

Структура веб-сторінки та її об'єкти.

Основні теги мови HTML

Web-сторінки мають вигляд звичайних текстових документів, в які введено вказівки, форматування. Принцип роботи браузера полягає в інтерпретації цих вказівок. При відображенні таких документів браузером самі вказівки не відображаються, проте впливають на спосіб відображення решти частини документу. Згадані вказівки називають *дескрипторами* або *тегами*. З їх допомогою текст можна робити кольоровим, використовувати шрифти різного розміру, вбудовувати мультиплікацію, відео фрагменти тощо. Формат дескрипторів задається в описі спеціальної мови розмітки. Вона називається *мовою розмітки гіпертекст-HTML* (*HyperTex Markup Language*). Наприклад, вигляд головної сторінки сайту Ukr.Net та її код подано на рис 1.

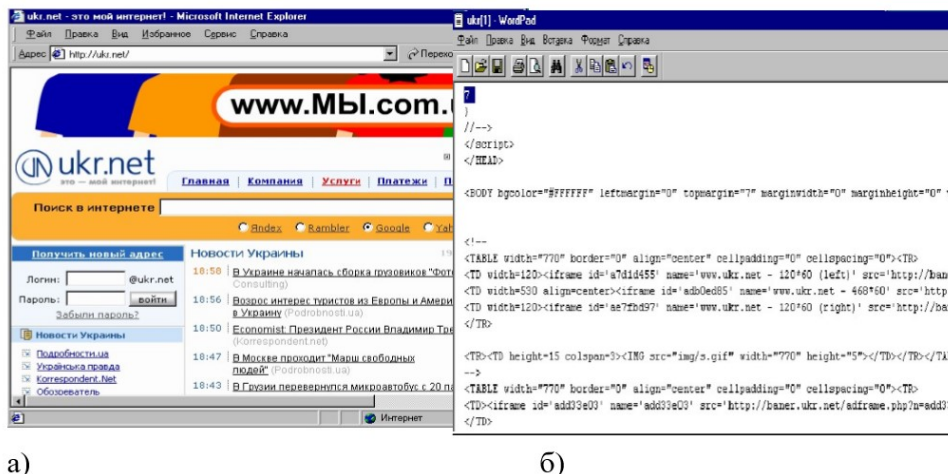


Рис. 1. Головна сторінка сайту Ukr.Net (а) та її код (б).

Документи, розмічені за допомогою цієї мови, називають *HTML-документами*. HTML-документи мають розширення .htm або .html.

Іноколи процес розробки Web-документів засобами мови HTML називають Web-програмуванням. Проте слід розуміти, що HTML не є мовою програмування у звичайному розумінні, а є мовою розмітки (опису). Термін Web-програмування мовою HTML має історичне походження. Під терміном Web-програмування будемо розуміти процес розробки Web-документів (і не лише засобами HTML).

Мова HTML розроблена спеціально для Web. Її популярність забезпечують зокрема такі властивості:

- ✓ документ, створений за допомогою деякої програми, наприклад текстового редактора, часто важко (а іноді і неможливо) використовувати в іншій програмі; HTML у цьому відношенні є універсальною;
- ✓ HTML — це відкритий стандарт;

- ✓ HTML не є власністю якої-небудь фірми;
- ✓ можливість використання гіпертексту;
- ✓ HTML підтримує мультимедіа.

У мові HTML використовуються поняття *контейнера*. Контейнер — це дескрипторна пара, яка складається з початкового і кінцевого *дескрипторів (тегів)*. Початковий дескриптор має вигляд <TAG>, де TAG ім'я певного HTML-дескриптора. Кінцевий дескриптор має вигляд </TAG>. Наприклад,

```
<B> Програми для перегляду Web-сторінок називаються браузером </B>
```

Контейнер дає вказівку браузеру відображати текст «Програми для перегляду Web-сторінок називаються браузером» напівжирним шрифтом. Тобто, контейнери впливають на частину документа, розміщену між ними. Зазначимо, що контейнери можуть бути вкладені.

Одиночний дескриптор, звичайно, має самостійне завдання, не пов'язане з конкретним текстом. Наприклад, дескриптор <HR> (від слів Horizontal Line) служить для створення і відображення горизонтальної лінії.

Дію дескриптора можна дещо змінити, задаючи певні *атрибути {параметри}*. Атрибути — це додаткові ключові слова, які відокремлюються від ключового слова, що визначає дескриптор, пропуском і розміщуються до символу «>». У контейнерах атрибути додаються тільки до початкового дескриптора. Атрибути задаються своїми значеннями. Наприклад,

```
<H1 ALIGN=»LEFT»>Транспортний рівень моделі TCP/IP</H1>
```

— це вказівка браузеру відобразити текст «Транспортний рівень моделі TCP/IP» у вигляді заголовка та вирівняти його по лівому краю.

HTML-код Web-сторінки може містити коментарі, тобто деякий текст, який не відображається браузером і служить для пояснення призначення сторінки або частини її коду. Текст, що повинен служити коментарем, необхідно помістити між символами «<!--» і «^». Наприклад,

```
<HR> <!--створюємо горизонтальну лінію ^
```

Коментар можна поставити у будь-яке місце коду сторінки, де дозволяються пропуски.

Контейнер <HTML></HTML> є ознакою того, що даний файл містить документ HTML. У HTML-документах є «голова» (заголовок) і «тіло» (основна частина).

Розділ <HEAD></HEAD> містить дескриптори, які описують документ в цілому. Зокрема, тут вказується назва документа.

Контейнер <TITLE></TITLE> служить для визначення назви документа.

Текст, включений в нього, відображається у верхній частині вікна браузера. Назва сторінки - це один з елементів, які мають важливе значення для привернення уваги «відвідувачів». Вона повинна відображати зміст сторінки. Системи пошуку орієнтуються саме на назву сторінки, тому сторінка із змістовною назвою має більше шансів бути вибраною у процесі проведення пошуку інформації з конкретної теми.

Контейнер `<BODY></BODY>` задає основну частину документа — його «тіло». Інформація, розміщена між дескрипторами `<BODY>` та `</BODY>`, відображається в області документа.

Всі HTML-документи мають однакову загальну структуру. Праворуч показано відображення цього коду браузером.

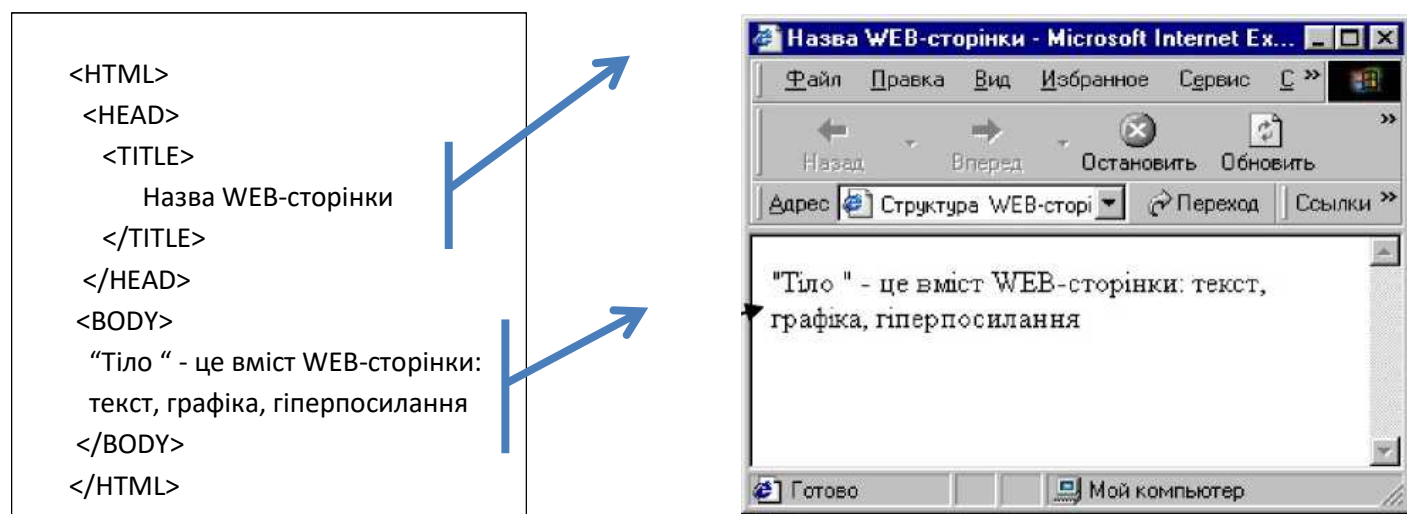


Рис. 2. Структура HTML-документа

Крім контейнера `<TITLE>...</TITLE>` у заголовок документа (контейнер `<HEAD>....</HEAD>`) часто включаються дескриптори `<BASE>`, `<META>`.

Дескриптор `<BASE>` використовується для вказування повної URL адреси документа. Це дозволяє підтримувати відносні гіперпосилання в робочому стані при переміщенні документа в інший каталог і навіть на інший комп'ютер.

Адреса URL задається за допомогою атрибута `HREF`, наприклад,

```
<BASE HREF="http://www.ia.net/~rmeegan">
```

Докладніше цей дескриптор описано при розгляді гіперпосилань.

Дескриптор `<META>` дозволяє автору документа описати інформацію або виконати дії, які ще не підтримуються офіційною версією HTML. Він найчастіше використовується завдяки його властивості `Keywords` (ключові слова). У багатьох пошукових системах використовуються слова, визначені в цьому елементі для складання вказівника документа. Наприклад,

```
<META NAME="Keywords" CONTENT="внз, університет, інформатика">  
<META NAME="Description" CONTENT="сторінки ДНУ">
```

Цей дескриптор дозволяє також задати метайнформацію про кодову сторінку, яка використовується для кодування літер кирилиці:

```
<META content="text/html; charset=Windows-1251" http-equiv=Content- Type>
```

Контейнер <BODY> має велику кількість атрибутів, важливих для визначення загального виду документа. Основні з цих атрибутів задано в таблиці 1.

Таблиця 1.

Атрибут	Призначення
ALINK	Визначає колір, яким виділяється активне гіперпосилання
BACKGROUND	Вказує URL зображення, яке слід використовувати як фон документа
BGCOLOR	Визначає колір фону документа
LINK	Визначає колір гіперпосилання, яке не було «відвідане»
TEXT	Визначає колір тексту документа
VLINK	Визначає колір гіперпосилання, яке було «відвідане»

Створити просту сторінку можна, наприклад, так:

1. Запустити текстовий редактор Блокнот.
2. Ввести текст HTML-документа, поданий на Рис. 2.
3. Зберегти під деяким іменем із розширенням .htm (наприклад, Структура WEB-сторінки.htm). При збереженні, у вікні «Сохранение документа» слід поміняти тип файла з «Текстовые документы» на «Все файлы». Інакше редактор у кінець назви файла автоматично додасть розширення .txt.
4. Запустити програму Internet Explorer.
5. За допомогою меню «Файл -> Открыть» відкрити створений файл (Структура WEB-сторінки.htm).

В наступному прикладі наводиться код сторінки з синім фоном:

```
<HTML>
<HEAD>
  <TITLE> Сторінка з фоном </TITLE>
</HEAD>
<BODY BGCOLOR="blue">
  Білий текст на синьому фоні
</BODY>
</HTML>
```

Використання фреймів

Мова HTML дозволяє поділити вікно браузера на окремі секції, які називаються фреймами (рамки). Вміст кожного фрейму знаходиться в окремому HTML-файлі. Кожен фрейм можна переглядати або змінювати не залежно від решти фреймів. Браузер, який “розпізнає” фрейми, завантажує різні сторінки у різні секції свого вікна.

Наприклад, можна побудувати сторінку так, що фірмовий знак буде зафіксовано у верхній частині вікна, збоку буде розміщено панель навігації сторінками сервера, а в решту вікна будуть завантажуватися різні сторінки, які користувач зможе переглядати звичайним чином. Коли користувач клацає кнопкою мишки на гіперпосиланні - змінюється лише частина вікна, а система навігації і фірмовий знак залишаються нерухомими.

Подібну структуру має сторінка, вигляд якої подано на Рис. 3

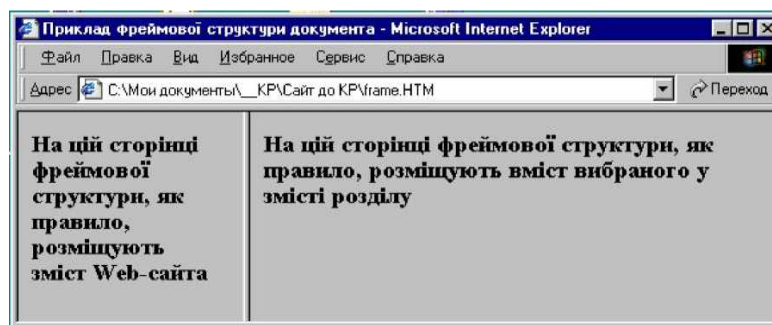


Рис. 3. Поділ вікна на два фрейми

Код даної сторінки наведено нижче:

```
<HTML>
<HEAD>
  <TITLE>Приклад фреймової структури документа</TITLE></HEAD>
<BODY>
  <frameset cols="30%,70%">
    <frame src="a1.htm">
    <frame src="a2.htm">
  </frameset>
</BODY>
</HTML>
```

Файл a1.htm

```
<HTML>
<HEAD>
  <TITLE>Приклад фреймової структури документа</TITLE>
</HEAD>
<BODY bgcolor="silver" text="black" link="#ff0000">
  <h3> На цій сторінці фреймової структури, як правило, розміщують зміст Web-сайта</h3>
</BODY>
</HTML>
```

Файл a2.htm


```

<HTML>
<HEAD>
  <TITLE>Приклад фреймової структури документа</TITLE>
</HEAD>
<BODY bgcolor="silver" text="black" link="#ff0000">
  <h3> На цій сторінці фреймової структури, як правило, розміщують вміст вибраного у змісті
розділу<ФЗ>
</BODY>
</HTML>

```

Створення списків, таблиць

Для створення списків використовують теги, що задають тип списку. Нумерований список починається тегом `` і завершується тегом ``, наприклад:

<pre> Один Два Три Чотири П'ять </pre>		<pre> 1. Один 2. Два 3. Три 4. Чотири 5. П'ять </pre>
--	---	---

Тег `` може мати такі параметри:

TYPE - вид лічильника:


A - великі латинські букви a - малі латинські букви

I - великі римські цифри

i - малі римські цифри

1 - звичайні цифри

START - встановлює число, з якого буде починатися відлік, наприклад:

<pre> <OL TYPE="I" START="8"> Вісім Дев'ять Десять Одинадцять Дванадцять </pre>		<pre> VIII. Вісім IX. Дев'ять X. Десять XI. Одинадцять XII. Дванадцять </pre>
--	---	---

Неномерований список починається тегом `` и завершується тегом ``. Кожен елемент списку починається з тегу ``, наприклад:

< UL >	Один
 Один	Два
 Два	Три
 Три	Чотири
 Чотири	П'ять
 П'ять	
	

Тег може включати параметр TYPE із значеннями disc, circle, square

```
<UL TYPE=disc><LI> disc </UL> . disc
<UL TYPE=circle><LI> circle
o circle
</UL>
<UL TYPE=square><LI> square
■square
</UL>
```

Дані значення - це зовнішній вид маркера, який за замовчуванням ставиться у вигляді диску, тобто disc.

Списки визначень мають такий вид:

<DL>	Термін
<DT> Термін	Визначення терміну
<DD> Визначення терміну	Визначення терміну
<DD> Визначення терміну	Визначення терміну
<DD> Визначення терміну	Визначення терміну
</DL>	

Таблиці у WEB-документах застосовуються не тільки для розміщення табличних даних, а й для вставки зображень і посилань, для раціонального компонування WEB- сторінки.

Елементи таблиць. Таблиці будуються за принципом вкладання і вводяться на WEB-сторінку за допомогою ряду елементів. Кожна таблиця починається тегом <TABLE> і закінчується тегом </TABLE>. Створювана таблиця ніби розгортається по рядках, а рядки заповнюються комірками. При цьому всередину тегів <TABLE> ...</TABLE> можуть вставлятися такі елементи:

TR - елемент створення рядка;
 TD - елемент, що визначає вміст комірки даних;
 TH - елемент, що визначає комірку заголовка.

Наприклад, для створення таблиці 3*2 використовується такий шаблон:

```
<TABLE>
<TR><TD> ... </TD><TD> ... </TD></TR>
```

```
<TR><TD> ... </TD><TD> ... </TD></TR>
<TR><TD>... </TD><TD> ... </TD></TR>
</TABLE>
```

де крапками позначений вміст кожної комірки.

Ячейка 1	Ячейка 2
Ячейка 3	Ячейка 4

Така таблиця реалізується наступним кодом:

```
<TABLE BORDER="2"
WIDTH="200"
BGCOLOR="#00FF00">
<TR>
<TD>Ячейка 1</TD>
<TD>Ячейка 2</TD>
</TR>
<TR>
<TD>Ячейка 3</TD>
<TD>Ячейка 4</TD>
</TR>
</TABLE>
```

У прикладі створена таблиця з фіксованою шириною (WIDTH="200" пікселів), але краще використовувати проценти, т.я. в цьому випадку розмір таблиці буде змінюватися в залежності від розміру вікна.

Таблиця починається тегом <TABLE> и завершується тегом </TABLE>.

Тег <TABLE> може включати такі атрибути:

Таблиця 2.

WIDTH="n"	Визначає ширину таблиці в пікселях або процентах, за замовчуванням ширина таблиці визначається вмістом клітинок.
BORDER="n"	Встановлює товщину рамки. За замовченням n=0 - таблиця відображається без рамки.
BORDERCOLOR="#FFFFFF"	Встановлює колір обрамлення, де #FFFFFF (білий) - шістнадцяткове значення кольору.
BGCOLOR="#FFFFFF"	Встановлює колір фону для всієї таблиці, де #FFFFFF - шістнадцяткове значення кольору.
BACKGROUND="image.gif"	Заповнює фон таблиці зображенням.
CELLSPACING="n"	Визначає відстань між рамками клітинок таблиці в пікселях.
ALIGN=LEFT	Визначає положення таблиці в документі. За замовчуванням таблиця притиснута до лівого краю сторінки. Допустимі

	значення атрибута: LEFT (зліва), CENTER (по центру сторінки), RIGHT (справа).
FRAME-'значение"	Управляє зовнішнім обрамленням таблиці і може приймати такі значення: VOID - обрамлення немає (значення за замовченням). ABOVE - межа тільки зверху. BELOW - межа тільки знизу. HSIDES - межа тільки зверху і знизу. VSIDES - межа тільки зліва і справа. LHS - тільки ліва межа. RHS - тільки права межа. BOX - малюються всі чотири сторони. BORDER - також всі чотири сторони.
RULES="n"	Управляє лініями, що розділяють клітинки таблиці. Можливі такі значення (n): NONE - немає ліній (значення за замовченням). GROUPS - лінії будуть тільки між групами рядів. ROWS - тільки між рядками. COLS - тільки між колонками. ALL - між всіма рядками і колонками.

Таблиця може включати заголовок, який розміщується між тегами <CAPTION></CAPTION>. Він має бути безпосередньо після тега <TITLE>. До заголовку можна застосувати атрибут ALIGN, що визначає його положення відносно таблиці:

TOP - значення за замовчуванням, заголовок над таблицею по центру.
LEFT - заголовок над таблицею зліва.
RIGHT - заголовок над таблицею справа.
BOTTOM - заголовок під таблицею по центру.

Рядки таблиці починаються тегом <TR> и завертаються тегом </TR>, а кожна клітинка таблиці починається тегом <TD> и завершується </TD>. Дані теги можуть мати такі атрибути:

Таблиця 3

Атрибути, що можуть застосовуватися до рядків і клітинок	
ALIGN=LEFT	Встановлює горизонтальне вирівнювання тексту в клітинках рядка. Може приймати значення LEFT (вирівнювання вліво), CENTER (вирівнювання по центру), RIGHT (вправо).
VALIGN=TOP	Встановлює вертикальне вирівнювання тексту в клітинках рядка. Допустимі значення: TOP (вирівнювання по верхньому краю), CENTER (вирівнювання по центру - це значення приймається за замовчуванням), BOTTOM (по нижньому краю).

BGCOLOR="#FFFFFF"	Встановлює колір фону рядка або клітинки.
BACKGROUND="image.gif"	Заповнює фон рядка або клітинки зображенням.
Атрибути, що можуть застосовуватися тільки до клітинок	
WIDTH="n"	Визначає ширину клітинки в n пікселях.
HEIGHT="n"	Визначає висоту клітинки в n пікселях
COLSPAN="n"	Розтягування клітинки по горизонталі. Наприклад, <TD COLSPAN = "2"> означає, що клітинка буде розтягнута на 2 колонки.
ROWSPAN="n"	Розтягує клітинки по рядку. Наприклад, <TD ROWSPAN="2" означає, що клітинка буде розтягнута на два рядки таблиці.
NOWRAP	Цей атрибут показує, що текст буде розміщено в один рядок
BACKGROUND="image.gif"	Заповнює фон клітинки зображенням

Крім цього, будь яка клітинка таблиці може бути визначена не тегами <TD></TD>, а тегами <TH></TH> - Table Header (заголовок таблиці). В принципі, це звичайна клітинка, але текст між цими тегами буде виділений напівжирним шрифтом і відцентрований.

Якщо клітинка порожня, то навколо неї рамка не малюється. Якщо рамка все ж потрібна навколо порожньої клітинки, то в неї треба ввести символний об'єкт [](#) (non-breaking space - нерозривний пропуск). Клітинка буде порожня, а рамка навколо неї буде ([](#) - обов'язково треба набирати малими буквами і закриватися символом «крапка з комою»).

Теги, що встановлюють шрифт (, <I>, <FONT SIZE="n", FONT COLOR="#FFFFFF"), необхідно повторювати для кожної клітинки.

Оформлення тексту, створення заголовків. Графічні об'єкти

Форматування тексту передбачає створення заголовків, абзаців, вирівнювання тексту, вибір стилів шрифту (напівжирний, курсив, підкреслений), зміну розміру шрифту.

У HTML існує шість рівнів заголовків і кожному з них відповідає свій розмір шрифту. За допомогою заголовків різних рівнів можна створювати чітку ієрархічну структуру документа. Заголовку I рівня відповідає найбільший розмір шрифту. Синтаксис HTML-коду, що створює заголовки різного рівня та відображення цього коду браузером:

```
<H1> Заголовок першого рівня </H1>
<H2> Заголовок другого рівня </H2>
<H3> Заголовок третього рівня </H3>
<H4> Заголовок четвертого рівня </H4>
<H5> Заголовок п'ятого рівня </H5>
```

<H6> Заголовок шостого рівня </H6>

Для кращого сприйняття сторінки користувачем не слід пропускати рівні заголовків.

Для відображення тексту в HTML немає спеціального дескриптора. Текст вводиться у будь-якому місці між дескрипторами <BODY></BODY>. У текст не потрібно вставляти ні порожні рядки, ні символи переведення рядка, оскільки при відображенні сторінки браузером вони не враховуються. Текст автоматично розміщується за шириною вікна браузера. Якщо ж необхідно на Web-сторінці помістити порожній рядок або забезпечити в потрібному місці перехід до наступного рядка, слід використовувати спеціальні дескриптори:

дескриптор розриву абзацу <P></P> та

дескриптор розриву рядка
.

Дескриптор розриву абзацу дає команду браузеру закінчити поточний абзац і помістити порожній рядок перед наступним абзацом. Дескриптор розриву рядка вказує про закінчення поточного рядка і переходу до наступного. Для дескриптора розриву абзацу використовувати закриваючий дескриптор </P> не обов'язково.

Використання згаданих дескрипторів розглянуто на прикладі:

<HTML>

<HEAD>

<TITLE> Приклад використання дескрипторів розриву рядка та абзацу </TITLE>

</HEAD>

<BODY>

Дескриптор розриву абзацу дає команду роузера вставити порожній рядок перед початком наступного абзацу.

<P> Дескриптор розриву рядка дає команду браузеру
 перейти у наступний рядок. </P>

</BODY>

</HTML>

Відображення сторінки з таким кодом подано на Рис. 4.

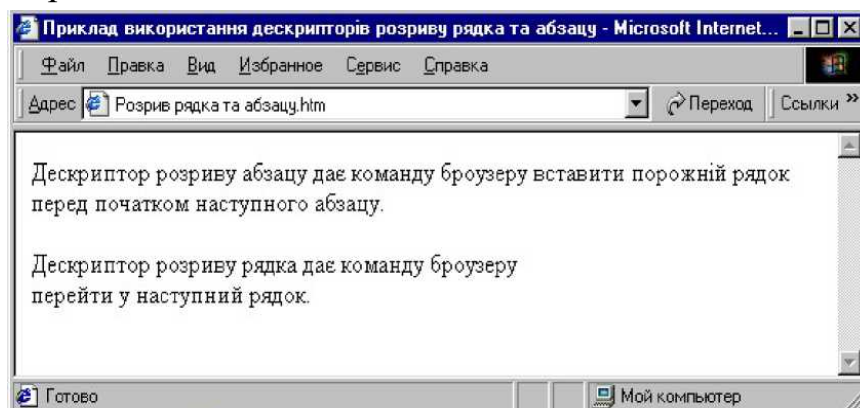


Рис. 4. Абзаци та пусті рядки

Текст абзацу чи заголовка може бути вирівняний по лівому краю, по правому

краю, або по центру вікна браузера. Для вирівнювання тексту використовується параметр *ALIGN* вирівнювання, де *вирівнювання* може приймати значення: LEFT для вирівнювання по лівому краю, RIGHT для вирівнювання по правому краю, CENTER для вирівнювання по центру. Найчастіше параметр ALIGN використовується в дескрипторах розриву абзацу та визначення заголовка.

Наприклад:

```
<HTML>
<HEAD>
< TITLE > Приклад вирівнювання тексту </ TITLE >
</HEAD>
<BODY>
<H2 ALIGN=CENTER>H^ заголовок буде розміщено посередині</H2>
<!-- Наступний абзац буде вирівняно по правому краю ^
<P ALIGN=RIGHT> Цей абзац вирівняно по правому краю Вивчення HTML невдовзі стане такою ж
необхідністю, як вміння ввімкнути комп'ютер, скопіювати файл чи скористатися текстовим
редактором. Якщо користувач вміє набирати текст, то зможе освоїти і використання HTML.
</P>
</BODY>
</HTML>
```

Відображення сторінки з таким кодом подано на Рис. 5

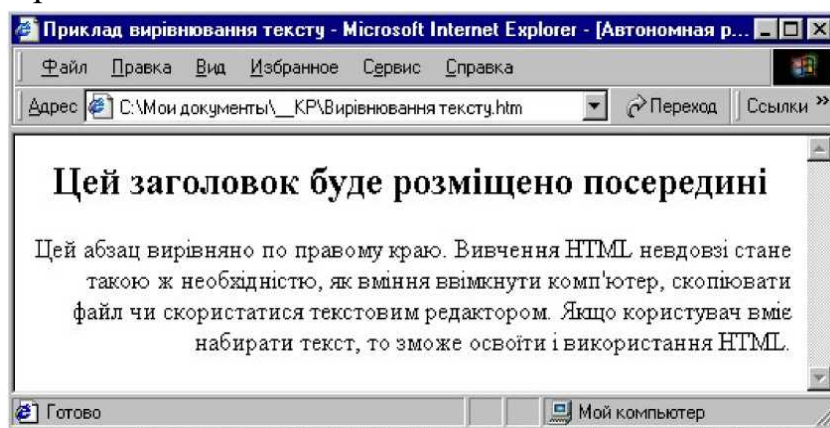


Рис. 5. Вирівнювання абзацу по центру та по правому краю

У HTML є контейнер, в який можна помістити заздалегідь відформатований текст. Контейнер `<PRE>...</PRE>` використовується для відображення відформатованого текстовим редактором блоку тексту, який складно було б прочитати, якщо б браузер переформатував його (наприклад, блоки програмного коду мовою C++). Цей контейнер дозволяє зберегти розбиття тексту на рядки (символи переходу на новий рядок, отримані в результаті натискування клавіші `<Enter>`, інтерпретуються як розриви рядка), пропуски всередині тексту інтерпретуються у точній відповідності з їх розміщенням у текстовому редакторі. Недоліком такого підходу є втрата гнучкості, а також те, що браузера, зазвичай, виділяють відформатований заздалегідь текст із допомогою моноширинного шрифту. Це призводить до того, що довгі фрагменти тексту зливаються. У

наступному прикладі створюється сторінка із заздалегідь відформатованим текстом:

```
<HTML>
<HEAD>
<TITLE> Приклад заздалегідь відформатованого тексту </TITLE> </HEAD>
<BODY>
<H1 ALIGN=CENTER> Книги про HTML </H1>
<PRE> НАЗВА  АВТОР  ВИДАВНИЦТВО  РІК
-----
Использование HTML  М.Браун  «ВИЛЬЯМС»  2000
</PRE>
</BODY>
</HTML>
```

Відображення документа з таким кодом у браузері подано на Рис. 6.

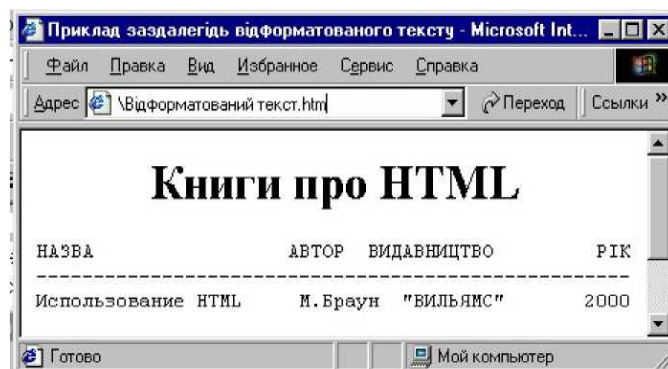


Рис.6. Використання відформатованого тексту

Мова HTML надає можливість вибирати для написання фрагментів тексту різні стилі: напівжирний, курсив, підкреслення тощо. Для цього в HTML існують спеціальні дескриптори, які називають дескрипторами стилів. Всі дескриптори стилів є контейнерами. Основні дескриптори стилів подано у таблиці 4.

Таблиця 4.

Основні дескриптори стилів

ДЕСКРИПТОР	СТИЛЬ
	Напівжирний шрифт
<I>	Курсив
<U>	Підкреслення
<SUB>	Підрядковий текст
<SUP>	Надрядковий текст
<STRIKE>	Перекреслення

Використання дескрипторів стилів розглянуто в наступному прикладі:

```
<HTML>
```

```

<HEAD>
<TITLE> Приклади виділення тексту дескрипторами стилів </TITLE>
</HEAD>
<BODY>
Приклад виділення тексту <B>напівжирним шрифтом</B><BR><BR>
Можна також виділити текст <I>курсивом</I> <BR><BR>
Особливо важливі місця можна <U>підкреслити</U><BR><BR>
Можна створити <SUP>нарядковий </SUP>
або <SUB>підрядковий </SUB> текст<BR><BR>
Крім того, текст можна <STRIKE>перекреслити</STRIKE>.
</BODY>
</HTML>

```

Відображення сторінки з таким кодом подано на Рис. 7

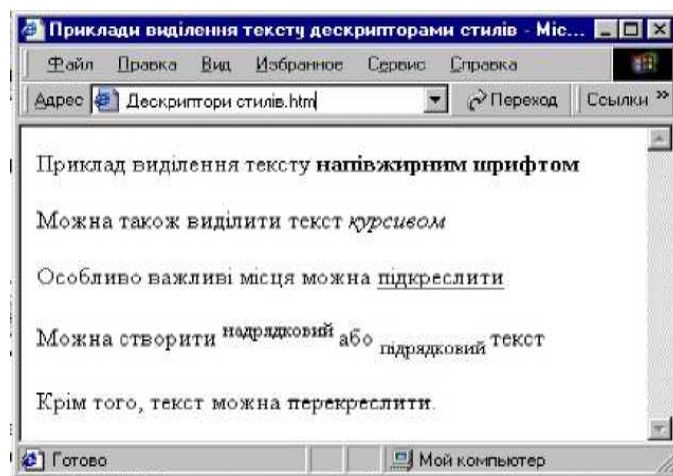


Рис. 7. Стилi форматування

Для вибору шрифту, його розміру та кольору в HTML служить контейнер `... `.

Розмір шрифту можна визначити за допомогою атрибута `SIZE` дескриптора ``:

```
<FONT SIZE=="значення">
```

Атрибут `SIZE` дозволяє вказувати абсолютний (значення від 1 до 7) та відносний (значення від -4 до +4) розміри шрифту по відношенню до розміру основного шрифту, що використовується в основній частині сторінки.

Для зміни розміру основного шрифту використовується елемент `<BASEFONT SIZE=значення>`. Значення атрибута `SIZE` може бути число від 1 до 7. Для браузера стандартним є розмір шрифту 3.

Атрибут `FACE` дескриптора `` дозволяє вказати браузеру шрифт, яким необхідно відображати документ. Якщо розробник сторінки не знає, які шрифти можуть бути на комп'ютері користувача, то в атрибуті `FACE` можна вказати через кому декілька назв шрифтів, наприклад:

Браузер переглядає список шрифтів зліва направо і використовує перший, який встановлено на комп'ютері користувача. Якщо браузер не знаходить на комп'ютері користувача жодного шрифту, що вказані в атрибуті FACE, документ відображається шрифтом, який задано в роузера користувача за замовчуванням.

За допомогою дескриптора можна також змінити колір шрифту. Для цього використовується атрибут COLOR, наприклад, код

текст

задає вказівку браузеру відобразити слово «текст» зеленим кольором. Детальніше про використання кольору у Web-документах розглянуто в розділі Web-графіка.

Крім дескрипторів у мові HTML існують також спеціальні символи форматування. Одним з таких символів є — нероздільний пропуск — ` `. Використання цього символу в коді HTML-файла забезпечує появу на Web-сторінці одного пропуску. Наприклад, за кодом:

[illegible]

браузер між словами «розмір» і «шрифту» помістить 6 пропусків* У таблиці 2.2.4 описано коди деяких спеціальних символів.

Таблица 5.

Коди спеціальних символів

КОДИ	СИМВОЛИ
©, ®	©, ®
¼,	1/4, 1/2, 3/4
&pi	π
&le. &de	◊
&	&

Графічні об'єкти. Важливу роль в оформленні Web-сторінок відіграють графічні зображення. Включення їх у HTML-документи дозволяє надати сторінці «настрою» (ліричного, гумористичного, ділового), подати інформацію, яку неможливо отримати з тексту, розбити сторінку на «теми», що дозволить користувачеві краще орієнтуватися в матеріалі. Для того, щоб сторінка подобалась відвідувачам, необхідно добитись оптимального співвідношення форми (дизайну) і змісту.

Швидкість появи зображення на екрані залежить від розміру файла зображення. Графіка, яка повільно завантажується, може «відлякати» відвідувачів сторінок. Для зменшення розміру файла необхідно використовувати зображення типу JPEG або GIF. Конвертацію графіки у ці формати можна здійснити за допомогою редакторів растрових зображень. Допустимими зображеннями є такі,

розмір яких не перевищує 40-50 Кбайт.

На HTML-сторінках графіку можна розміщувати різними способами.

Вбудовані зображення — це графічні зображення, які завжди залишаються в одному і тому самому місці сторінки (справа від тексту чи іншого об'єкта, що безпосередньо передує зображенню) і не обрамлюються текстом. Для того, щоб помістити на Web-сторінці вбудоване зображення використовується елемент . Дескриптор є одиночним. В атрибуті SRC (від слова source) необхідно вказати повну або відносну URL-адресу файла зображення. Наприклад, код:

<P>Це зображення нової моделі модема

цей текст розміщується одразу після малюнка
використано для розміщення на сторінці зображення, файл якого знаходиться в тому самому каталозі, що і сам HTML-файл (Рис.2.2.8).

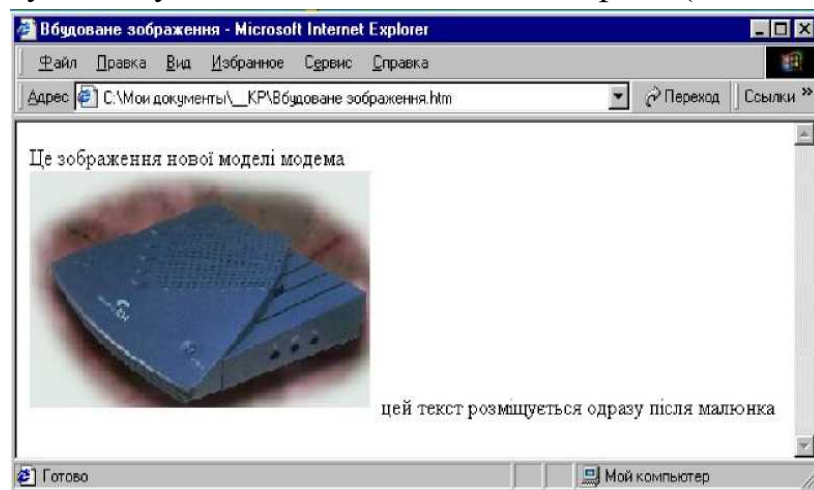


Рис. 8. Графічне зображення на Web-сторінці

За замовчуванням текст вирівнюється по нижній частині зображення. Для зміни типу вирівнювання використовують дескриптор IMG з атрибутом ALIGN. Нижче наведені значення, які можна присвоїти цьому атрибуту:

Таблиця 6

Значення атрибуту	Призначення атрибуту
TOP	вирівнює текст по верхньому краю зображення;
MIDDLE	вирівнює текст по середній частині зображення;
BOTTOM	вирівнює текст по нижньому краю зображення

Наведемо код документа, що вміщує три зображення, для кожного з яких використовується один з варіантів вирівнювання. На Рис. 9 подано Web- сторінку, створену на основі цього, коду.

<HTML>
<HEAD>
<TITLE>

Використання атрибута ALIGN в дескрипторі IMG

</TITLE>

</HEAD>

<BODY>

<P>

Цей текст вирівнюється по верхньому краю зображення </P>

<P>

Цей текст вирівнюється по середній частині зображення </P>

<P>

Цей текст вирівнюється по нижньому краю зображення </P> </BODY>

</HTML>

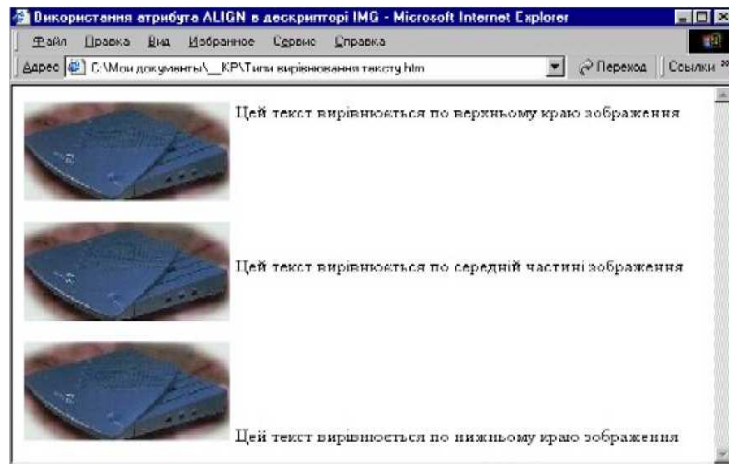


Рис. 9. Вирівнювання тексту відносно зображення

Плаваючі зображення — це зображення, що не прив'язуються до одного рядка тексту, а ніби «плавають» вздовж одного з полів і обрамлюються текстом.

Для створення плаваючих зображень використовується дескриптор з атрибутом ALIGN. Для плаваючих зображень значення цього атрибута можуть бути LEFT (зображення плаває вздовж лівого краю тексту) або RIGHT (зображення плаває вздовж правого краю тексту). Наприклад:

Нижче наводиться код HTML-документа, в якому поміщено плаваюче зображення.

<HEAD>

<TITLE>

Використання атрибута ALIGN в дескрипторі IMG

</TITLE>

</HEAD>

<BODY>

<P> </P>

Цей текст розміщується праворуч від зображення і нижче

Цей текст розміщується праворуч від зображення і нижче

Цей текст розміщується праворуч від зображення і нижче

Цей текст розміщується праворуч від зображення і нижче

Цей текст розміщується праворуч від зображення і нижче

 Цей текст розміщується праворуч від зображення і нижче

 Цей текст розміщується праворуч від зображення і нижче

 Цей текст розміщується праворуч від зображення і нижче
 </BODY>
 </HTML>

На Рис.10 подано інтерпретацію цього коду браузером.

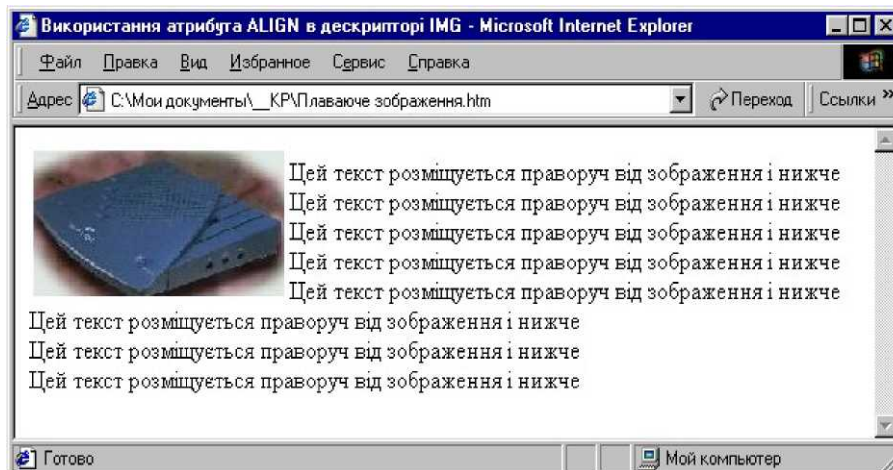


Рис.10. Плаваюче зображення

У дескрипторі можна використовувати атрибути HEIGHT і WIDTH для визначення висоти та ширини зображення. За допомогою цих атрибутів «повідомляється» розмір зображення. Це дозволяє браузеру підготувати макет сторінки та розмістити текст до того, як будуть завантажені всі зображення. За допомогою цих атрибутів можна також змінювати розміри зображення.

Так, наприклад, для збільшення зображення необхідно в атрибутах HEIGHT і WIDTH вказати більші величини, ніж реальні розміри зображення (які можна визначити з допомогою графічного редактора). Зауважимо, що зменшувати розмір зображення таким методом не доцільно, оскільки розмір файла, а отже і час завантаження при цьому не змінюються.

За допомогою атрибута BGCOLOR дескриптора <BODY> можна визначити колір фону сторінки, вказавши у ньому шестицифрове шістнадцяткове число. Цей атрибут має такий формат:

```
<BODY BGCOLOR=#rrggbb>
```

rr, *gg*, *bb* — двоцифрові шістнадцяткові числа, які визначають відповідно червону, зелену та голубу складові кольору фону.

Наприклад:

```
<BODY BGCOLOR=#000000><!-- задається чорний колір фону ^
<BODY BGCOLOR=#ffffff><!-- задається білий колір фону ^
<BODY BGCOLOR=#ff0000><!-- задається червоний колір фону ^
<BODY BGCOLOR=#00ff00><!-- задається зелений колір фону ^
<BODY BGCOLOR=#0000ff><!-- задається синій колір фону ^
```

Для визначення основних кольорів можна використовувати також зарезервовані імена (таблиця 2.2.6.).

Таблиця 7.

Зарезервовані імена кольорів

КОЛІР	ШІСТНАДЦЯТКОВИЙ ЕКВІВАЛЕНТ
BLACK (чорний)	#000000
MAROON (темно-бордовий)	#800000
GREEN (зелений)	#008000
OLIVE (оливковий)	#808000
NAVY (темно-синій)	#000080
PURPLE (фіолетовий)	#800080
TEAL (зеленувато-блакитний)	#008080
GRAY (сірий)	#808080
SILVER (сріблястий)	#C0C0C0
RED (червоний)	#FF0000
LIME (лимонний)	#00FF00
YELLOW (жовтий)	#FFFF00
BLUE (синій)	#0000FF
FUCSIA (фуксія)	#FF00FF
AQUA (морська хвиля)	#00FFFF
WHITE (білий)	#FFFFFF

Наприклад,

```
<BODY BGCOLOR="RED">
```

<!--задається червоний колір фону ^

Для створення графічного фону використовується атрибут

BACKGROUND дескриптора <BODY>. У цьому атрибуті потрібно вказати ім'я файла графічного зображення. Наприклад:

```
<BODY BACKGROUND=="my__picture.gif">
```

Колір основного тексту можна змінити за допомогою атрибута TEXT дескриптора <BODY>. Колір задається так, як і у випадку зміни кольору фону.

Зображення-гіперпосилання — це зображення на Web-сторінці, клацнувши на якому, можна відкрити нову Web-сторінку або службу Internet. Наприклад, можна помістити на Web-сторінці зображення поштової скриньки, клацнувши на якому користувачі зможуть відправляти повідомлення електронної пошти. HTML-код зображення-гіперпосилання має вигляд:

```
<A HREF="URL"><IMG SRC^"URL-файла зображення"></A>
```

З метою швидшого завантаження Web-сторінки має сенс створювати зменшені копії зображень (мініатюри зображень), які є гіперпосиланнями на

відповідні їм зображення-оригінали. Мініатюри завантажуються швидко, даючи користувачеві уявлення про зображення-оригінали. Доступ до зображення-оригіналу можна отримати у будь-який момент, клацнувши лівою кнопкою миші на зображенні-гіперпосиланні.

Щоб використовувати мініатюри, необхідно мати дві копії зображення — велике зображення-оригінал і зменшене зображення. Для зменшення розмірів зображень використовують можливості графічних програм. Щоб зв'язати мініатюру з оригіналом необхідно помістити дескриптор, що задає мініатюру, всередині дескриптора `<A HREF> `, який є гіперпосиланням на велике зображення, наприклад:

```
<A HREF="big.gif"><IMG SRC="small.gif"></A>
```

Тут задано вказівку роузера розмістити на Web-сторінці зображення (файл small.gif), що є гіперпосиланням на зображення, задане файлом big. Gif.

Гіперпосилання на веб-сторінці

Гіперпосилання — це основа HTML. За допомогою гіперпосилань користувачі можуть переходити з однієї сторінки до іншої.

Існує три основних типи гіперпосилань: внутрішні, зовнішні та відносні.

Внутрішні гіперпосилання — це посилання на об'єкти в межах одного документа. З їх допомогою користувач може переміщатись всередині однієї Web-сторінки. Такі посилання доцільно використовувати на довгих сторінках, щоб мати можливість швидко переміщуватись між їх розділами.

Зовнішні гіперпосилання — це посилання на сторінки, розміщені на іншому Web-сервері.

Відносні або локальні посилання — це посилання на Web-сторінки, розміщені на тому ж сервері. Адреси цих посилань задаються відносно адреси Web-сторінки, на якій вони поміщені.

У HTML для зовнішніх гіперпосилань використовується синтаксис:

```
<A HREF=»URL»>Текст посилання </A>
```

де URL — повна адреса документа, на який здійснюється посилання. Текст, поміщений у контейнер ` `, зображується в браузері з підкресленням та іншим кольором (найчастіше синім). Клацнувши на ньому лівою кнопкою миші, даємо вказівку браузеру завантажити сторінку, розміщену за адресою URL.

Наприклад, вказівкою:

```
<A HREF=»http://www.firma.com/ibm/computers.html»> Комп'ютери компанії IBM </A>
```

створюється гіперпосилання на Web-сторінку, розміщену за адресою

<http://www.ibm.com/ibm/computers.html>.

Зв'язуючи гіперпосиланнями сторінки одного сервера, немає необхідності включати в URL доменне ім'я. Наприклад, вказівкою:

```
<A HREF="computers.htm">Комп'ютери виробництва IBM</A>
```

створюється гіперпосилання на файл computers.htm, який повинен знаходитися на тому самому сервері і в тому самому підкаталозі, що і файл, в якому знаходиться це гіперпосилання.

Якщо файл знаходиться в іншому підкаталозі, то необхідно вказати шлях до нього, наприклад:

```
<A HREF="ibm/computers.htm">Комп'ютери виробництва IBM</A>
```

Є й інші можливості для спрощення записів довгих адрес URL. Так, наприклад, HTML-код:

```
<HTML>
<HEAD>
<TITLE> Основні розділи Web-сайта ВНЗ</TITLE>
</HEAD>
<BODY>
<A HREF="http://www.npu.kiev.ua/REKTORAT/in-rek.htm">Ректорат</A>
<A HREF="http://www.npu.kiev.ua/FAKULT/in-fak.htm">Факультети</A>
<A HREF="http://www.npu.kiev.ua/ASPIRANT/index.htm">Аспірантура</A>
<A HREF="http://www.npu.kiev.ua/ABITUR/abit-bez.htm">Лбітупісі ітү</Д>
</BODY>
</HTML>
```

можна записати компактніше:

```
<HTML>
<HEAD>
<TITLE> Основні розділи Web-сайта ВНЗ</TITLE>
<BASE HREF="http://www.npu.kiev.ua/">
</HEAD>
<BODY>
<A HREF="REKTORAT/in-rek.htm">Ректорат</A>
<A HREF="FAKULT/in-fak.htm">ОаКynbTeTH</A>
<A HREF="ASPIRANT/index.htm">Аспірантура</A>
<A HREF="ABITUR/abit-beg.htm">Абітүріентү</A>
</BODY>
</HTML>
```

Тут використовується дескриптор <BASE>, в якому вказано базову адресу URL, відносно якої подаються всі інші відносні адреси Web-сторінок. Зазначимо, що цей дескриптор використовується всередині контейнера <HEAD> </HEAD>, областю дії його є вся сторінка.

Створюючи великі Web-сторінки, необхідно також забезпечити можливість для користувача легко переміщуватись з одного їх розділу до іншого. Це досягається завдяки внутрішнім гіперпосиланням. Для реалізації такого посилання необхідні два елементи: посилання, яке вказує, куди треба перейти, і мітка, яка фіксує місце переходу.

Посилання задається кодом:

```
<A HREF = "і ківвл мітки">Текст посилання</A>
```

Мітка також використовує контейнер <A> і задається так:

```
<A NAME = "#назва мітки">текст</A>
```

Назва мітки — це довільний текст, який не появляється на екрані й однозначно визначає місце переходу. Якщо на сторінці є кілька міток, то всі вони повинні мати різні назви. Символ «#» при описі посилання вказує, що за ним записана назва мітки, а не назва файла.

Приклад використання внутрішніх гіперпосилань:

```
<HTML>
<HEAD>
<T^E>Колекція програм мовою VisualBasic</TITLE>
</HEAD>
<BODY>
Нижче наводяться програми мовою VisualBasic:
<BR>
<!-- далі створюємо внутрішні гіперпосилання ^
<A HREF="#графік">Побудова графіків функцій</A><BK>
<A HREF="#масив">Сортування масиву</A><BR>
<A HREF="#рівняння">Розв'язування квадратних рівнянь</A>
<BR>
<P>
<!-- далі описуємо місця, на які здійснюються посилання^
<A NAME="графік"> Побудова графіків функцій - приклад програми</A>
<BR>
(текст програми)
<BR><BR>
<A NAME="масив"> Впорядкування масиву - приклад програми</A>
<BR>
(текст програми)
<BR><BR>
<A NAME="рівняння"> Розв'язування квадратних рівнянь - приклад програми</A>
<BR>
(текст програми)
</BODY>
</HTML>
```

Можна зробити так, щоб після натискування на деякому посиланні звучала

музика або відтворювався відеофільм. Посилання на мультимедійні файли .avi, .wav створюються так само, як посилання на HTML-документи. В атрибуті HREF контейнера `<A>` необхідно вказати ім'я файла, в якому міститься потрібна мелодія або відеофільм. Наприклад:

```
<P>Натисніть<A HREF="my_music.wav">Тут</A>, щоб почути мою улюблену композицію
```

Файл my_music.wav повинен знаходитись на тому самому сервері, у тому самому каталозі, що і HTML-файл, з якого він викликається. Для створення гіперпосилань на файли мультимедіа, розміщені на інших серверах Internet, використовуються повні URL-адреси.

Часто виникає необхідність включити у Web-сторінку посилання на адресу електронної пошти. Для цього в атрибуті HREF необхідно задати адресу електронної пошти у вигляді «mailto: адреса». Наприклад, щоб створити посилання на адресу moi@npu. Kiev. Ua, необхідно у код сторінки помістити такий дескриптор:

```
<A HREF="mailto: moi@npu.kiev.ua">moi@npu.kiev.ua</A>
```

згідно з яким роузера відображає текст «moi@npu.kiev.ua» як гіперпосилання. Якщо клацнути на ньому лівою кнопкою миші, браузер відкриє поштову програму в режимі створення листа, і в полі «Адреса:» помістить адресу moi@npu.kiev.ua. Текст, що поміщається в контейнер `<A>`, може бути будь-яким, проте доцільно використовувати адресу Email, оскільки не всі користувачі сторінки можуть користуватися поштовими програмами, налагодженими на комп'ютері, з якого переглядається сторінка. Їм необхідно прочитати реальну адресу електронної пошти.

Динамічні ефекти та засоби їх створення

Динамічними називаються ефекти, коли графічні зображення на Web- сторінці змінюються з часом, елементи сторінки змінюють розміри або навіть свій зміст після клацання над ними мишею, текст «біжить» уздовж екрана тощо.

Розглянемо ефект рухомого тексту у смузі, що має висоту **HEIGHT** = "висота в пікселях" і фон **BGCOLOR** = "колір фону". Ефект створюється за допомогою парного тега `<MARQUEE>...</MARQUEE>`, а саме:

```
<MARQUEE BGCOLOR="aqua" HEIGHT = 40> Особиста сторінка Ваше прізвище та ім 'я</MARQUEE>. За допомогою цього тега можна заставити рухатись і графічний об'єкт.
```

Текст "*Особиста сторінка Ваше прізвище та ім 'я*" буде пробігати у смузі справа наліво, заходитиме за край екрана і з'являтимуться знову справа. Даний тег рекомендують застосовувати до заголовків сторінки.

Ефект відбивання від країв екрана забезпечує параметр **BEHAVIOR** = "**alternate**", а зупинити рядок біля лівого краю екрана може значення цього параметра "**slide**". Значення "**right**" параметра **DIRECTION** забезпечить ефект руху

у протилежний бік.

Обмеження кількості проходів задається параметром **LOOP^u**, де *цифра* - кількість проходів. Смугу можна центрувати за допомогою параметрів **HSPACE** і **VSPACE**. Швидкість руху задає параметр

SCROLLAMOUNT=число, де *число* - величина швидкості руху, де конкретне значення вибирають з діапазону від 1 (повільно) до 10 (швидко).

Приклад. У тексті коду веб-сторінки у файлі *file1.htm* вставте рухомий рядок, наприклад, такого змісту:

```
<MARQUEE BGCOLOR="aqua" HEIGHT = 40 LOOP=5> Приватна сторінка Ваше прізвище та ім'я</MARQUEE>.
```

1) Далі змініть цей рядок на такий:

```
<MARQUEE BGCOLOR="aqua" HEIGHT = 50 LOOP=4 DIRECTION ="rihgt" SCROLLAMOUNT=5> <FONT  
FACE = "Courier" SIZE = +4 COLOR = "red"> Особиста сторінка Ваше прізвище та ім'я  
</FONT></MARQUEE> перегляньте веб-сторінку.
```