

Ви вже переконалися, що за допомогою автоматизованих засобів, які надають у розпорядження користувача служби безкоштовного хостинга, створювати й адмініструвати веб-сайти надзвичайно просто. Крім того, це не потребує особливих знань із веб-дизайну. Однак у такому випадку ваші можливості обмежені, і ви не зможете створити сайт за своїми уподобаннями. Щоб мати змогу це зробити, потрібно опанувати спеціальну мову, призначену для розробки веб-сторінок. Тоді перед вами відкриються захоплюючі можливості щодо створення свого власного віртуального світу!

У цьому розділі ви ознайомитеся із принципами створення веб-сторінок за допомогою мови HTML, навчитеся визначати структуру сторінок, розміщувати і формувати текст, працювати з таблицями та фреймами, пов'язувати створені файли за допомогою гіперпосилань. Виконавши дві запропоновані практичні роботи, ви власноруч створите веб-сторінки — можливо, перші у вашому житті.

Основною перевагою та особливістю веб-сторінок є те, що інформація, розміщена на них, має вигляд гіпертексту. *Гіпертекст* — це текст, у який вбудовані спеціальні коди, що задають форматування тексту, наявність у ньому ілюстрацій, мультимедійних вставок та гіперпосилань. Ці коди визначені у мові програмування *HTML* (Hyper Text Markup Language — мова гіпертекстової розмітки). Тобто веб-сторінка фактично є документом у форматі HTML. Процес вставляння в текст кодів HTML називають *розміткою*. Браузери підтримують мову HTML і під час завантаження веб-сторінки читають наявні у ній коди та виконують задані за їх допомогою операції (форматують текст, вставляють у нього зображення

тощо). Таким чином, вигляд сторінки у вікні браузера залежить від того, які коди містяться у її документі.

Поняття тегу

Коди мови HTML, за допомогою яких розмічають вихідний текст, називають *тегами*. Під час відображення документа в браузері самих тегів не видно, але вони впливають на зовнішній вигляд документа. Усі теги починаються із символу < і закінчуються символом > — їх називають *кутовими дужками*. Після відкритої кутової дужки розміщують ключове слово, яке вказує на призначення тегу. Регістр у назвах тегів не має значення, хоча загальноживаними є великі літери. Приклади тегів HTML: <TITLE>, <BODY>, <TABLE>, , , </CENTER>.

Зазвичай тег впливає на певний фрагмент документа, наприклад на абзац. У таких випадках використовують *пару тегів* (відкривальний і закривальний). Перший із них задає ефект, а другий — припиняє його дію. Закривальний тег починається із символу / (слеш). Приклади парних тегів: <HTML></HTML>, , <HEAD></HEAD>, <H3></H3>, <ADDRESS></ADDRESS>, .

Деякі теги задають разовий ефект у місці своєї появи, тоді закривальний тег не потрібен. Приклади одинарних тегів:
, <HR>, <META>, <BASEFONT>, <FRAME>, <INPUT>.

Якщо між кутовими дужками помилково вказане ключове слово, яке відсутнє в мові HTML, то весь тег буде проігноровано.

Відкривальні теги можуть містити атрибути, які впливають на ефект, створюваний тегом. *Атрибути* — це додаткові ключові слова, відокремлені від основного ключового слова тегу та один від одного символами пробілу.

Для деяких атрибутів слід задавати значення, яке відділяють від імені атрибута символом =. Значення атрибута беруть у лапки, але в багатьох випадках їх можна і не ставити. Закривальні теги атрибутів не містять.

Приклади тегів з атрибутами:

```
<BODY BGCOLOR="#000000" TEXT="#FFFFFF"
BACKGROUND="rain.gif">
```

```
<OPTION SELECTED>  
<FRAME SCR="file.html" NORESIZE>
```

Для того щоб через деякий час ви або інша людина, яка переглядатиме код HTML, могла краще його зрозуміти, у текст вставляють примітки — *коментарі*. Вони починаються зі спеціального тегу `<! —`. Будь-який записаний після нього текст браузер розглядає як коментар і не відображає під час відтворення документа. Закінчують коментар тегом `—>`. Коментар може містити будь-які символи, крім `>`, а отже, не може включати в себе теги.

Структура HTML-документа

HTML-документ складається з основного тексту і тегів розмітки. Фактично це звичайний текстовий файл, для створення і редагування якого можна використати будь-який текстовий редактор. Зазвичай HTML-документи містяться у файлах із розширенням `.htm` або `.html`.

Документ HTML має чітко визначену структуру.

- Починається з тегу `<HTML>` і закінчується відповідним йому тегом `</HTML>`. Така пара тегів повідомляє браузеру, що це HTML-документ.
- Містить два розділи — заголовків і тіла документа (розміщуються саме в такому порядку). Розділ заголовків помічений тегами `<HEAD>` і `</HEAD>` та містить Інформацію про документ загалом. Зокрема, він повинен містити в собі теги `<TITLE>` та `</TITLE>`, між якими розміщено текст, що відобразиться в заголовку вікна браузера. Крім цього, у розділі заголовків може міститися тег `<META>`, призначений для технічного опису документа (це інформація для пошукових програм), а також тег `<STYLE>` для опису стилів (наборів параметрів форматування), використаних у документі.
- Сам текст документа міститься в його тілі, яке розташоване між тегами `<BODY>` та `</BODY>`.

Отже, основну структуру HTML-документа визначають чотири парні теги. Їхня наявність передбачена у всіх таких документах;

```
<HTML>  
<HEAD>  
<TITLE>Заголовок документа</TITLE>
```

```
</HEAD>
<BODY>
Текст, що відображається на екрані
</BODY>
</HTML>
```

Для того щоб розбити текст на логічні частини, використовують *заголовки*. Мова HTML підтримує шість рівнів заголовків документів. Вони позначені тегами від `<H1>...</H1>` до `<H6>...</H6>`. На екрані ці заголовки відображаються різними шрифтами (зазвичай напівжирними). Хоча в мові HTML є теги форматування, які можуть змінювати зображення шрифту, користуватися ними для заголовків не рекомендовано.

Текст, що міститься всередині тегу заголовка, відобразатиметься відповідно до його рівня. Найвищий рівень має заголовок H1, найнижчий — H6. Текст заголовка може бути вирівняний по центру, за правим або лівим краєм — для цього використовують атрибут ALIGN. Наприклад: `<H2 ALIGN=center \ right \ left>Текст заголовка </H2>` (тут символ \ означає можливість вибору між значеннями center, right або left).

На рис. 2.1 показано заголовки різних рівнів.

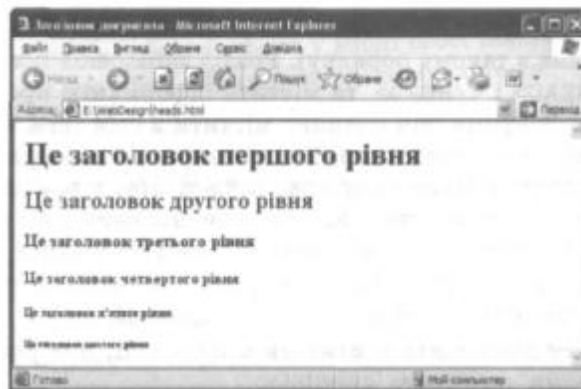


Рис. 2.1. Заголовки різних рівнів

Для визначення *звичайних абзаців* у мові HTML використовують теги `<P>` і `</P>`. Текст, розміщений між ними, власне і буде одним абзацом. При цьому слід пам'ятати, що всі «зайві» пропуски між

словами і переходи на новий рядок під час відтворення HTML-документа браузер ігнорує. Коли браузер відтворює веб-сторінки, абзаци відділяються один від одного за допомогою відступів.

У разі переходу на новий рядок без створення абзацу використовується одинарний тег `
`. Розділювачами в тексті можуть також бути горизонтальні лінії, які візуально відділяють різні частини документа, — їх створюють за допомогою одинарного тегу `<HR>`.

Кольори та зображення для всього документа і його тла задають за допомогою тегу `<BODY>`. Він може мати такі атрибути:

- `BACKGROUND="URL"` — замість URL вказують адресу малюнка, який має бути тлом для сторінки;
- `BGCOLOR=значення` — задає колір, який має бути фоновим для документа;
- `TEXT=значення` — задає колір тексту;
- `LINK=значення` — визначає колір гіперпосилань у документі;
- `ALINK=значення` — задає колір гіперпосилань під час кліцання;
- `VLINK=значення` — задає колір переглянутих гіперпосилань.

Браузер перетворює текст HTML-документа під час його виведення на екран, тобто ігнорує зайві пропуски, символи табуляції, символи кінця рядка, форматує текст відповідно до використаних тегів. Проте іноді виникає потреба у тому, щоб помістити в документ відформатований текст, для якого браузер повинен зберегти всі пропуски, навіть якщо кілька пропусків розміщені один за одним, знаки переносу на інший рядок, табуляції та інші символи форматування. Для цього використовують тег `<PRE>Текст</PRE>`. Втім, якщо браузер зустрине в такому тексті теги розмітки, вона буде відповідним чином застосована.

Теги форматування тексту

Текст у HTML-документах форматують за допомогою відповідних тегів. У табл. 2.1 наведено теги, які використовують для форматування символів, та заданий ними ефект.

Таблиця 2.1. Теги та задані ними ефекти

Теги	Ефект	Приклад
 та 	Напівжирне накреслення	bold
<I> та </I>	Курсивне накреслення	<i>italic</i>
<U> та </U>	Підкреслене	underlined
<S> та </S>	Перекреслене	stricken
^{та}	Верхній індекс	superscript
_{та}	Нижній індекс	subscript

Окрім цього, можна визначати різні параметри шрифту. Зокрема, за допомогою тегу <BASEFONT> задають гарнітуру (рисунок), розмір і колір шрифту. Цей тег одинарний, діє на весь текст, розміщений нижче, і має такі атрибути.

- FACE="значення" — гарнітура. Якщо зазначеної гарнітури на комп'ютері користувача немає, то текст буде оформлено з використанням гарнітури за умовчанням. В атрибуті FACE іноді записують через кому кілька гарнітур, тоді браузер вибирає гарнітуру за списком: за відсутності першої використовує другу і т. д. Значення атрибута слід подавати в лапках.
- FACE=значення — розмір шрифту (ціле число від 1 до 7).

Шрифти різних розмірів мають такий вигляд:

1 2 3 4 5 6 7

- COLOR=значення — колір шрифту. Як значення кольору можна використовувати назви кольорів англійською мовою, такі як red, green, blue тощо, або шістнадцяткові значення (записані в шістнадцятковій системі числення), що починаються з символу # і послідовно задають червоний, зелений та синій кольори, які мають бути змішані для отримання потрібного. Наприклад, #FF0000, #008000, #0000FF тощо.

Для оформлення сайту зазвичай використовують набір кольорів, який називають *безпечною палітрою*. Він складається з 216 елементів. Особливістю безпечних кольорів є те, що вони не змінюються у разі відображення різними браузерами або на різних моніторах, тобто ця палітра забезпечує найточнішу передачу на різних моні-

торах того, що задумав веб-дизайнер. Якщо будь-яка з трьох складових шістнадцяткового значення відрізняється від 00, 33, 66, 99, CC чи FF, то колір не є безпечним. Значення атрибутів SIZE та COLOR можна записувати без лапок.

Гарнітуру, розмір та колір шрифту для фрагмента тексту задають за допомогою тегу Текст. Атрибути цього тегу такі самі, як у тегу <BASEFONT>, за винятком атрибута SIZE: його значення зі знаком мінус (або плюс) означає, що розмір шрифту буде зменшено (чи збільшено) на відповідну величину відносно заданого за умовчанням або у тегу <BASEFONT>. Значення без знака задає абсолютний розмір шрифту.

Є й інші теги, які можна застосовувати для зміни параметрів тексту.

- <BIG>Текст</BIG> — збільшення розміру шрифту. Розмір символів тексту збільшується на одиницю відносно поточного рівня.
- <SMALL>Текст</SMALL> — зменшення розміру шрифту. Розмір символів зменшується на одиницю відносно поточного рівня.
- Текст — виділення важливих фрагментів тексту. Текст відображатиметься курсивом.
- Текст — створення перекресленого тексту. Текст буде перекреслено горизонтальною лінією.

Цікавого оформлення веб-сторінки надає пара тегів <MARQUEE> та </MARQUEE> — вони додають до веб-сторінки «рухомий рядок» із тексту, розташованого між ними.

Вправа 2.1

Створимо HTML-документ за допомогою найпростішого текстового редактора.

1. Відкрийте текстовий редактор Блокнот (Notepad).
2. Для створення HTML-документа наберіть такий текст:

```
<HTML>
<HEAD>
<TITLE>Купівля\продаж</TITLE>
</HEAD>
```

```

<BODY>
<H1>Торговельні об'єкти Львівщини</H1>
<H2>Супермаркети</H2>
<P>На Львівщині діє мережа магазинів Інтермаркет. У кожному районі є торговельний об'єкт.<P>Одним із таких магазинів став нещодавно відкритий супермаркет "АРСЕН".
<HR>
Це торговельний центр із багатьма відділами та зручною системою обслуговування покупців.<BR>Його можна назвати торговельним містечком.
</BODY>
<HTML>

```

3. Збережіть файл на жорсткому диску у створеній заздалегідь папці, назвавши його, наприклад, 1.html. Закрийте текстовий редактор.
4. Запустіть браузер Internet Explorer та відкрийте файл 1.html. Перегляньте отриманий результат.
5. Проаналізуйте, як відтворено веб-сторінку. Якщо потрібно внести зміни, виберіть команду Перегляд HTML-коду в меню Вигляд. Збережіть зміни та оновіть веб-сторінку за допомогою клавіші F5 або кнопки Оновити панелі інструментів.

На рис. 2.2 показано, як буде відтворено цей документ у вікні браузера Internet Explorer.

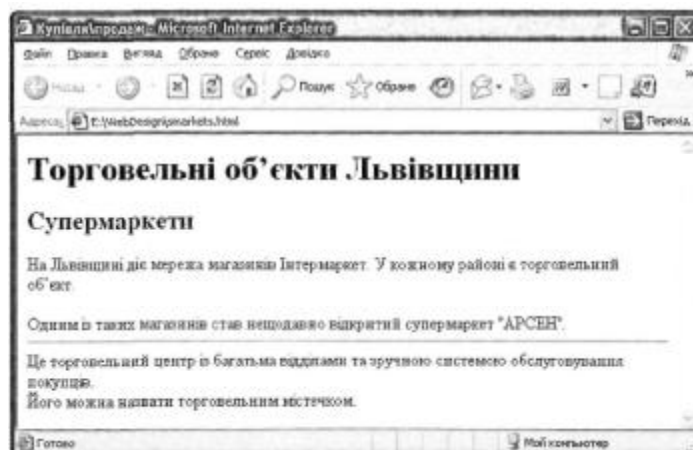


Рис. 2.2. Приклад веб-сторінки

Вправа 2.2

Помістимо на веб-сторінку наведений нижче текст, відформатувавши його за допомогою відповідних тегів.

Троянда сталиста

Граціозні форми, ніжне забарвлення, витончений аромат **троянди** з незапам'ятних часів чарували людей. Тому й створено **безліч легенд** про цю чарівну, ніжну й чудову квітку.

За староіндійською легендою, прекрасна богиня **Лакшмі** народилася з пуп'янка троянди.

У древньому індійському міфі розповідається, що прегарний юнак **Адоніс**, який уособлював прихід весни, постав із куща троянди.

У Персії так шанували царицю квітів, що навіть країна одержала назву Гюлістан — країна троянд.

У Римі цю квітку вважали символом хоробрості, нею нагороджували за доблесть. В Афінах часів **Аристофана** трояндовий вінок одягали на шию переможцям змагань. Гірлянди квітів прикрашали колони і стіни залів у дні тріумфальних святкувань.

Існує припущення, що **русальна неділя** — давнє свято в Україні та Білорусі — пов'язана з **розаріями** — святом римлян на честь цих прекрасних квітів.

*(Мамчур Ф.І., Гладун Я.Д.,
Лікарські рослини на присадибній ділянці)*

1. Введіть HTML-код веб-сторінки, на якій відобразатиметься наведений вище відформатований текст.

У цьому прикладі для деяких фрагментів тексту необхідно застосувати по два теги форматування, наприклад, <I><U>Україні</U></I>. Слід пам'ятати про правило вкладених тегів, згідно з яким їх потрібно закривати у зворотному до відкриття порядку. Тобто якщо спершу відкрили тег курсиву, а потім — тег підкреслення, то спочатку слід закривати тег підкреслення, а потім — курсиву.

2. Збережіть html-документ у файлі з розширенням **.html**. Вигляд створеної веб-сторінки у вікні браузера показаний на рис. 2.3.

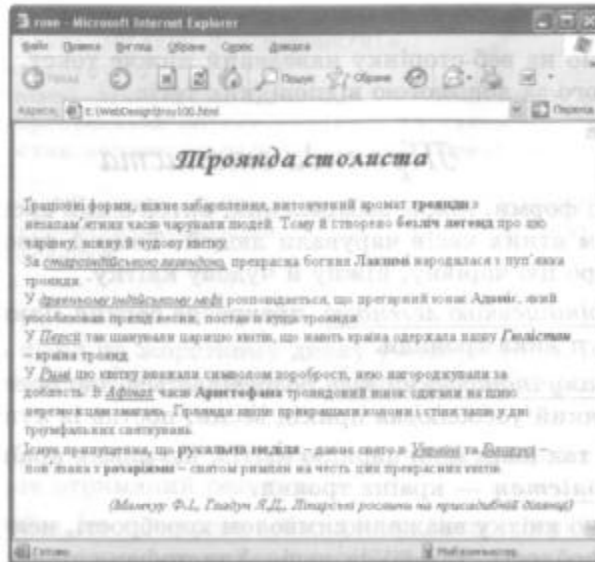


Рис. 2.3. Веб-сторінка про троянду столосту

Нумеровані й марковані списки

Для подання тексту в структурованому вигляді, коли кожен абзац виділено за допомогою позначки або номера, призначені списки. У HTML-документах використовують три види списків:

- невпорядкований (маркований);
- упорядкований (нумерований);
- список визначень.

У *невпорядкованому списку* для виділення елементів використовуються певні позначки (крапки, квадрати тощо). Починається такий список із тегу , а закінчується — . Кожний елемент списку починається з тегу . Тег може мати атрибут TYPE, який визначає форму позначки. Цей атрибут набуває таких значень:

- disk — зафарбоване коло;
- circle — коло;
- square — маленький чорний квадрат.

В *упорядкованому списку* всі елементи пронумеровані. Починається список із тегу , а закінчується — . Кожний елемент такого списку також починається з тегу . Тег може мати атрибут TYPE, який визначає тип нумерації. Цей атрибут набуває таких значень:

- A — велика літера;
- a — мала літера;
- I — велика римська цифра;
- i — мала римська цифра;
- 1 — арабська цифра.

У разі потреби за допомогою атрибута START можна задати відмінний від одиниці початковий номер елемента, наприклад <OL TYPE="I" START="5">, Нумерований список, початковий елемент якого позначено латинською літерою E, яка є п'ятою в алфавіті, описують так:

```
<OL TYPE="A" START="5">
<LI>Перший елемент списку</LI>
<LI>Другий елемент списку</LI>
<LI>Третій елемент списку</LI>
<LI>Четвертий елемент списку</LI>
</OL>
```

Список визначень (definition list) порівняно з маркованим або нумерованим має дещо складнішу будову. Кожен елемент такого списку складається з двох частин: терміна та його визначення. У такий спосіб можуть бути оформлені словники та термінологічні довідники. Однак цей тип списку використовують рідко, і тому описувати його докладніше не будемо.

Розглянемо приклади використання різних типів списків.

Приклад 2.1. Маркований список, в якому для позначення елементів використовують коло

```
<HTML>
<TITLE>Райдуга</TITLE>
<BODY>
<BASEFONT SIZE=6>
<UL TYPE=circle>
  <LI><FONT COLOR=red>Червоний</FONT></LI>
  <LI><FONT COLOR=orange>Оранжевий</FONT></LI>
```

```

<LI><FONT COLOR=yellow>Жовтий</FONT></LI>
<LI><FONT COLOR=green>Зелений</FONT></LI>
<LI><FONT COLOR=lightblue>Блакитний</FONT></LI>
<LI><FONT COLOR=blue>Синій</FONT></LI>
<LI><FONT COLOR=darkmagenta>Фіолетовий</FONT></LI>
</UL>
</BODY>
</HTML>

```

На рис. 2.4 показаний вигляд цього документа у вікні браузера Internet Explorer.

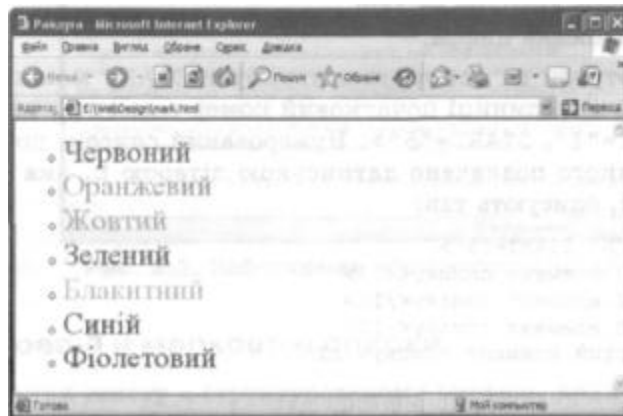


Рис. 2.4. Маркований список

Приклад 2.2. Нумерований список, в якому для позначення елементів використовують малі римські цифри

```

<HTML>
<TITLE>Райдуга</TITLE>
<BODY>
<BASEFONT SIZE=6>
<OL TYPE=i>
  <LI><FONT COLOR=red>Червоний</FONT></LI>
  <LI><FONT COLOR=orange>Оранжевий</FONT></LI>
  <LI><FONT COLOR=yellow>Жовтий</FONT></LI>
  <LI><FONT COLOR=green>Зелений</FONT></LI>
  <LI><FONT COLOR=lightblue>Блакитний</FONT></LI>
  <LI><FONT COLOR=blue>Синій</FONT></LI>
  <LI><FONT COLOR=darkmagenta>Фіолетовий</FONT></LI>
</OL>

```

```
</BODY>  
</HTML>
```

На рис. 2.5 показаний вигляд цього документа у вікні браузера Internet Explorer.

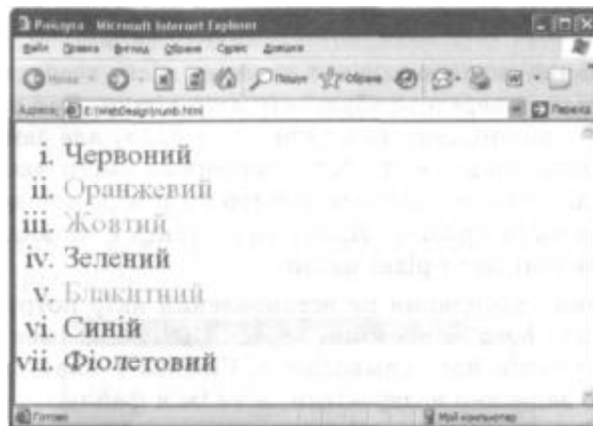


Рис. 2.5. Нумерований список

Зверніть увагу на те, що за допомогою тегу `<BASEFONT>` для всього документа призначено розмір шрифту 6, а за допомогою тегів `` визначено колір шрифту для кожного елемента списку.

Текстові гіперпосилання

Гіперпосилання — це вказівка для браузера, до якого об'єкта у межах або поза межами HTML-документа він має звернутися. За допомогою гіперпосилань користувачі можуть переходити з однієї веб-сторінки на іншу, завантажувати файли тощо. Як гіперпосилання може бути оформлений фрагмент тексту або графічний об'єкт. Коли веб-сторінка відображається у вікні браузера, текстове посилання зазвичай виділено синім кольором і підкресленням. Для переходу в місце, на яке вказує посилання, користувачу достатньо клацнути його текст.

Для створення гіперпосилання необхідно використовувати теги `<A>` та ``, визначивши для тегу `<A>` атрибут `HREF`. Його значенням має бути адреса URL, на яку вказує посилання. Текст посилання розташовують між тегами `<A>` і ``.

Якщо веб-сторінка, на яку вказує посилання, розмішена на іншому веб-сайті, то значенням атрибута HREF має бути повна URL-адреса з назвою протоколу включно; такі посилання називають *зовнішніми*. Якщо ж гіперпосилання вказує на іншу сторінку того самого сайту, то для пошуку документа достатньо задати лише відносний шлях; таке посилання називають *внутрішнім*.

Гіперпосилання може вказувати на певне місце всередині сторінки, якщо туди попередньо вбудувати *якір-мітку*. Для визначення якоря також використовують теги <A> і , але замість атрибута HREF задають атрибут NAME, значенням якого має бути ім'я якоря. Воно може складатися з літер та цифр, але не повинно містити символів пробілу. Якщо на сторінці є кілька міток, то всі вони повинні мати різні назви.

Для створення посилання на встановлений якір потрібно у тегу <A> зазначити його ім'я в кінці адреси URL після імені документа, відокремивши його символом #. Символ # вказує на те, що після нього записано назву мітки, а не ім'я файлу.

Посилання на мітку всередині поточного документа задають так:

```
<A HREF="#назва_мітки">Текст посилання</A>.
```

Якщо в атрибуті HREF задати адресу електронної пошти зі словом `mailto:` перед нею, то після вибору такого посилання можна надіслати електронний лист, де в полі Кому буде записано цю адресу.

У прикладі, який наводиться нижче, розглянуто застосування гіперпосилань різного типу.

Приклад 2.3. HTML-документ, в якому використовуються гіперпосилання та якорі

```
<HTML>
<HEAD>
<TITLE>Посилання та якорі</TITLE>
</HEAD>
<BODY>
Нові версії стандартних програм операційної системи
та найважливіші драйвери
можна знайти на веб-сайті компанії
<A HREF="http://www.microsoft.com/">Microsoft</A>.
```

```

<P>А тепер можна перейти на <A HREF="main.htm">
мою особисту сторінку</A>.
<P>Про те, як зв'язатися з автором, розповідається
в <A HREF="#olenap">
кінці цієї сторінки</A>
<BR><BR>
Можна використати матеріали, що розміщені
<A href="text.doc"> в цьому текстовому документі.</A>
<BR><BR>
<A NAME="olenap"
HREF="mailto:olena@zirka.lviv.ua">Olена@zirka.lviv.ua</A>
</BODY>
</HTML>

```

Такий вигляд матиме HTML-документ після його відтворення браузером (рис. 2.6).

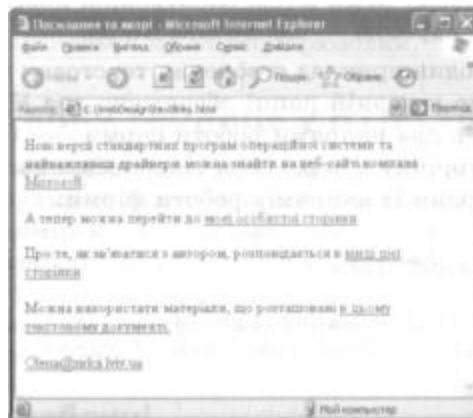


Рис. 2.6. Веб-сторінка з текстовими гіперпосиланнями

У цьому прикладі слово «Microsoft» міститься у тегу зовнішнього гіперпосилання, а текст «мою особисту сторінку» — у тегу внутрішнього посилання. Текст «кінці цієї сторінки» розміщений у тегу посилання на якор, а «в цьому текстовому документі» — у тегу гіперпосилання з атрибутом HREF, в якому задано зв'язок не з веб-сторінкою, а з текстовим документом, котрий збережений у тій самій папці, що й поточний HTML-документ. Текст «Olена@zirka.lviv.ua» вміщено у тег, що описує мітку якоря, а в атрибуті HREF зазначено адресу електронної пошти.

Якщо відвідувач веб-сторінки скористається зовнішнім гіперпосиланням Microsoft, то відкриється сторінка, що міститься за адресою: <http://www.microsoft.com/>. Після клацання гіперпосилання мою особисту сторінку відкриється веб-сторінка, записана у файлі main.htm, який міститься у тій самій папці, що й поточна сторінка. Після клацання внутрішнього посилання в цьому текстовому документі відкриється вікно з текстовим документом text.doc, що міститься у поточній папці. Якщо скористатися посиланням на якір кінці цієї сторінки, то зображення поточної веб-сторінки зміститься так, що текст мітки «[Olena@zirka.lviv.ua](#)», з якою зв'язаний якір, буде розміщений у видимій на екрані частині документа.

Водночас текст «[Olena@zirka.lviv.ua](#)» є гіперпосиланням, скориставшись яким, користувач зможе надіслати листа за допомогою настроєної на своєму комп'ютері електронної пошти за вказаною в атрибуті HREF адресою — olena@zirka.lviv.ua.

Розглянемо ще один приклад створення текстових гіперпосилань. Припустимо, що у певній папці збережено два HTML-документи, які описують два напрями роботи фірми, — 1.html та 2.html. Створимо веб-сторінку з переліком гіперпосилань, кожне з яких ілюструватиме один із напрямів роботи фірми:

```
<HTML>
<TITLE>Моя фірма</TITLE>
<BODY>
<H1><A HREF="1.html">Наша продукція</A></H1>
<H1><A HREF="2.html">Наші замовники</A></H1>
</BODY>
</HTML>
```

У цьому прикладі таги гіперпосилань містяться в тегах заголовків першого рівня. Тексти гіперпосилань будуть розташовані в окремих рядках і оформлені як заголовки першого рівня.

За умовчанням текстові гіперпосилання виділені синім кольором і підкреслені, а якщо ними вже скористалися, то темно-червоним. Для зміни цих кольорів додамо відповідні атрибути до тегу <BODY>: <BODY LINK=magenta ALINK=yellow>. Це означає, що після першого завантаження веб-сторінки всі гіперпосилання будуть фіолетового кольору, а якщо відвідувач скористається одним із них, то колір його тексту стане жовтим (рис. 2.7). Це зручно, коли на веб-сторінці є багато посилань, і відвідувач бажає по черзі

переглянути їх: тоді надання іншого кольору відвіданим гіперпосиланням дає змогу систематизувати такий перегляд.

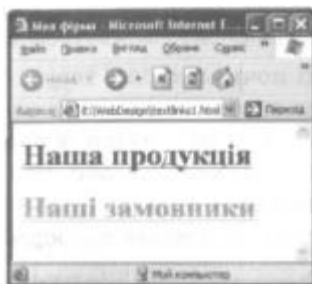


Рис. 2.7. Веб-сторінка з гіперпосиланнями, виконаними як заголовки першого рівня

Якщо додати до тегу `<BODY>` атрибут `BGColor=cyan`, то колір тла веб-сторінки зміниться на світло-бірюзовий.

Використання таблиць у HTML-документах

У найпростішій таблиці інформація розміщена у комірках, утворених у результаті поділу прямокутника на стовпці та рядки. У більшості комірок зберігаються дані, деякі комірки, зазвичай розміщені у верхній або боковій частині таблиці, містять заголовки. У HTML-документах таблицю заповнюють зліва направо, комірку за коміркою, починаючи з лівого верхнього кута і закінчуючи правим нижнім.

Створення таблиці

Для створення таблиці використовують чотири елементи. Таблицю описують за допомогою тегів `<TABLE>...</TABLE>`; вона повинна мати один або кілька рядків `<TR>...</TR>`, у кожному з яких може міститися заголовок `<TH>...</TH>` або дані `<TD>...</TD>`.

За умовчанням таблиця має невидимі межі комірок. Для того щоб зробити їх видимими, використовують атрибут `BORDER`, який має цілочислове значення, що визначає товщину рамки в пікселях, наприклад `<TABLE BORDER=5>`.

Таблиця може мати заголовок, який задається тегами `<CAPTION>` та `</CAPTION>`. Тег `<CAPTION>` може містити атрибут `ALIGN`

з одним із значень `top` або `bottom`, які визначають розташування заголовка відповідно перед таблицею або після неї (за умовчанням встановлено значення `top`). Наприклад, `<CAPTION ALIGN=bottom>Розклад уроків</CAPTION>`.

Кожний рядок таблиці починається з тегу `<TR>` і закінчується тегом `</TR>`. Якщо рядок містить заголовки стовпців таблиці, то використовують теги `<TH>` і `</TH>`, якщо ж дані — то `<TD>` і `</TD>`. Наприклад, `<TH>Понеділок</TH>` або `<TD>1. Алгебра</TD>`. У таблиці слово «Понеділок» буде виділене браузером напівжирним шрифтом, а текст «1. Алгебра» відформатовано стандартним способом.

Приклад 2.4. HTML-документ, що містить таблицю

Визначимо структуру табл. 2.2. Вона містить назви п'яти робочих днів тижня, на кожен з яких припадає 4-5 уроків.

Таблиця 2.2. Розклад уроків

Понеділок	Вівторок	Середа	Четвер	П'ятниця
1. Алгебра	1. Українська мова	1. Зарубіжна література	1. Географія	1. Хімія
2. Іноземна мова	2. Хімія	2. Геометрія	2. Фізична культура	2. Іноземна мова
3. Фізична культура	3. Алгебра	3. Іноземна мова	3. Фізика	3. Геометрія
4. Географія	4. Інформатика	4. Трудове навчання	4. Українська мова	4. Українська література
	5. Фізика	5. Українська література		

HTML-код для створення цієї таблиці матиме такий вигляд:

```
<HTML>
<TITLE>Приклад простої таблиці</TITLE>
<BODY>
<TABLE BORDER="1">
<CAPTION>Розклад уроків</CAPTION>
<TR>
<TH>Понеділок</TH>
<TH>Вівторок</TH>
<TH>Середа</TH>
<TH>Четвер</TH>
```

```

        <TH>П'ятниця</TH>
</TR>
<TR>
    <TD>1. Алгебра</TD>
    <TD>1. Українська мова</TD>
    <TD>1. Зарубіжна література</TD>
    <TD>1. Географія</TD>
    <TD>1. Хімія</TD>
</TR>
<TR>
    <TD>2. Іноземна мова</TD>
    <TD>2. Хімія</TD>
    <TD>2. Геометрія</TD>
    <TD>2. Фізична культура</TD>
    <TD>2. Іноземна мова</TD>
</TR>
<TR>
    <TD>3. Фізична культура</TD>
    <TD>3. Алгебра</TD>
    <TD>3. Іноземна мова</TD>
    <TD>3. Фізика</TD>
    <TD>3. Геометрія</TD>
</TR>
<TR>
    <TD>4. Географія</TD>
    <TD>4. Інформатика</TD>
    <TD>4. Трудове навчання</TD>
    <TD>4. Українська мова</TD>
    <TD>4. Українська література</TD>
</TR>
<TR>
    <TD></TD>
    <TD>5. Фізика</TD>
    <TD>5. Українська література</TD>
    <TD></TD>
    <TD></TD>
</TR>
</TABLE>
</BODY>
</HTML>

```

Який вигляд матиме ця веб-сторінка у вікні браузера, показано на рис. 2.8.

Зверніть увагу на те, що за відсутності даних у комірці таблиці рамка навколо неї не відображається. Для того щоб рамку було

видно, можна ввести спеціальний код ` ` (non-breaking space — нерозривний пробіл). Після цього порожня комірка буде взята в рамку.

Рис. 2.8. Документ, який містить таблицю

Об'єднання комірок

Кілька комірок можуть бути об'єднані в одну як по горизонталі, так і по вертикалі. Об'єднання першого типу застосовують тоді, коли потрібно створити для кількох стовпців спільний заголовок. Коли вміст кількох комірок послідовно у стовпці однаковий, їх об'єднують по вертикалі.

Для об'єднання комірок використовують такі атрибути: `COLSPAN` (об'єднання по горизонталі, у рядку) і `ROWSPAN` (по вертикалі, у стовпці) тегу `<TD>`. Значеннями цих атрибутів є кількість об'єднаних стовпців або рядків. Наприклад, `COLSPAN=3` означає, що комірка розтягнута на 3 стовпці, а `ROWSPAN=2` — що комірка займає 2 рядки.

Приклад 2.5. HTML-документ, який містить таблицю з об'єднаними комірками

Створимо HTML-документ, що містить дві таблиці, в яких є об'єднані комірки (табл. 2.3, табл. 2.4).

Таблиця 2.3. Таблиця з комірками, об'єднаними по вертикалі

10-A
10-B

Таблиця 2.4. Таблиця з комірками, об'єднаними по горизонталі

Класи		
11-A	11-B	11-B

Код цього документа наведено нижче.

```
<HTML>
<TITLE>Приклад ROWSPAN та COLSPAN</TITLE>
<BODY>
<TABLE BORDER="1" >
<CAPTION>Приклад ROWSPAN</CAPTION>
  <TR>
    <TD ROWSPAN="2">Класи</TD>
    <TD>10-A</TD>
  </TR>
  <TR>
    <TD>10-B</TD>
  </TR>
</TABLE>
<TABLE BORDER="1" >
<CAPTION>Приклад COLSPAN</CAPTION>
  <TR>
    <TD COLSPAN="3">Класи</TD>
  </TR>
  <TR>
    <TD>11-A</TD>
    <TD>11-B</TD>
    <TD>11-B</TD>
  </TR>
</TABLE>
</BODY>
</HTML>
```

На рис. 2.9 показано вигляд цього HTML-документа у вікні браузера. Ширина таблиць добирається автоматично. Їх заголовки таблиць відображено у двох рядках, оскільки вони не можуть виходити за межі таблиць.

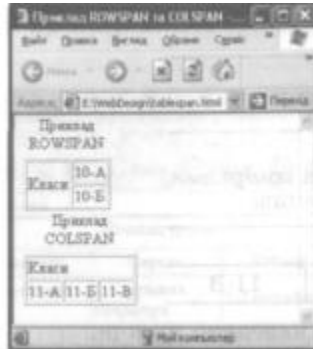


Рис. 2.9. Таблиці з об'єднаними комірками

Вправа 2.3

Створимо веб-сторінку з таблицею, що має таку ж структуру, як і табл. 2.5, в якій одночасно об'єднано і стовпці, і рядки.

Таблиця 2.5. Таблиця, в якій об'єднано кілька рядків і кілька стовпців

		Види спорту			
		Плавання		Гімнастика	
11-А	11-Б	11-А	11-Б	11-А	11-Б
15	19	10	5	11	8

1. Створіть HTML-документ, де у відповідних тегах `<TD>...</TD>` визначені атрибути `ROWSPAN` та `COLSPAN`.
2. Збережіть документ із розширенням `.html`. На рис. 2.10 показано, який вигляд має така таблиця у браузері. Комірка, в якій записано текст «Кількість учнів», об'єднує два рядки і два стовпці.

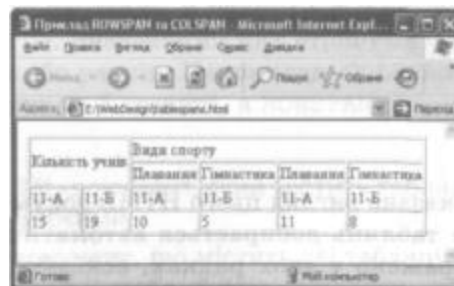


Рис. 2.10. Таблиця, де водночас об'єднано рядки і стовпці

Встановлення ширини таблиці та комірок

Розглянемо далі, як настроювати ширину всієї таблиці, а також окремих комірок. Зауважимо, що ширина всіх комірок в одному стовпці однакова, а в одному рядку може бути різною. Ширину таблиці задають атрибутом `WIDTH` у тегу `<TABLE>`, а комірки — за допомогою того самого атрибута у тегу `<TD>` або `<TH>`. Ширину таблиці можна визначати як у пікселях, так і у відсотках від ширини вікна, а комірки — у пікселях або у відсотках від ширини таблиці.

Приклад 2.6. HTML-документ, в якому задана ширина таблиці та деяких комірок

Розглянемо HTML-документ, в якому буде створено таблицю певної ширини, а також задано ширину деяких комірок.

```
<HTML>
<TITLE>Приклад встановлення ширини таблиці</TITLE>
<BODY>
<TABLE BORDER="1" WIDTH=500>
  <TR>
    <TD ROWSPAN="2" WIDTH=50%>Класи</TD>
    <TD WIDTH=250>10-A</TD>
  </TR>
  <TR>
    <TD>10-B</TD>
  </TR>
</TABLE>
</BODY>
</HTML>
```

На рис. 2.11 показано, який вигляд має цей документ у вікні браузера.

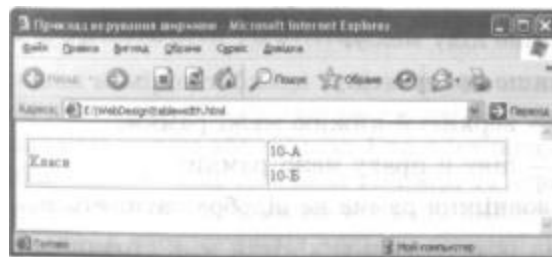


Рис. 2.11. Таблиця, ширину якої задано за допомогою атрибута `WIDTH`

Вся таблиця має ширину 500 пікселів. Ширина комірки з текстом «Класи» становить 50 % від ширини всієї таблиці, а комірки з текстом «10-А» — 250 пікселів. Комірка з текстом «10-Б» теж завширшки 250 пікселів (хоча для неї ширину не було задано), оскільки вона розташована в одному стовпці з попередньою.

Форматування тексту та оформлення рамок таблиці

Текст всередині комірок можна вирівнювати як по горизонталі, так і по вертикалі. Горизонтальне вирівнювання вмісту всього рядка задають за допомогою атрибута ALIGN у тегу <TR>; цей самий атрибут у тегах <TD> або <TH> в окремих комірках виконує таку саму роль. Атрибут ALIGN може набувати значень center, right, left або justify, які задають вирівнювання вмісту таблиці відповідно по центру, за правим чи лівим краєм або за шириною. Приклади тегів <TR> та <TD> із використанням вирівнювання: <TR ALIGN=right>; <TD ALIGN=center>1. Алгебра</TD>.

Вертикальне вирівнювання задають у тегах <TR>, <TD> або <TH> за допомогою атрибута VALIGN, який може набувати значень top, bottom та middle, що визначають вирівнювання відповідно за верхнім та нижнім краєм і по центру.

Для оформлення зовнішньої рамки таблиці можна використовувати атрибут FRAME тегу <TABLE>. Його значення визначають, що браузер відобразить:

- box — всі чотири сторони рамки;
- above — лише верхню межу рамки;
- below — лише нижню межу;
- lhs — лише ліву межу;
- rhs — лише праву межу;
- hside — верхню й нижню межі рамки;
- vside — ліву й праву межі рамки;
- void — зовнішня рамка не відобразиться.

Відображення розділювальних ліній між стовпцями та рядками таблиці визначають за допомогою атрибута RULES тегу <TABLE>.

Він може набувати таких значень:

- all — відобразити всі вертикальні та горизонтальні лінії;
- 4 rows — лише горизонтальні лінії між рядками;
- cols — лише вертикальні лінії між стовпцями;
- none — не показувати розділювальних ліній.

Вправа 2.4

Настроїмо зовнішній вигляд таблиці.

1. Створіть HTML-документ, який містить таблицю із такими настройками: товщина рамки таблиці дорівнює 15 пікселам, відображаються всі її чотири сторони і лише горизонтальні розділювальні лінії між рядками. Нехай ширина таблиці становить 300 пікселів.
2. Збережіть документ у файлі з розширенням .html. На рис. 2.12 показано, який вигляд має таблиця у вікні браузера.

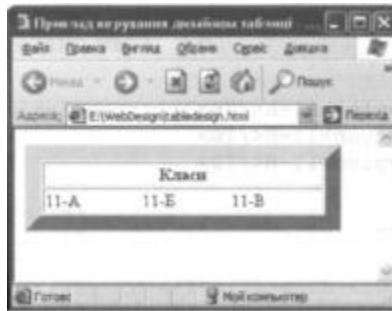


Рис. 2.12. Таблиця із рамкою товщиною 15 пікселів

Колірне оформлення таблиць

Для оформлення комірок таблиць за допомогою кольорів застосовують атрибут **BG**COLOR. Колір можна визначати, вказуючи його назву англійською мовою або символ # та шістнадцяткове число. Якщо потрібно задати колір для всієї таблиці, то атрибут **BG**COLOR задають у тегу `<TABLE>`, для зміни кольору лише в одному рядку — в тегу `<TR>`, а налаштування кольору окремої комірки забезпечує наявність цього атрибута всередині тегу `<TD>` або `<TH>`.

Колір рамки таблиці можна задати за допомогою таких атрибутів:

- **BORDERCOLOR** — колір ВСІєї рамки;
- **BORDERCOLORLIGHT** — колір світлої частини рамки;
- **BORDERCOLORDARK** — колір темної частини рамки.

Ці атрибути вставляють у тег `<TABLE>`. Щоб вони діяли, необхідна наявність ще й атрибута **BORDER**, який задає товщину рамки. Розглянемо приклад оформлення таблиці за допомогою різних кольорів.

Приклад 2.7. HTML-документ, який містить різнокольорову таблицю

```
<HTML>
<TITLE>Приклад керування кольором</TITLE>
<BODY>
<TABLE BGCOLOR="#F0F0F0" BORDER=10 BORDERCOLOR="#808080"
BORDERCOLORLIGHT="#9 9CCFF" BORDERCOLORDARK="#9900FF">
  <TR BGCOLOR=lightblue>
    <TH COLSPAN="3">Класи</TH>
  </TR>
  <TR>
    <TD BGCOLOR=pink>11-A</TD>
    <TD BGCOLOR=cyan>11-B</TD>
    <TD BGCOLOR=plum>11-B</TD>
  </TR>
</TABLE>
</BODY>
</HTML>
```

Якщо необхідно задати колір шрифту для тексту таблиці, то це можна зробити за допомогою тегу `` всередині відповідних тегів `<TD>`. Наприклад:

```
<TD BGCOLOR=magenta><FONT COLOR=pink>Текст</FONT></TD>
```

Використання таблиць для розміщення об'єктів на веб-сторінці

Під час відображення веб-сторінок браузер розміщує елементи на екрані один за одним у тому порядку, в якому вони записані у HTML-документі. Для позиціювання елементів розроблено спеціальні засоби мови HTML, проте часто для цього використовують

таблиці. На веб-сторінці створюють таблицю без рамок і у її комірки поміщають різні об'єкти.

Розмістимо назви класів по діагоналі так, як це показано в табл. 2.6, але самої таблиці на веб-сторінці відображати не будемо.

Таблиця 2.6. Таблиця, в якій інформація розміщується по діагоналі

Старші класи

10-A

10-B

11-A

11-B

11-B

Нижче наведено текст HTML-документа, в якому для позиціонування елементів використано табл. 2.6.

```
<HTML>
<TITLE>Приклад позиціонування</TITLE>
<BODY>
<TABLE>
  <TR>
    <TH>Старші класи</TH>
    <TD></TD><TD></TD><TD></TD><TD></TD><TD></TD><TD></TD><TD></TD></TR>
  <TR>
    <TD></TD><TD>10-A</TD><TD></TD><TD></TD><TD></TD><TD></TD><TD></TD></TR>
  <TR>
    <TD></TD><TD></TD><TD>10-B</TD><TD></TD><TD></TD><TD></TD><TD></TD></TR>
  <TR>
    <TD></TD><TD></TD><TD></TD><TD>11-A</TD><TD></TD><TD></TD><TD></TD></TR>
  <TR>
    <TD></TD><TD></TD><TD></TD><TD></TD><TD>11-B</TD><TD></TD><TD></TD></TR>
  <TR>
    <TD></TD><TD></TD><TD></TD><TD></TD><TD></TD><TD>11-B</TD></TR>
</TABLE>
</BODY>
</HTML>
```

На рис. 2.13 показано, який вигляд має цей HTML-документ у вікні браузера.

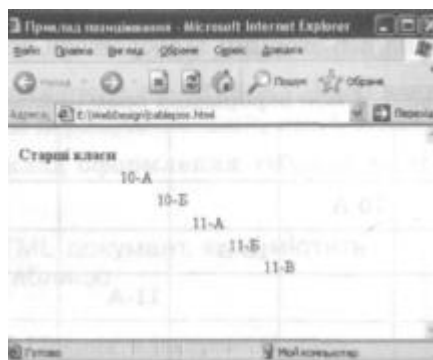


Рис. 2.13. Позичування елементів на веб-сторінці за допомогою таблиці

Комірки таблиці можна заповнювати не лише текстом, а й зображеннями, іншими таблицями та рухомими рядками. Текст або зображення всередині комірки може бути гіперпосиланням. Наприклад:

```
<TD><A HREF="lib.html">11-B</A></TD>
```

Вправа 2.5

Розробимо веб-сайт, на головній сторінці якого в кутах уявного трикутника будуть розміщені три гіперпосилання на сторінки про навчальні заклади. Кожне посилання пов'язане з файлом, в якому міститься інформація про навчальний заклад.

1. Визначте структуру головної сторінки сайту за допомогою табл. 2.7.

Таблиця 2.7. Структурна таблиця головної сторінки сайту

Школа «Дивосвіт»	
Гімназія «Сихівська»	Ліцей «Оріяна»

2. Підготуйте текстове наповнення для веб-сторінок, які пов'язані гіперпосиланнями з головною сторінкою.
 - Веб-сторінка про школу «Дивосвіт» — «Цю школу відвідують діти молодших класів, вони додатково вчаться малюванню, музиці та танцям».
 - Веб-сторінка про гімназію «Сихівська» — «Діти відвідують гімназію з п'ятого класу, додатково вивчають іноземні мови, інформатику, відвідують спортивні секції».
 - Веб-сторінка про ліцей «Оріяна» — «У ліцеї основну увагу приділяють вивченню математики та інформатики».
3. Розробіть HTML-код відповідних сторінок, скориставшись текстовим редактором Блокнот (Notepad).
4. Запишіть HTML-код для веб-сторінки про школу «Дивосвіт» і збережіть цей файл як 1.html.

```
<HTML>
<TITLE>Школа «Дивосвіт»</TITLE>
<BODY BGCOLOR=CYAN><BASEFONT SIZE=6 COLOR=olive>
<H1 ALIGN=CENTER><FONT FACE="Monotype Corsiva">
Школа «Дивосвіт»</FONT></H1>
<P>Цю школу відвідують діти молодших класів,
вони додатково вчаться малюванню, музиці та танцям.</P>
</BODY>
</HTML>
```

5. Запишіть HTML-код для веб-сторінки про гімназію «Сихівська» і збережіть цей файл як 2.html.

```
<HTML>
<TITLE>Гімназія «Сихівська»</TITLE>
<BODY BGCOLOR=lightcoral><BASEFONT SIZE=6 COLOR=indigo>
<H1 ALIGN=CENTER><FONT FACE="Monotype Corsiva">Гімназія
«Сихівська»</FONT></H1>
<P>Діти відвідують гімназію з п'ятого класу,
додатково вивчають іноземні мови, інформатику,
відвідують спортивні секції.</P>
</BODY>
</HTML>
```

6. Запишіть HTML-код для веб-сторінки про ліцей «Оріяна» і збережіть цей файл як 3.html.

```
<HTML>
<TITLE>Ліцей «Оріяна»</TITLE>
<BODY BGCOLOR=lavender><BASEFONT SIZE=6 COLOR=dark:cyan>
```

```

<H1 ALIGN=CENTER><FONT FACE="Monotype Corsiva">Ліцей «Орія-
На»</FONT></H1>
<P>У ліцеї основну увагу приділяють вивченню
математики та інформатики.</P>
</BODY>
</HTML>

```

7. Запишіть HTML-код для головної сторінки та збережіть цей файл як **index.html**. Для позиціювання гіперпосилань на головній сторінці використайте табл. 2.7. Її межі мають бути невидимі.

На рис. 2.14 показано головну сторінку сайту, на якій міститься інформація про навчальні заклади.



Рис. 2.14. Головна сторінка сайту, на якій міститься інформація про навчальні заклади

8. Відкрийте у вікні браузера головну сторінку створеного веб-сайту, клацніть одне з гіперпосилань, і на екрані відобразиться інформація про вибраний навчальний заклад.
9. Клацніть кнопку **Назад** на панелі інструментів браузера. Після цього знову відобразиться головна сторінка, на якій можна вибрати інше гіперпосилання.

HTML-документи, які містять фрейми

Ви, мабуть, помітили, що користуватися створеним сайтом не дуже зручно, оскільки для переходу на сторінку з головним меню необхідно клацати кнопку **Назад**. Тому в мові HTML є засіб, за допомогою якого можна створити меню, що постійно відображатиметься під час перегляду сайту. Інформація, що відкривається

за допомогою його посилань, буде розташована в іншому наперед визначеному місці.

Список усіх таких посилань, який фактично є змістом веб-сайту і постійно відображається на екрані, називають меню сайту або навігаційним меню. Типове меню складається з графічних зображень у вигляді кнопок або з текстових гіперпосилань, призначених для переходу в основні розділи сайту. Продумане навігаційне меню може легко провести відвідувача сайту всіма основними і додатковими розділами. Постійне відображення меню суттєво спрощує навігацію розділами, тому відвідувачі можуть більше часу присвячувати опрацюванню веб-сайту.

Фрейми, їхні теги й атрибути

Веб-сторінка сайту може містити кілька блоків або вікон, які називають *фреймами*, або *кадрами*. У кожному з них відображається свій HTML-документ. В одному фреймі може міститися навігаційне меню, а в іншому відкриватися веб-сторінки, на які вказують його пункти.

Для того щоб створити веб-сторінку з фреймами, потрібно кілька HTML-документів. В одному з них задають розмітку екрана, тобто розташовують у вікні браузера фрейми, кожному з яких призначають свої документи.

Сторінка з розміткою, як і звичайна, починається з тегу <HTML> і закінчується тегом </HTML>. Для поділу екрана на кілька фреймів використовують теги <FRAMESET> і </FRAMESET>. Перший має бути розташований після тегу заголовка, але перед тегом <BODY>. Іноді в таких документах зовсім не використовують тег <BODY>.

Два фрейми можна розташовувати поруч по горизонталі або один над одним. У першому випадку використовують атрибут COLS, а в другому — атрибут ROWS тегу < FRAME SET >. Для поділу вікна на фрейми через кому записують два числа, які визначають розміри фреймів. Для трьох фреймів потрібно три числа. Розміри фреймів вимірюють у пікселях або відсотках від розміру екрана. Якщо потрібно зазначити, що фрейм займає те місце, яке залишилося, використовують символ *.

Наприклад, тег <FRAMESET ROWS="150, *"> задає поділ вікна на два горизонтальні фрейми, один з яких має висоту 150 пікселів.

а другий займає те місце, що залишилося. Тег `<FRAMESET COLS="20%, 55%, *">` задає поділ вікна на три вертикальні фрейми, один з яких займає 20 % від ширини екрана, другий — 55 %, а третій займає те місце, що залишилося. Можна використовувати одночасно і горизонтальний, і вертикальний поділ вікна на фрейми — це роблять за допомогою вкладення тегів `<FRAMESET>` один в один.

Після поділу екрана на вікна для кожного фрейму слід задати HTML-документ, який відобразатиметься в ньому. Для цього використовують тег `<FRAME>` з атрибутами, що керують властивостями фреймів:

- `SRC` — задає ім'я файлу, що відобразатиметься у фреймі;
- `NAME` — задає ім'я фрейму;
- `SCROLLING` — визначає наявність (значення `yes`) або відсутність (значення `no`) смуг прокручування у вікні фрейму (за умовчанням — `yes`);
- `NORESIZE` — забороняє користувачу змінювати розміри фрейму;
- `BORDER` — визначає ширину розділювальної смуги між фреймами в пікселях;
- `BORDERCOLOR` — визначає колір розділювальної смуги між фреймами;
- `MARGINHEIGHT` — додає порожнє поле, висота якого визначена в пікселях, між верхньою межею фрейму і початком тексту або графіки;
- `MARGINWIDTH` — додає порожнє поле, ширина якого визначена в пікселях, між боковими межами фрейму і початком тексту або графіки.

Оскільки фрейми підтримують не всі браузер, необхідно помістити тег `<NOFRAME>` перед тегом `<BODY>`, а між `<BODY>` і `</BODY>` записати повідомлення, яке з'являтиметься у вікні, якщо браузер не підтримує фрейми. Після `</BODY>` має йти `</NOFRAME>`.

Вправа 2.6

Розмістимо на сторінці три документи про навчальні заклади (див. вправу 2.5) так, як це показано в табл. 2.8.

Таблиця 2.8. Схема розташування на сторінці фреймів, які містять інформацію про навчальні заклади

Документ про гімназію «Сихівська» (2.html)	Документ про ліцей «Оріяна» (3.html)
Документ про школу «Дивосвіт» (1.html)	

Як видно з таблиці, сторінку поділено спочатку на два рядки у співвідношенні приблизно 60 % та 40 %, перший рядок поділено на два стовпці, а в утворених комірках розташовано відповідні веб-сторінки про навчальні заклади. Документи **1.html**, **2.html** та **3.html** було створено під час виконання вправи 2.5.

1. Створіть файл головної сторінки та збережіть його як **main.html** у тій самій папці, що й решту документів.

```
<HTML>
<TITLE>Навчальні заклади</TITLE>
<FRAMESET ROWS="60%, 40%">
<FRAMESET COLS="50%,50%">
<FRAME SRC="2.html">
<FRAME SRC="3.html">
</FRAMESET>
<FRAME SRC="1.html">
</FRAMESET>
</HTML>
```

2. Відкрийте файл **main.html**. Який вигляд має сторінка із поділом на фрейми можна побачити на рис. 2.15.

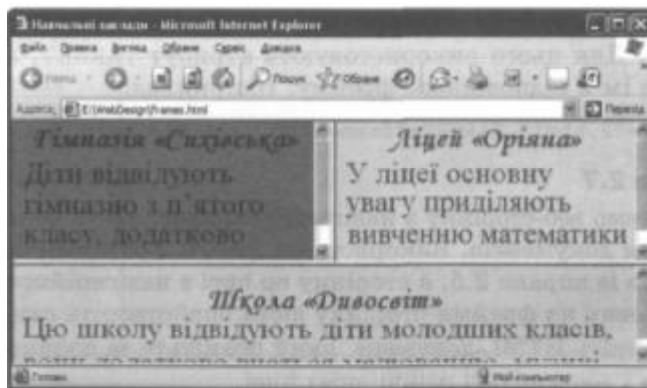


Рис. 2.15. Вікно містить три фрейми, в кожному з яких відображається окремий HTML-документ

Всі файли відображені у вікні браузера одночасно, а оскільки документи не вмістилися у визначені для них області, то кадри мають смуги прокручування.

Використання посилань у фреймах

У вправі 2.6 не використані гіперпосилання — між фреймами немає взаємозв'язку. Проте це не завжди зручно: навчальних закладів може бути багато, а поділ сторінки на велику кількість фреймів призводить до розпорошування уваги відвідувача сайту. Тому розглянемо веб-сторінку з навігаційною панеллю, яка розміщена у лівій меншій частині екрана, при цьому необхідна інформація відобразатиметься у більшій правій частині так, як це показано в табл. 2.9.

Таблиця 2.9. Схема розміщення фреймів, в якій використовується навігаційна панель

Навігаційна панель	Документ з інформацією про вибраний відвідувачем сайту навчальний заклад
--------------------	--

Якщо заплановано використовувати певний фрейм для відображення інформації, яка змінюватиметься після вибору відвідувачами іншого посилання, то йому надають ім'я. Наприклад, `<FRAME NAME="frame1">`. Веб-сторінка з гіперпосиланнями, що виконуватиме роль навігаційної панелі, складається із сукупності тегів `<A>...`. У них крім адреси файлу, пов'язаного з гіперпосиланням, слід зазначити ім'я фрейму, в якому цей файл відобразатиметься. Для цього використовують атрибут `TARGET`, значенням якого є ім'я відповідного фрейму. Наприклад:

```
<A HREF="2.html" TARGET="frame1">Гімназія «Сихівська»</A>
```

Вправа 2.7

Сформуємо веб-сторінку з навігаційною панеллю та областю відображення документів. Використайте файли з описами навчальних закладів із вправи 2.5, а сторінку `np.html` з навігаційною панеллю та поділену на фрейми сторінку `Index.html` створіть самостійно.

1. Створіть HTML-документ, який розмітьте за допомогою фреймів і збережіть у файлі `index.html`.

```
<HTML>  
<TITLE>Навчальні заклади</TITLE>
```

```

<FRAMESET COLS="20%,*">
<FRAME SRC="np.html">
<FRAME NAME="frame1">
</FRAMESET>
</HTML>

```

- Створіть HTML-документ, який міститиме навігаційну панель, і збережіть його у файлі np.html.

```

<HTML>
<TITLE>Школи, ліцеї, гімназії</TITLE>
<BODY>
<H2><A HREF="1.html" TARGET="frame1">Школа «Дивосвіт»
</A></H2>
<H2><A HREF="2.html" TARGET="frame1">Гімназія
«Сихівська»</A></H2>
<H2><A HREF="3.html" TARGET="frame1">Ліцей «Оріяна»</A>
</H2>
</BODY>
</HTML>

```

- Відкрийте файл index.html у вікні браузера. На рис. 2.16 показано, який вигляд має поділене на фрейми вікно з навігаційною панеллю після першого завантаження веб-сторінки, поки користувач не вибрав гіперпосилання.

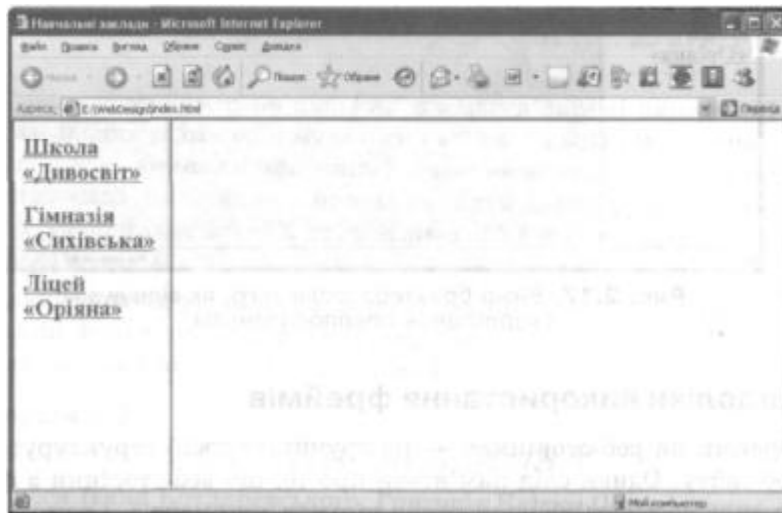


Рис. 2.16. Веб-сторінка index.html після першого завантаження

У документі index.html у п'ятому рядку, в тегу <FRAME>, задано лише один атрибут — NAME, тому після першого завантаження веб-сторінки права частина буде порожньою. Щоб цього уникнути, використовують атрибут SRC для повідомлення адреси файлу, який повинен відкриватись у цьому фреймі під час першого завантаження сторінки. Зазвичай це файл веб-сторінки з графічними вставками та загальною інформацією про сайт.

На рис. 2.17 показано, який вигляд має поділене на фрейми вікно з навігаційною панеллю після того, як відвідувач вибрав гіперпосилання на сторінку школи «Дивосвіт». Як бачите, навігаційне меню завжди в полі зору відвідувача в той час, коли він мандрує сторінками сайту.

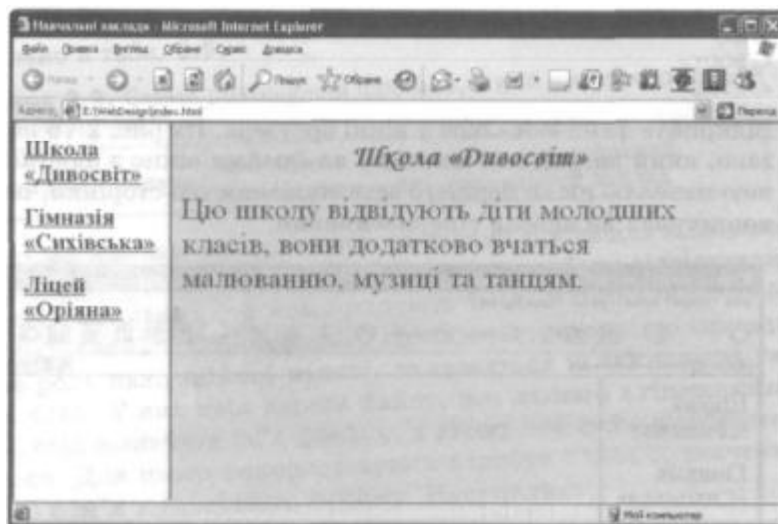


Рис. 2.17. Вікно браузера після того, як відвідувач скористався гіперпосиланням

Недоліки використання фреймів

Фрейми на веб-сторінках — це зручний спосіб структурування веб-сайту. Однак слід пам'ятати про те, що веб-сторінки з фреймами потребують завантаження кількох документів, і тому для їх відтворення браузеру необхідно більше часу, ніж для відтворення