦ Save As |
| Перейти до перегляду документа, з яким є гіперзв'язок в поточному документі. | Перемістити курсор миші у виділену підкресленням і/або контрастним кольором дільницю на екрані і натиснути ліву клавішу миші. |
| Рухатися назад по ланцюжку документів, переглянених в поточному сеансі роботи. | “Натиснути” екранну кнопку << Back |
| Повернутися до довільного документа, перегляненого в поточному сеансі роботи. | Меню: Go, потім указати назву необхідного документа з списку. |
| Рухатися вперед по ланцюжку переглянених документів. | “Натиснути” екранну кнопку >> Forward |
| Запам'ятати URL поточного документа в списку “закладень” для подальшого повернення до нього | Меню: Bookmark ¦ Add Bookmark |
| Відкрити документ з списку “закладень”. | Меню: Bookmark  Вибрати документ з меню, що спускається |

Програми перегляду здатні правильно відобразити вміст файлів самих різних форматів, починаючи від найпростіших текстових і закінчуючи графічними, звуковими і іншими спеціалізованими форматами; при цьому, як ми вже знаємо, “рідним” форматом для WWW є HyperText Markup Language (HTML). А що ж відбувається, коли програма перегляду “не розуміє” формату файла, що завантажується? У перший раз зіткнувшись з цією нерідкою ситуацією, можна і розгубитися, тому нижче буде стисло описано два можливих варіанти розвитку подій:

1. Програма перегляду знає, яка з інших програм, встановлених на вашому комп'ютері, розуміє формат даного файла. - В цьому випадку програма перегляду запустить необхідну програму-помічника або програму-префікс і передасть їй отриманий файл для обробки. Як правило, програма перегляду, перш ніж запустити програму-помічника, питає вашого дозволу. Це робиться по тій причині, що навіть самий невинний на перший погляд файл може бути потенційно небезпечний для вашого комп'ютера. Зокрема, документи в форматі Word for Windows 6.0 можуть містити макрокоманди, здатні знищувати або переписувати файли на вашому жорсткому диску.

2. Програма перегляду не знає, в якої програми попросити допомоги при інтерпретації даного файла. - В цьому випадку вам буде запропоновано чотири альтернативи:

More Info... - дати додаткову інформацію про ситуацію, що виникла і запропонувати встановити програму-префікс (plug-in), який розуміє даний формат файла,

Pick App... - вибрати програму-помічника, вже встановлену на вашому комп'ютері, для обробки даного файла,

Save File... - зберегти файл на локальному диску,

Cancel - відмінити завантаження даного файла. Часто команду Save File... використовують для того, щоб отримати програмне забезпечення і документацію з колекцій (“архівів”).

**Проблема русифікації**

Творці програмного забезпечення для роботи в WWW спочатку не були дуже стурбовані потребами людей, бажаючих публікувати і читати інформацію на своїх рідних мовах, що не використовують латинський алфавіт, в тому числі і на російській мові. Останнім часом ситуація починає помітно змінюватися до кращого, але все ж перегляд і публікація документів на російській мові зв'язана з деякими труднощами.

1.6. Коди і символи

Файл, що містить гіпертекстовий WWW-документ, являє собою текстовий файл. “Всередині” комп'ютера сучасної архітектури і при передачі по мережах кожний символ тексту представляється у вигляді цілого числа, яке, в свою чергу, кодується комбінацією з восьми двійкових розрядів, званих бітами. Така комбінація з восьми біт, що обробляються ЕОМ як одне ціле, отримала назву байт. Кожний біт в байті може мати рівно два стани: “включений” і “вимкнений”, або “1” і “0”.

Легко пересвідчитися, що існує рівно 256 комбінацій з восьми біт, кожний з яких може бути зайнятий або нулем, або одиницею. Таким чином виходить, що вісьма бітами (тобто, одним байтом) можуть бути представлені числа, або коди, від 0 до 255 (тобто, від 00000000 до 11111111 в двійковій системі числення). Кожному коду можна поставити у відповідність певний текстовий символ, наприклад, букву або цифру, або керуючий символ, такий як повернення каретки, перехід на новий рядок і т. п. Щоб текст виглядав однаково на моніторах різних комп'ютерів, необхідний певний стандарт на відповідність кодів і символів, що представляються ними для текстової інформації. Такий стандарт, прийнятий в цей час на переважній більшості комп'ютерних систем, отримав назву American Standard Code for Information Interchange (ASCII, вимовляється як “аскі”). Цей стандарт охоплює лише коди від 0 до 127. У кодовій таблиці ASCII не знайшлося місця для багатьох спеціальних символів, що часто використовуються. Також, з буквених символів там присутні тільки символи англійського алфавіта. Щоб закодувати букви національних алфавітів, в тому числі російського, зберігши при цьому сумісність з таблицею ASCII, необхідно використати коди в діапазоні від 128 до 255. Ось тут-то і починаються складності.

**Кодові таблиці для кодування букв російського алфавіта (кирилиця)**

Існує досить багато різних кодових таблиць, співпадаючих в діапазоні кодів від 0 до 127 зі стандартом ASCII і використовуючих діапазон кодів від 128 до 255 для спеціальних символів і букв російського алфавіта. Серед них найбільш поширені наступні чотири:

1. Microsoft Windows Cyrillic code page 1251

Відома також як CP-1251 або Windows-1251. Найбільш широко застосовується в “русифікованих” системах Microsoft Windows 3.1, 95, 98 і NT.

2. KOI8-r

Базується на державному стандарті Коду Обміну Інформацією КОІ8 (ГОСТ 19768-74). Застосовується в основному на комп'ютерах з операційною системою UNIX. Прийнята за стандарт кодування російськомовних текстів при обміні по електронній пошті. Більшість WWW-серверів зберігають російськомовні документи в цьому кодуванні. У цей час є набори шрифтів для Microsoft Windows в кодуванні KOI8-r. Вони були розроблені спеціально для мережевих програм, працюючих під Windows.

3. CP-866 Microsoft/IBM code page 866

Відома також як “альтернативне кодування ГОСТа” (в деяких документах її означають Alt-GOST або alt). Застосовується в основному на персональних комп'ютерах IBM PC з операційною системою MS-DOS при роботі в текстовому режимі. Програми перегляду під DOS практично не використовуються (звичайно на IBM-сумісних машинах вони працюють під Microsoft Windows). Однак, ми згадуємо це кодування, оскільки текст WWW-документа можна створювати в текстовому редакторі, працюючому під DOS.

4. ISO-8859-5

Розташування російських букв в ній практично співпадає з так званим “основним кодуванням ГОСТа” (іноді можна зустріти її позначення як Main-GOST). Застосовується рідко, хоч і є міжнародним стандартом кодування російського алфавіта, зареєстрованим International Standards Organization (ISO).

Декілька особняком від вищепереліченого кодування стоїть кодування Unicode, яке, за задумом її розробників, що входять в так званий консорціум Unicode, повинне раз і назавжди вирішити проблему зберігання в текстових файлах символів будь-кого з існуючих на Землі систем писемності. На жаль, поки ще Unicode використовується вельми рідко.

При перенесенні файлів з текстами на російській мові з одного комп'ютера на інший, або навіть з однієї програми в іншу, досить часто виникає необхідність перекодування таких файлів. Для перекодування файлів використовують спеціальні програми.

Для читання документів на російській мові ви повинні встановити в програмі перегляду шрифт, що використовує одну з кодових таблиць, що містять букви російського алфавіта (кирилиця).

**Узгодження кодування сервера і програми перегляду**

Якщо спробувати прочитати російськомовний WWW-документ, закодований за допомогою однієї кодової таблиці, програмою перегляду, що використовує шрифти, розраховані на іншу таблицю, то російський текст буде виглядати як безглуздий набір знаків. Наприклад, слово Привіт!, вислане сервером в кодуванні KOI8-r, при використанні програмою перегляду шрифту в кодуванні Windows-1251 виглядає на екрані як рТЙЧіФ! Як же примусити сервер і програму перегляду настроїтися на яке-небудь одне кодування?

Іноді турботу про відповідність кодових таблиць сервера і програми перегляду бере на себе сервер. При цьому він повинен визначити кодування, на яке настроєна програма перегляду, і висилати документи саме в цьому кодуванні. Для автоматичного визначення використовується можливість протоколу HTTP 1.0 передавати в заголовку запиту перелік допустимих форматів документів і наборів символів MIME content-type і charset. З багатьох причин цей підхід досить часто не спрацьовує. У такому випадку автори документів, розміщених на сервері, часто вдаються до більш універсального прийому, пропонуючи читачеві з декількох гіперпосилань вибрати ту, яка вказує на потрібний документ в бажаному кодуванні.

Деякі програми перегляду уміють самі підстроюватися під кодування документа, що висилається сервером, якщо кодування правильно вказане в заголовку відповіді WWW-сервера в спеціальному полі charset, передбаченому протоколом HTTP 1.0. На жаль, багато які сервери не настроєні так, щоб додавати це поле автоматично.

1.7. Області використання WWW

**Наука**

Як відмічалося у введенні, WWW була створена в одній з ведучих науково-дослідних установ - ЦЕРНе (CERN) - саме з метою поширення наукової інформації. У цей час в WWW можна знайти велику кількість публікацій в самих різних областях науки і техніки.

Незабаром можна чекати лавиноподібного збільшення числа наукових публікацій в WWW і народження нових спеціалізованих наукових електронних журналів, не поступливих за якістю відомим друкарським виданням. Причому, сам гіпертекстовий характер WWW з можливістю негайного доступу до документів по гіперпосиланням як не можна краще відповідає характеру наукової інформації.

Досвід показує, що можливості WWW як джерела власне наукової інформації, тобто, результатів конкретних досліджень або довідкових даних, поки ще більш ніж скромні. Але ось як засіб налагодження контактів і пошуку інформації про те, “де що робиться і де що публікується”, World Wide Web вже не знає собі рівних.

**Університетські інформаційні системи**

Практично кожний великий університет в світі має університетську інформаційну систему, засновану на WWW. Задача такої системи - дати інформацію про факультети, кафедри і лабораторії, наукових дослідженнях і учбових планах, університетському суспільному і культурному житті, необхідну як для самих співробітників і студентів університету, так і для всіх зацікавлених осіб. Можна назвати наступні основні цілі створення університетських інформаційних систем:

залучення абітурієнтів;

залучення джерел фінансування науково-дослідних робіт;

допомога співробітникам і студентам університету в пошуку необхідної університетської учбової та наукової інформації.

**Учбові додатки**

Можливості гіпертекста і мультимедіа роблять WWW вельми благодатною середою для створення розподілених повчальних систем. WWW надає можливості створення інтерактивних повчальних систем, в яких сервер може не тільки надавати інформацію користувачеві, але і вести з ним діалог.

Уявіть собі, наприклад, допомога по радіоелектроніці: на дисплеї виникає схема радіоприлада і пропонується указати на цій схемі несправність. Той, що навчається вказує за допомогою миші на певні контрольні точки схеми і йому видаються результати вимірювання електричного потенціалу в цих точках. Після цього той, що навчається вказує на несправний елемент. У разі помилки сервер висилає документ з необхідними поясненнями і пропонує повторити завдання, а у разі правильної відповіді - переходить до наступного завдання. За результатами такої “контрольної” може бути виставлена оцінка.

Видалений доступ до багатого учбового матеріалу відкриває принципово нові можливості самоосвіти і заочного навчання, а також істотно полегшує проведення контрольних і домашніх робіт у вищих і середніх учбових закладах.

У WWW можна знайти і “скачати” програми практично на всі випадки життя, від найпростіших утиліт до великих програмних комплексів. Переглядаючи опис якої-небудь програми в тій або іншій колекції, потрібно звертати увагу не тільки на функціональні можливості програми і на те, для якої комп'ютерної платформи вона призначена, але і на умови її поширення і використання (так звану ліцензію). Найчастіше зустрічаються наступні різновиди ліцензій:

Commercial software (“commercialware”) - комерційне програмне забезпечення. Передбачається, що користувач повинен придбати ліцензію у власників даної програми перед установкою її у себе на комп'ютері. Часто попадаються безкоштовні демонстраційні версії комерційних програм, у яких штучно відключаються деякі функціональні можливості або ставиться тимчасової “замок”, що не дає запускати дану програму після закінчення деякого проміжку часу після її установки.

Shareware - умовно-безкоштовне програмне забезпечення. Цей різновид схожий на commercialware, але вам дозволяється встановлювати і копіювати програму до її реєстрації у власників. Часто дається можливість безкоштовного використання програми протягом певного терміну, а також безкоштовної реєстрації при використанні в освітніх і інших некомерційних цілях. Іноді реєстрація взагалі не обов'язкова, але дає певні переваги, наприклад, технічну підтримку.

Freeware - безкоштовне програмне забезпечення. Від вас потрібна лише повага авторських прав розробника програми.

Як правило, програмне забезпечення для IBM PC-сумісних комп'ютерів під DOS або Microsoft Windows зберігається в архівах у вигляді файлів, які можуть самі установлюватися (з розширенням exe), що виконуються або у вигляді запакованих (“архівованих”) файлів в форматі, сумісному з PKZIP/PKUNZIP (з розширенням zip). EXE-файл, що виконується потрібно завантажити в яку-небудь директорію на вашому диску (краще пусту), запустити його і слідувати інструкціям по установці. Що стосується ZIP-файлів, то після їх повного завантаження на ваш жорсткий диск для коректного розпаковування під Windows 95 або Windows NT з відновленням структури каталогів і довгих імен файлів найкраще скористатися програмою WinZip.

Стара добра програма PKUNZIP, працююча під DOS, на жаль, не розуміє довгих імен файлів. Як правило, серед файлів, витягнутих з zip-файла, знаходиться файл setup.exe або install.exe. Запускайте його і далі слідуйте інструкціям. Незайво заздалегідь прочитати файли readme.txt і license.txt, також що звичайно входять в комплект постачання програмного забезпечення.

Крім PKZIP/UNZIP-сумісних архіватор в світі Internet при пересилці файлів широко використовуються архіватор gzip і compress, що прийшли з операційної системи UNIX. Стислі цими програмами файли мають розширення .gz і .Z відповідно. На відміну від PKZIP, який може в один стислий файл вмістити декілька файлів або навіть підкаталогів, gzip і compress стискають файли тільки “по одному”. Для зберігання декількох файлів в одному стислому цими програмами файлі спочатку використовують програму tar, яка упаковує декілька файлів в один, а потім вже стискають цей файл програмами gzip або compress. У результаті отримують файл з подвійним суфіксом. tar.gz або. tar.Z. Вищезазначена програма WinZip - честь і хвала її авторам! - розуміє і ці формати.

**Особисті “візитні картки”**

Це вельми зручне - мати свою “візитну картку”, часто звану “homepage”, на WWW-сервері.

Уявіть собі, що ви посилаєте лист по електронній пошті на яку-небудь фірму, в університет або просто можливому другові по переписці. Досить при цьому повідомити адресату URL вашої “візитної картки”, і він зможе знайти там будь-яку інформацію про вас, яку ви тільки побажаєте повідомити. Звичайно така “картка” містить фотографії, резюме, список публікацій і поштову адресу, а також розказує про друзів, сім'ю і захоплення. Однак, багато хто схильний так “прикрашати” свої візитні картки і перевантажувати їх безліччю посилань на інші документи і абсолютно зайвими подробицями, що такі “шедеври”, крім їх авторів, ніхто не читати не буде.

**Віртуальні бібліотеки, виставки і картинні галереї**

WWW дає непогану можливість залучитися до скарбів світової літератури і мистецтва. Зрозуміло, “пробіжка” по віртуальних виставках ніяк не замінює відвідування виставок і музеїв, де розміщені самі оригінали, як не замінює його перегортання навіть шикарно виданих на папері альбомів і каталогів.

На жаль, технічні можливості, доступні в Україні, поки явно недостатні для того, щоб натішитися багатою палітрою фарб або HiFi-звуком. Репродукції картин, які виглядають прекрасно на величезному високоякісному моніторі робочої станції Silicon Graphics або навіть на IBM-сумісної “персоналці” з відеоадаптер SVGA і пристойним дисплеєм, на 386-ой PC з VGA і тайваньським монітором більше нагадують розмазані по екрану брудні плями, чим самі оригінали. Що стосується аудіо можливостей, то багато які українські комп'ютери так і не навчилися розмовляти. Проте, парк комп'ютерів в Україні стрімко оновлюється, і на перший план виходить інша проблема: пересилка графічної і звукової інформації займає так багато часу при найбільш поширеному в Україні модемному підключенні до Internet, що може вивести з себе навіть самого терплячого шанувальника мистецтва.

**WWW і комерція**

Останнім часом керівництво багатьох комерційних компаній усвідомило, що WWW - це справжня золота жила, що обіцяє величезні прибутки при належному використанні. Можливості реклами і продаж продукції, що надаються WWW, потенційно перевершують аналогічні можливості преси, телебачення і радіомовлень. Особливо зручна робота в WWW для фірм, що виробляють програмні продукти, для яких всі основні стадії - реклама, продаж і навіть “відвантаження готової продукції” - доступні через WWW. З жалем потрібно зазначити, що в Україні подібна діяльність поки вельми утруднена по двох основних причинах:

1. Дуже мало клієнтів, що мають доступ до WWW;

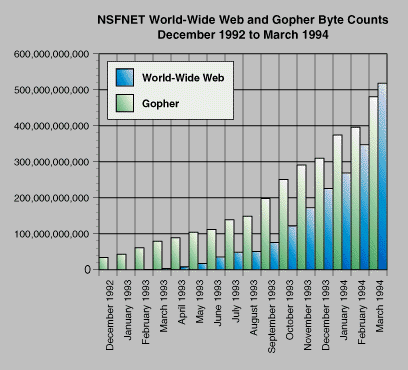
2. Недостатньо розвинені механізми оплати за допомогою кредитних карт або банківських чеків, найбільш зручні при даному методі оформлення замовлень.

**Дозвілля і захоплення**

Незважаючи на те, що спочатку WWW призначалася для “серйозних” цілей, в даний момент переважна частина інформації, що знаходиться там відноситься саме до сфери дозвілля і захоплень. Пошук і публікація в WWW матеріалів про ваші захоплення - стояча справа. Таким способом ви знайдете багато друзів і однодумців у всьому світі і встановите масу корисних особистих контактів.

1.8. Наскільки популярний Web?

З січня по грудень 1993, обсяг інформації, що передається по NSF зріс в 187 раз. У Грудні 1993 WWW займав вже 11 місце по кількості трафіка, а роком раніше він був всього лише 127.



Малюнок 2. Розвиток WWW.

У Червні 1993, Matthew Gray в MIT написав маленьку програму, яка подорожувала по всій мережі Web і визначала кількість запитів на отримання інформації від WWW. Маленька “мандрівниця по WWW” нарахувала за місяць біля 100 запитів, на які абонентами було отримано більш двохсот тисяч документів. У Березні 1994 кількість запитів перевищила вже 1200. Звичайно, програма згодом удосконалювалася, і ряд чинників міг впливати на останній результат, але незважаючи на це, сміливо можна затверджувати, що популярність WWW у 1994 в порівнянні із 1993 значно зросла.

Розділ II.

Мова програмування HTML

Можна працювати на Web без знання мови HTML, оскільки тексти HTML можуть створюватися різними спеціальними редакторами і конвертерами. Але писати безпосередньо на HTML неважко. Можливо, це навіть легше, ніж вивчати HTML-редактор або конвертер, які часто обмежені в своїх можливостях, містять помилки або проводять поганий HTML код, який не працює на різних платформах.

Мова HTML існує в декількох варіантах і продовжує розвиватися, але конструкції HTML ймовірніше усього будуть використовуватися і надалі. Вивчаючи HTML і пізнаючи його глибше, створюючи документ на початку вивчення HTML і розширюючи його наскільки це можливо, ми маємо можливість створювати документи, які можуть бути переглянені багатьма броузерами Web, як зараз, так і в майбутньому. Це не виключає можливості використання інших методів, наприклад, метод розширених можливостей, що надається Netscape Navigator, Internet Explorer або деякими іншими програмами. Якщо це дійсно служить Вашим цілям і Ви хочете сформувати власну думку про названі програми, користуйтеся ним. Але робота з HTML - це спосіб засвоїти особливості створення документів в стандартизованій мові, використовуючи розширення, тільки коли це дійсно необхідно.

HTML був ратифікований World Wide Web Consortium. Він підтримується декількома широко поширеними броузерами, і, можливо, стане основою майже всього програмного забезпечення, що має відношення до Web.

2.1. Структура HTML документа

Символи, взяті в кутові дужки <> є HTML командами, по яким “броузер” розпізнає, як потрібно перетворити частини тексту, укладені між цими командами.

Документ загалом повинен бути відмічений як документ в форматі HTML. Для цього він повинен починатися командою <HTML> і закінчуватися командою </HTML>.

Документ складається з 2 частин:

- Заголовка (Head),

- Власне документа (Body).

Для виділення заголовка потрібно ввести: <HEAD> Заголовок документа <HEAD>

Кожний WWW - документ має назву, яка вводиться в титульному рядку броузера.

Для введення титульного рядка в заголовок документа потрібно скористатися наступними командами: <HEAD> <TITLE> TITLE List </TITLE> </HEAD>

Потрібно зазначити, що титульний рядок повинен бути на англійській мові в латинському кодуванні оскільки вона відображається в спеціальних полях броузера.

Для запису основного тексту потрібно ввести: <BODY> Основний текст </BODY>

Таким чином, загальна схема документа в форматі HTML виглядає таким чином:

<HTML>

<HEAD> <TITLE> Титульний рядок документа </TITLE> </HEAD>

<BODY> Основний текст документа </BODY>

</HTML>

При написанні команд HTML не має значення, якими буквами - рядковими або прописними Ви пишете команди.

Створення заголовків

Заголовки в документі створюються за допомогою команд: <Hi> Заголовок </Hi>, причому при i = 1 заголовок самий великий, а при i = 6 - самий дрібний.

Наприклад,

|  |  |
| --- | --- |
| Текст, що Відображається | Запис в форматі HTML |
| **Заголовок1** | <H1> Заголовок1 </H1> |
| **Заголовок2** | <H2> Заголовок2 </H2> |
| **Заголовок3** | <H3> Заголовок3 </H3> |
| **Заголовок4** | <H4> Заголовок4 </H4> |
| **Заголовок5** | <H5> Заголовок5 </H5> |
| Заголовок6 | <H6> Заголовок6 </H6> |

2.3. Створення списків

Списки призначені для представлення інформації у впорядкованому вигляді.

У HTML - документах використовується 3 види списків:

1. Неупорядковані списки,

2. Упорядковані списки,

3. Списки - визначення.

**Неупорядкований список** використовує для виділення записів спеціальні символи. Неупорядкований список визначається наступними командами початку і кінця списку: <UL> і <UL>

Кожний елемент списку починається з команди: <LI>

У команді <LI> може бути параметр TYPE, який визначає зовнішній вигляд символа, що використовується для виділення рядка в списку і може мати наступні значення:

<LI TYPE=DISK> - кругла жирна точка,

<LI TYPE=CIRCLE> - коло,

<LI TYPE=SQUARE> - маленький чорний квадрат.

В **упорядкованому списку** всі записи пронумеровані.

Упорядкований список визначається наступними командами початку і кінця списку: <OL> і </OL>

Кожний елемент списку починається з команди: <LI>

У команді <LI> може бути параметр TYPE, який визначає тип нумерації і може мати наступні значення:

<LI TYPE = А> - великими буквами,

<LI TYPE = а> - малими буквами,

<LI TYPE = I> - прописними римськими цифрами,

<LI TYPE = i> - рядковими римськими цифрами,

<LI TYPE = 1> - арабськими цифрами.

У **списку - визначенні** всі записи здвинуті ліворуч за допомогою табуляції.

Список - визначення визначається наступними командами початку і кінця списку: <DL> і </DL>

Кожний елемент списку починається з команди: <DD>

2.4. Форматування параграфів і виведення в декілька стовпців.

У документі HTML неможливо створити абзац, використовуючи клавішу [Enter].

Використання цієї клавіші поліпшує зовнішній вигляд початкового тексту, але не впливає на отримане зображення.

Для переходу на наступний рядок скористайтеся командою <br>

Для створення пустого рядка скористайтеся командою <р>

Для створення горизонтальної смуги скористайтеся командою <hr>

Якщо смуга створюється не на весь рядок, а на якусь її частину, наприклад, на 40%, то команда виглядає таким чином: <hr width= 40 %>

Для створення широкої смуги потрібно задати її ширину (в пікселях) <hr size= 10 >

Для вирівнювання смуги ЛІВОРУЧ/ПРАВОРУЧ/ПО ЦЕНТРУ задайте відповідний параметр ALIGN=LEFT/RIGHT/CENTER

Чорна лінія задається наявністю параметра NOSHADE.

Ви можете розташувати текст по центру за допомогою наступної команди: <CENTER>

Іноді необхідно створити текстовий документ, розташування рядків якого (табуляція, відступи, вирівнювання) визначається розробником домашньої сторінки, а не програмою-редактором.

Для створення такого тексту скористайтеся командами: <PRE> текст </PRE>

Отриманий в броузері образ тексту буде точно відповідати тексту в початковому HTML файлі.

Аналогічні функції виконує пара команд: <LISTING> текст </LISTING>

Netscape 3.0 і вище підтримує виведення в декілька стовпців, в газетному форматі за допомогою команд <MULTICOL> і </MULTICOL>.

Нижче приводиться формат команд:

<MULTICOL COLS= “N” GUTTER= “M”> текст </MULTICOL>.

Тут N - кількість стовпців, M - відстань між стовпцями.

Броузери, які не підтримують виведення в декілька стовпців, забезпечать виведення нормально, в 1 стовпець.

2.5. Форматування символів

Для виділення окремих частин тексту можна скористатися наступними стилями:

У - жирний (Bold);

I - курсив (Italic);

U - з підкресленням (Underline);

S - з перекресленням (STRIKE);

Нижче наведена таблиця команд виділення тексту.

|  |  |
| --- | --- |
| Текст, що відображається | Запис в форматі HTML |
| Виділення **тексту** жирним шрифтом | Виділення <b> тексту</b> жирним шрифтом |
| Виділення *тексту* курсивом | Виділення <i> тексту</i> курсивом |
| Виділення тексту підкресленням | Виділення <u> тексту </u> підкресленням |
| Виділення ~~тексту~~ перекресленням | Виділення <strike> тексту</strike> перекресленням |

Для зміни розміру текстових символів скористайтеся командами:

<FONT SIZE = +-i > </FONT>, де i в межах від 1 до 7 відносно початкового розміру букв.

Крім того, можна збільшити або зменшити висоту шрифту за допомогою команд <BIG> і <SMALL>. Ці команди використовуються в парі з командами </BIG> і </SMALL>.

Ще дві пари команд:

<SUB> і </SUB> - підрядковий індекс, наприклад, H2SO4.

<SUP> і </SUP> - надрядковий індекс, наприклад, (a2 - b2) = (а - b)(а + b).

Всі вищенаведені команди визначають фізичне форматування символів. Ви задаєте конкретну зміну характеристик символів.

Крім того, є логічне форматування символів - Ви даєте опис, а броузер вирішує як змінювати характеристики. Загалом, хто довіряє броузеру - користуйтеся цими командами. Вони виглядають таким чином:

|  |  |  |
| --- | --- | --- |
| <CITE> | Цитата | <CITE> |
| <EM> | Особливо важливий текст | </EM> |
| <STRONG> | Сильне виділення тексту | </STRONG> |
| <KBD> | Текст, введений користувачем | </KBD> |
| <CODE> | Лістинг програми | </CODE> |
| <SAMP> | Послідовність літералів | </SAMP> |
| <VAR> | Ім'я змінної | </VAR> |

Броузер користувача сам вибирає шрифт, якщо Ви його конкретно не задали в описі сторінки. Якщо в сторінці заданий шрифт, якого немає у користувача, то Ви можете задати список, що складається з декількох шрифтів, і броузер вибере самий лівий, а якщо його в системі немає, то наступний і т. д. Якщо жодного з шрифтів немає, броузер вибирає свій шрифт.

Шрифт задається за допомогою команди: FONT FACE

2.6. Управління кольором

Кольори і зображення фону задаються за допомогою команди <BODY>.

Ця команда може задаватися тільки на початку HTML файла і не може бути змінена надалі. Її параметри:

bgcolor = “# код кольору” - колір фону документа,

text = “# код кольору” - колір тексту документа,

link = “# код кольору” - колір тексту, що використовується як посилання,

vlink = “# код кольору” - колір посилання на переглянутий раніше документ,

alink = “# код кольору” - колір посилання в момент натиснення на неї правої кнопки миші.

Код кольору задається в кодуванні RGB - шість шістнадцяткових чисел.

Важливо зазначити, що колір фону не відображається на папері, якщо розпечатати HTML-документ. Нижче приводяться коди найпростіших кольорів:

|  |  |
| --- | --- |
| Білий = FFFFFF | Малиновий = FF00FF |
| Жовтий = FFFF00 | Бірюзовий = 008080 |
| Червоний = FF0000 | Темно-синій = 000080 |
| Сірий = 808080 | Коричневий = 808000 |
| Синій = 0000FF | Блакитний = 00FFFF |
| Зелений = 00FF00 | Темно-зелений = 008000, |
| Чорний = 000000 | Індиго = 800080 |
| Пурпурний = CC33FF | Темно-червоний = 800000 |

Для зміни кольору поточного тексту можна в будь-якому місці домашньої сторінки скористатися командами: <FONT COLOR=Код кольору> Поточний текст </FONT COLOR>

Визначено декілька іменованих кольорів. Їх значення ми бачимо в таблиці:

<FONT COLOR = black>

Якщо Ви хочете створити “шпалери”, тобто кольоровий фон з використанням Image-файла з ім'ям, наприклад, Ground.gif, то замість параметра BGCOLOR записуємо: BACKGROUND = “Ground.gif”

Якщо при цьому ви хочете, щоб фонове зображення не рухалося, тобто щоб створився ефект “водяних знаків”, то задайте додатковий параметр:

BACKGROUND = “Ground.gif” BGPROPERTIES=FIXED

2.7. Використання спеціальних символів

У мові HTML символ < використовується як перший символ кожної команди.

Цей символ не може бути використаний для позначення відношення “менше”. Для відображення на екрані символа < (“менше”) і деяких інших символів в HTML прийняті спеціальні позначення:

|  |  |
| --- | --- |
| Текст, що відображається | Запис в форматі HTML |
| < | &lt |
| > | &gt |
| & | &amp |
| “ | &quot |

Мова HTML завжди зменшує кількість підряд розташованих пропусків до одного, незалежно від того, скільки пропусків було в початковому тексті. Для того, щоб задати необхідну кількість підряд розташованих пропусків потрібно скористатися спеціальним символом пропуску: &nbsp.

Наприклад, в наступному записі початкового тексту: (А &nbsp;&nbsp;&nbsp; У) в тексті, що відображається між символами А і В буде 3 пропуски.

При необхідності зобразити на екрані команду мови HTML виникає додаткова проблема - редактор може перетворити її у відповідну команду прямо в початковому тексті.

І замість відображення відповідної команди на екрані Ви отримуєте розташований по центру текст в наступних рядках. Щоб цього уникнути, можна, наприклад, вставити всередину команди дві парні команди. Наприклад, вищенаведену команду потрібно записати таким чином:<<u></u>CENTER>

2.8. Створення таблиць

Таблиця створюється за допомогою команд <TABLE> і </TABLE>.

Команда <TABLE> може мати наступні параметри:

- UNITS - може приймати значення RELATIVE або PIXELS. Визначає одиниці вимірювання в інших параметрах. По замовчанню, UNITS = PIXELS.

- BORDER - визначає лінії, що розмежовують клітки в таблиці. Ширина лінії задається командою BORDER = N.

- CELLPADDING - Визначає мінімальний проміжок навколо вмісту таблиці.

- BGCOLOR - Визначає колір фонового зображення в таблиці.

Є ще декілька команд, задаючих колір обрамлення: BORDERCOLOR, BORDERCOLORDARK, BORDERCOLORLIGHT, але вони використовуються тільки в Microsoft Internet Explorer.

Для формування таблиці, що складається з декількох рядків, використовують команду <TR>, що розділяє рядки. Команди <TH> і </TH> використовуються для позначення заголовка стовпця кліток.

Параметри ROWSPAN і COLSPAN команд <TD> і <TH> використовуються для формування кліток даних, що об'єднують більше за один рядок або стовпець.

2.9. Вбудування гіпертекстових посилань

**Перехід в інше місце того ж документа.**

Для організації переходу всередині одного HTML - файла потрібно ввести необхідний текст в двох місцях:

- Записати мітку в тому місці HTML - файла, куди необхідно перейти.

- Записати перехід на цю мітку в тому місці, звідки буде виконаний перехід.

Для організації мітки записують: <А NAME = “МІТКА” ></А>, де МІТКА - будь-яке ім'я, що складається з букв англійського алфавіта і цифр.

Для переходу на цю мітку запишіть:<А href=МІТКА >Перейти на мітку</А>.

**Перехід в інший документ тієї ж сторінки.**

Для переходу з будь-якого місця HTML-файла h1.html в початок HTML-файла h2.html потрібно записати: <А href = “h2.html”>Перехід до розділу h2</А>.

**Перехід в інший документ.**

У будь-якому місці HTML-файла можна ввести гіпертекстові посилання на інший документ або файл.

Гіпертекстові посилання вводяться в HTML-файл за допомогою уніфікованого локатора ресурсів - URL.

Цей локатор визначає правила написання різних видів посилань.

Нижче приводяться найважливіші префікси для URL:

|  |  |
| --- | --- |
| Префікс URL | Функція |
| FTP:// | Посилання на сервер FTP |
| HTTP:// | Посилання на об'єкт, який буде передаватися з використанням протоколу HTTP |
| FILE://localhost | Посилання на локальний диск |
| GOPHER:// | Посилання на сервер Gopher |
| MAILTO:// | Посилання на електронну поштову адресу |
| NEWS:// | Посилання на електронну конференцію |
| NNTP:// | Посилання на сервер електронної конференції |
| TELNET:// | Посилання на сервер електронної конференції |

Для запису гіпертекстового посилання в HTML-файл використовують наступну команду: <А href = “URL”>Перехід по гіпертекстовому посиланню</А>.

При натисненні лівої кнопки миші на текст Перехід по гіпертекстовому посиланню Ви перейдете на ресурс, заданий в URL. Для повернення скористайтеся клавішею “Back” броузера.

**Виведення на екран.**

Зображення можуть бути виведені на екран так само, як текст. Файли зображення можуть бути задані в декількох різних форматах, найкращі з яких - GIF (файли \*.gif) і JPEG (файли \*.jpg).

Нижче приводиться формат команди:

<IMG SRC= “ІМ'Я ФАЙЛА” ALT = “Текст” ALIGN = DIRECTION WIDTH = WIDTH HEIGHT = HEIGHT >, де:

- ІМ'Я ФАЙЛА - це ім'я даного файла у вашій сторінці або повне ім'я шляху і файла, якщо файл не належить даній сторінці.

- Текст - це текст, що виводиться замість зображення, якщо файл зображення по якийсь причині недоступний.

- DIRECTION - місце розташування зображення. Приймає наступні можливі значення:

ALIGN=TOP - подальший текст розташовується у верхній частині зображення.

ALIGN=BOTTOM - подальший текст розташовується в нижній частині зображення.

ALIGN=LEFT - зображення розташовується в лівій частині листа. Текст обтікає зображення праворуч.

ALIGN=MIDDLE - зображення розташовується в центрі листа.

ALIGN=RIGHT - зображення розташовується в правій частині листа. Текст обтікає зображення ліворуч.

WIDTH = WIDTH - ширина необхідного зображення на екрані,

HEIGHT = HEIGHT - висота необхідного зображення на екрані.

**Гіпертекстові посилання за допомогою зображення.**

Допустимо, що Ви виводите на екран зображення IMAGE-файл, який називається, наприклад, image.gif.

Якщо Ви хочете організувати сторінку так, щоб при натисненні лівої кнопки миші на зображення, перейти на іншу домашню сторінку, наприклад, на:

HTTP://WWW.HOME.COM,

потрібно записати наступну команду:

<А href= “HTTP://WWW.HOME.COM”> <IMG SRC = “image.gif”>

**Створення посилання на поштову адресу.**

Наприклад моя поштова адреса в GEOCITIES.COM: info@geocities.com

Для організації посилання в моїй домашній сторінці на мою поштову адресу необхідно записати в кінці головного розділу домашньої сторінки:

<А href = mailto: info@geocities.com > Перейти на мою поштову адресу </А>

У наступній строчці необхідно задати таке ж посилання з допомогою аплікації MAIL12.GIF замість тексту:

<А href = mailto: info@geocities.com ><IMS SRC = “MAIL12.GIF” ></А>

2.10. Кадри.

**Команди роботи з кадрами.**

Ви можете розділити екран на декілька незалежних віконних кадрів, кожний з яких відображає окремий HTML документ.

На екрані кадри являють собою прямокутники. У HTML використовуються наступні команди роботи з кадрами:

<FRAMESET>, </FRAMESET> - визначають склад і розміри кадрів на екрані,

<FRAME>, </FRAME> - визначають HTML-файл для кожного кадру,

<NOFRAMES>, </NOFRAMES> - для повідомлень броузеру, що не обробляє кадри.

**Формат команди FRAMESET.**

Команда ділить ціле вікно (або частину вікна, визначену попередньою командою FRAMESET) на декілька вертикальних (параметр COLS) або горизонтальних (параметр ROWS) кадрів. Кожний з цих кадрів може визначати HTML-файл, що відображається в ньому (за допомогою команди FRAME) або, відповідно, ділитися далі за тими ж правилами з вкладеною командою FRAMESET.

По кінцю визначення кадрів, що задаються командою FRAMESET, необхідно записати команду /FRAMESET, інакше вони можуть бути побудовані неправильно.

*Формат параметрів COLS і ROWS.*

Ці параметри дозволяють визначити розміри і кількість побудованих вертикально (параметр COLS) або горизонтально (параметр ROWS) кадрів.

Атрибути параметрів задаються в одній з 3 форм:

- абсолютний розмір в пікселях,

- заданий процент в загальній довжині або ширині,

- залишок після завдання попередніх кадрів.

Символ “\*” означає виділення кадру площі, що залишилася. Наприклад, команда: <FRAMESET ROWS= “100, \*”> створює 2 кадри: верхній становить 100 пікселів у висоту, нижній - простір, що залишився.

**Формат команди FRAME.**

Дана команда існує тільки всередині блоку FRAMESET - /FRAMESET. Її призначення - визначення функцій конкретного кадру.

Команда /FRAME практично не використовується так як її функції виконують команди /FRAMESET і наступна FRAME.

Параметр SRC задає ім'я HTML - файла, що відображається в даному кадрі.

Параметр NAME застосовується для створення імені кадру.

Тоді інший кадр може посилатися на даний для відображення в ньому свого гіпертекста.

Це дуже поширена команда її можна бачити в багатьох сторінках. Її перевага в тому, що на екрані можна одночасно бачити і HTML-файл, що викликає, і HTML-файл, що викликається, і для перегляду наступного файла гіпертекста досить натиснути на відповідну кнопку файла, який викликає.

Параметр TARGET команди BASE задає кадр, що виділяється по замовчанню для відображення гіпертекста.

Параметр NORESIZE застосовується для того, щоб користувач не зміг змінити розміри даного кадру на екрані.

Лінійки прокрутки задаються за допомогою параметра SCROLLING.

Тут діють наступні правила:

- Якщо параметр SCROLLING не заданий, то лінійки прокрутки створюються автоматично тільки тоді, коли розмір тексту, що відображається перевищує розмір кадру (це відноситься як до розміру по вертикалі, так і по горизонталі).

- Якщо задане SCROLLING = “YES”, то лінійки прокрутки створюються завжди.

- Якщо задане SCROLLING = “NO”, то лінійки прокрутки не створюються.

- Якщо задане SCROLLING = “AUTO”, то система працює так само, як якщо параметр SCROLLING не заданий.

Товщина граничної лінії між кадрами може задаватися в пікселях за допомогою параметрів MARGINHEIGHT і MARGINWIDTH, якщо розробника не влаштовують значення, задані по замовчанню.

*Спеціальні значення параметра TARGET.*

У сучасних броузерах зарезервовано 4 спеціальних значення параметра TARGET:

а) \_blank - броузери завжди задає це значення для нового, неіменованого вікна.

б) \_self - задається по замовчанню для всіх кадрів, в операторі FRAME яких немає параметра TARGET. Всі документи, що викликаються по гіперпосиланню, завантажуються і відображаються в тому ж кадрі, що і документ, який їх викликає.

Значення \_self в явному вигляді не застосовується доти, поки ми не вводимо команду BASE з параметром TARGET - тоді, можливо, знадобиться відмінити це загальне призначення для якого-небудь конкретного кадру.

в) \_parent - файл, що викликається по гіперпосиланню, завантажується в кадр або вікно, яке є батьківським по відношенню до того, що викликає. Іншими словами, файл, що викликається буде розташований в тому ж вікні, що і його “дідусь”.

г) \_top - файл, що викликається по гіперпосиланню, завантажується у вікно, що містить посилання, і перекриває всі кадри даного вікна.

**Формат команди NOFRAME.**

Ця команда призначена для броузерів, що не обробляють кадри.

У броузері, що обробляє кадри, вся інформація, що міститься між командами NOFRAME і /NOFRAME, ігнорується.

У броузері, що не обробляє кадри, ця інформація обробляється.

Звичайно розробники сторінок, використовуючі кадри, вставляють в тіло команди NOFRAME повідомлення про те, що даний броузер не обробляє кадри і повідомляє рекомендації, наприклад, перейти на Netscape 2.0 і вище.

2.11. Спеціальні ефекти

**Створення блимаючого тексту.**

Зауваження: працює в Netscape 3.0, не працює в Explorer 3.0

Ви можете створити зображення тексту що блимає за допомогою наступних команд: <blink> і </blink>

Не зловживайте блимаючим зображенням. Його краще усього застосовувати для залучення уваги до нової інформації в домашній сторінці.

**Створення рядка, що біжить.**

Зауваження: працює в Explorer 3.0, не працює в Netscape 3.0

Рядок, що біжить створюється за допомогою команд: <MARQUEE> і <MARQUEE>. Нижче приводяться приклади команд:

Команда <MARQUEE> </MARQUEE> - цей текст буде рухатися з права на ліво повільно.

Команда <MARQUEE BEHAVIOR = “ALTERNATE” CROLLAMOUNT = “50” SCROLLDELAY = “8” > </MARQUEE> - цей текст буде рухатися з права на ліво і назад.

Команда <MARQUEE BEHAVIOR= “SLIDE” > </MARQUEE> - цей текст буде рухатися і зупинятися.

Команда <MARQUEE SCROLLDELAY= “5” SCROLLAMOUNT = “50”> </MARQUEE> - цей текст буде рухатися дуже швидко.

**Карта образів.**

Зауваження: працює в Explorer 3.0, працює в Netscape 3.0

Карта образів дозволяє створити декілька посилань до різних адрес, використовуючи тільки одне зображення.

Нехай у Вас є картинка imgmap.gif розміром 100 х 100 пікселів. Припустимо, що картинка складається з 4 частин, де зображені Ваші друзі:

- В лівому верхньому кутку - Діма (у нього є домашня сторінка dima.html )

- В правому верхньому кутку - Алік (Його домашня сторінка alik.html )

- В лівому нижньому кутку - Вова (Його домашня сторінка vova.html )

- В правому нижньому кутку - Ігор (Його домашня сторінка igor.html )

Якщо Ви хочете, щоб при натисненні лівої кнопки миші на зображення кожного з Ваших друзів Ви попадали на його домашню сторінку, запишіть наступні команди:

<map name= “map ”>

<area shape = “rect” coords = “0, 0, 49, 49” href = “dima.html”>

<area shape = “rect” coords = “0, 50, 49, 99” href = “alik.html”>

<area shape = “rect” coords = “50, 0, 99, 49” href = “vova.html”>

<area shape = “rect” coords= “50, 50, 99, 99” href = “igor.html”>

</mар>

<img src = “imgmap.gif” usemap = “# map” border= 0>

**Плаваючі вікна.**

Зауваження: працює в Explorer 3.0, не працює в Netscape 3.0.

Це вікно створюється за допомогою наступної команди:

<IFRAME NAME = “content\_frame” width = “400” height = “240” SRC = “index.html ”> Це плаваюче вікно </IFRAME>

2.12. Форми

Форма - це засіб, що дозволяє організувати в сторінці діалог з її користувачем.

Розробник сторінки створює її за допомогою спеціальних елементів:

- кнопки,

- списки,

- текстові рядки,

- текстові поля.

Форми передають інформацію програмі обробки у вигляді пар:

Ім'я поля - Значення поля.

Форма починається командою <FORM> і закінчується командою </FORM>.

Дана команда має 2 основних параметра: ACTION і METHOD.

Параметр ACTION є обов'язковим. Він вказує адресу обробника форми (наприклад, адреса програми обробки або поштова адреса або просто адреса URL для переходу).

Параметр METHOD не є обов'язковим. Він приймає одне з двох значень: GET або POST, пов'язаних з методом протоколу передачі інформації з анкети.

По замовчанню, METHOD= GET.

Приклад команди FORM:

<FORM ACTION = “http://www.aport.ru” METHOD= POST >

Для завдання керуючої кнопки запуску передачі даних треба записати наступну команду:

<INPUT TYPE = submit NAME = “botton1” VALUE = “Довільний текст”>

де:

INPUT - команда створення керуючого елемента.

TYPE - параметр, що визначає керуючий елемент.

submit - значення цього параметра (в цьому випадку керуюча кнопка).

NAME - параметр, задаючий ім'я змінної.

botton1 - ім'я змінної.

VALUE - параметр, задаючий текст, що відображається на керуючому елементі.

Довільний текст - значення тексту на керуючому елементі.

Нижче приводяться основні типи керуючих елементів INPUT.

1) Текстовий рядок: TYPE=text визначає вікно для введення одного рядка тексту.

Параметр SIZE задає розмір вікна в символах.

Приклад:

<INPUT TYPE = text NAME= “text1” SIZE = 30 VALUE = “Введіть текст”>

У отримане вікно, в якому буде зображатися текст: Введіть текст.

Ви можете ввести будь-який текст або залишити записаний початковий текст “Введіть текст” без змін.

При натисненні кнопки submit в програму обробки буде передано:

text1=Ввести текст (або інший, змінений текст).

Параметр SIZE задає розмір вікна в символах.

При необхідності обмежити довжину тексту N, застосовуйте додатковий параметр: MAXLENGTH= N.

2) Вікно для введення пароля: TYPE=password

Аналогічний типу text, але при введенні кожний символ замінюється символом \*.

Приклад. Треба ввести пароль довжиною не більше за 5 символів.

<INPUT TYPE=password NAME= pass size= 10 maxlength= 5 >

3) Текстове вікно: TEXTAREA

Задається в наступному вигляді: <TEXTAREA NAME= text3 COLS= 30 ROWS= 5 WRAP=VIRTUAL> Спочатку заданий текст </TEXTAREA>

Параметр COLS задає кількість символів в рядку, параметр ROWS задає кількість рядків у вікні, необов'язковий параметр WRAP=VIRTUAL задає лінійку прокрутки.

4) Радіокнопка: TYPE=radio

Може бути задано декілька радіокнопок з однаковим ім'ям (тобто значенням параметра NAME), але вони будуть взаємовиключаючими: може бути натиснена (CHECKED) тільки одна з них.

Приклад:

<INPUT TYPE = radio NAME = color VALUE = “Червона” checked> Червона

<INPUT TYPE = radio NAME = color VALUE = “Синя” > Синя

<INPUT TYPE = radio NAME = color VALUE = “Зелена” > Зелена

Ми отримали 3 кнопки, з них відмічена Червона. При натисненні на іншу кнопку попередня звільняється.

Допустимо, ми натиснули Синю кнопку, тоді при натисненні на кнопку submit в програму обробки буде введено:

color=Синій

5) Помічений квадрат: TYPE=checkbox

На відміну від радіокнопок, може бути декілька квадратів з однаковими іменами.

Наприклад, в наступному прикладі спочатку задана кнопка “Горілка”:

<INPUT TYPE = checkbox NAME = box VALUE = “Горілка” checked> Горілка

<INPUT TYPE = checkbox NAME = box VALUE = “Коньяк” > Коньяк

<INPUT TYPE = checkbox NAME = box VALUE = “Лікер” > Лікер

Якщо Ви натиснете ще на кнопку “Коньяк”, то при натисненні на кнопку submit в програму обробки буде введено:

box=Горілка

box=Коньяк

Прихований текст: TYPE=hidden

При відображенні не видний користувачеві. Служить для прямої передачі службової інформації від домашньої сторінки обробнику форми, прозоро для користувача.

7) Меню: SELECT

Нехай Вам представилася можливість вибрати одну з путівок у відповідності з наступним списком:

<SELECT NAME= “script”>

<OPTION VALUE = “tour1” > Путівка в Сочи

<OPTION VALUE = “tour2” > Путівка в Париж

<OPTION VALUE = “tour3” > Путівка на Колиму

Якщо Ви вибрали Путівку в Париж, то при натисненні на кнопку submit в програму обробки буде введено: script = tour2

Команда SELECT може містити атрибут MULTIPLE, при якому можна вибрати декілька елементів меню одночасно.

Висоту вікна можна задати атрибутом SIZE= N.

Якщо в рядку OPTION заданий атрибут checked, то даний елемент задається по замовчанню.

8) Кнопка введення: TYPE=submit

Це і є та сама кнопка, про яку ми так часто згадували.

<INPUT TYPE = submit VALUE = “Введення даних” >

9) Кнопка скасування: TYPE = reset

Кнопка виконує ініціалізацію початковими значеннями всіх полів.

<INPUT TYPE=reset VALUE= “Очистити поля” >

2.13. Команди META

Створивши сторінку, Ви, безсумнівно, захочете щоб Вас змогли “знайти” всі користувачі, що цікавляться даним питанням.

Тут можна йти двома основними шляхами:

1. “Відмітитися” в пошукових системах. Є служби, що дозволяють зареєструватися на декількох серверах одночасно.

2. Вставити всередині блоку <HEAD>...</HEAD> команди META.

Основне призначення команд META - інструкції для пошукових машин про тематику Вашої сторінки.

Для цих цілей використовуються наступні параметри:

- description - короткий опис сторінки,

- keyword - список ключових слів в сторінці.

З інших застосувань команд META важливо привести команди, що дають інструкції броузеру про кодування Вашої сторінки.

<META HTTP-EQUIV = “Content-type” CONTENT = “text/html; charset=windows-1251”> для кодування в Windows/1251,

<META HTTP-EQUIV = “Content-type” CONTENT = “text/html; charset=koi8-r”> для кодування в KOI-8.