

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА
ЕКОНОМІЧНИЙ ФАКУЛЬТЕТ
КАФЕДРА БІЗНЕС-СТАТИСТИКИ І ЕКОНОМІЧНОЇ КІБЕРНЕТИКИ

Створення HTML-сторінки.

**Методичні рекомендації для виконання завдань з самостійної
роботи студентів за курсом «Економічна інформатика»**

Вінниця 2020

004.43:330.47(075.8)

Г 276

Створення HTML-сторінки. Методичні рекомендації для виконання завдань з самостійної роботи студентів за курсом «Економічна інформатика». – Вінниця: ДонНУ імені Василя Стуса, 2020. – 45 с.

Автор: *І. Г. Гевлич*, канд. техн. наук, доцент

Рецензенти: *Н. В. Дутова*, канд екон. наук, доцент;
О. В. Таранич, канд. екон. наук, доцент.

Рекомендовано вченою радою економічного факультету (протокол № 6 від 31 січня 2020 р.)

Методичні рекомендації підготовлені відповідно до робочої програми за курсом «Економічна інформатика». Пояснюють сутність завдань з самостійної роботи студентів за темою «Основи створення html-сторінок», містять рекомендації щодо порядку їх виконання, перелік рекомендованих посилань.

Рекомендуються для студентів СО «Бакалавр» всіх форм навчання, викладачів.

© І. Г. Гевлич, 2020

© ДонНУ імені Василя Стуса, 2020

ЗМІСТ

ВСТУП	4
I. Завдання для з самостійної роботи студентів за темою «Створення HTML-сторінки».	6
II. Методичні рекомендації для виконання завдання з самостійної роботи студентів за темою «Створення HTML-сторінки»	7
1. Адресація Web-сторінок	7
2. Основні синтаксичні елементи HTML	8
3. Основні елементи Web-сторінки	10
4. Форматування тексту	11
4.1. Рядки й абзаци. Дескриптори <P> і 	11
4.2. Заголовки. Контейнер <H>	13
4.3. Форматування тексту за допомогою контейнерів фізичних стилів	15
4.4. Форматування тексту за допомогою контейнерів логічних стилів	16
4.5. Зміна розміру шрифту. Дескриптори FONT і BASEFONT	18
4.7. Використання попередньо відформатованого тексту. Контейнер <PRE>	20
4.8. Створення горизонтальних ліній. Дескриптор <HR>	21
5. Гіпертекстові посилання (зв'язки). Контейнер <A>	23
5.1. Зовнішні посилання	24
5.2. Внутрішні посилання	25
5.3. Відносні посилання	26
6. Використання графіки та кольору	27
6.1. Способи завдання кольору	28
6.2. Зміна кольорів тексту і фону. Атрибути TEXT, COLOR, BGCOLOR, LINK, ALINK, VLINK, BACKGROUND	29
6.3. Вставка зображень. Дескриптор 	30
7. Списки. Контейнери , , <MENU>,	32
8. Таблиці	35
8.1. Структура таблиці	36
8.2. Контейнер завдання таблиці <TABLE>	37
8.3. Контейнер завдання рядку <TR>, <TD>, <TH>	37
8.4. Контейнери завдання комірок <TD>, <TH>	38
9. Використання відеороликів на веб-сторінці	40
10. Використання аудіозаписів на веб-сторінці	41
11. Використання елементів анімації	42
11.1. Gif-Анімація	42
11.2. Рухомий рядок	43
СПИСОК ВИКОРИСТАНИХ ПОСИЛАНЬ	45

ВСТУП

Метою вивчення навчальної дисципліни «**Економічна інформатика**» є формування у майбутніх фахівців теоретичних знань і практичних навичок застосування сучасних інформаційно-комунікаційних технологій при дослідженні соціально-економічних систем.

У результаті вивчення навчальної дисципліни студент повинен набути таких **результатів навчання**:

- 1) набуття навичок використання прикладних систем оброблення економічних даних під час розв'язування завдань фахового спрямування;
- 2) використання можливостей програмного забезпечення для реалізації прикладних завдань, що розраховані на конкретного споживача;
- 3) аналіз, оцінка фактів і прогнозування очікуваних результатів прийнятих рішень;
- 4) забезпечення технічної підтримки наукової й творчої діяльності студентів.

Навчальна дисципліна формує **міждисциплінарні взаємозв'язки** із іншими дисциплінами, такими як: вища математика, статистика, теорія ймовірностей й математична статистика, регіональна економіка.

Вивчення навчальної дисципліни «Економічна інформатика» передбачає формування та розвиток у студентів **компетентностей та програмних результатів навчання** згідно із відповідними освітніми програмами.

HTML (Hypertext Markup Language) – стандартна мова розмітки для створення веб-сторінок і веб-додатків. Використання команд мови HTML є визначальним при з'єднанні в єдиний інформаційний простір інформації будь-якої модальності, представленої в цифровій формі (текст, аудіо-, відеофайли, графіка, анімація тощо). Крім цього, мова HTML дозволяє задати базові параметри документа як веб-сторінки, зокрема:

- як повинні виглядати на екрані користувача ті чи інші елементи тексту;
- як смислові, структурні та композиційні елементи тексту мають бути розміщені на двовимірному просторі веб-сторінки;
- яким чином текстові елементи будуть поєднуватися з інформаційними елементами іншої модальності – картинками, діаграмами, відеороликами, анімацією тощо;
- які елементи в тексті документа пропонують інтерактивні дії користувача.

Тема «Основи створення html-сторінок» є однією з визначальних у навчальній дисципліні «Економічна інформатика», а знання команд HTML

формує практичні навички, необхідні сучасному фахівцеві-економісту.

Історія HTML (а разом з ним і WWW) почалася в кінці 1980-х років, коли у вчених з Європейської лабораторії елементарних часток (CERN) виникла необхідність обмінюватися різними документами за допомогою мережі Інтернет. Розв'язок цієї проблеми був знайдений у 1989 році співробітником CERN Бернерс-Лі, який на основі мови SGML створив HTML. Він же і реалізував першу програму для перегляду HTML-документів – браузер. Ідея з гіпертекстом була дуже швидко прийнята суспільством, і незабаром в Інтернеті була створена мережа гіпертекстових документів, яку назвали World Wide Web.

У сучасному світі високих технологій є нормою мати власну Web-сторінку. Це дозволяє рекламувати продукцію чи послуги, повідомляти інформацію про себе, шукати і знаходити роботу, обмінюватися повідомленнями. Для створення Web-сторінки можна скористатися послугами професіоналів, однак цікавіше створювати такі сторінки за власним смаком, використовуючи власну фантазію і здібності.

Для створення Web-сторінок необхідно знати мову гіпертекстової розмітки (HTML). Простота цієї мови робить її доступною для широкого кола користувачів.

I. Завдання для з самостійної роботи студентів за темою «Створення HTML-сторінки».

Розробити сайт автобіографічного змісту, що складається з п'яти окремих сторінок:

- перша сторінка має містити інформацію про автора: ПІБ, фотографію, автобіографічний текст;
- друга сторінка має розкривати інформацію про родину автора;
- третя сторінка має бути присвячена навчанню автора;
- четверта сторінка має розкривати хобі автора;
- п'ята сторінка має розкривати мрії, плани, перспективи автора.

Кожна сторінка має включати текстову та графічну інформацію (фотографії, анімацію), а також мати рухомий рядок.

Усі сторінки повинні бути з'єднані переходами для навігації сайтом та мати внутрішні та зовнішні посилання.

На сайті має бути щонайменше одна таблиця, нумерований та маркований списки, аудіо- та відеоінформація.

Інформація має бути унікальною, не запозиченою з інших джерел, та міститися у папці, названій прізвищем автора.

II. Методичні рекомендації для виконання завдання з самостійної роботи студентів за темою «Створення HTML-сторінки».

1. Адресація Web-сторінок

Будь-які інформаційні ресурси в комп'ютерних мережах зберігаються на спеціальних комп'ютерах-серверах. Існує декілька різновидів таких серверів, що різняться між собою за способом зберігання, обробки і передачі інформації. Web-сторінка представляє собою файл, що зберігається на комп'ютері. Як правило, це Web-сервер. Для перегляду Web-сторінки використовуються браузері – програми для перегляду гіпертекстових сторінок. Прикладами таких програм є Internet Explorer, Opera, Mozilla тощо.

Будь-яка Web-сторінка, представлена в мережі, повинна мати свою адресу (Uniform Resource Locators - URL). Ця адреса вказана в адресному рядку браузера і пересилається комп'ютером користувача на відповідний Web-сервер. Відшукавши за вказаною адресою відповідний файл, Web-сервер відсилає його на комп'ютер користувача. Браузер комп'ютера користувача відображає вміст цього файлу у відповідності з присутніми в цьому файлі правилами розмітки тексту.

Адреса, яка вказується в адресному рядку браузера, має наступний вигляд:

scheme://host:port/шлях_до_файлу/ім'я_файлу

Тут scheme визначає протокол, який використовується для передавання файлу. Після цього параметра стоять дві крапки. Наприклад:

http – використовується протокол передавання гіпертексту;

ftp – використовується протокол обміну файлами, зазвичай використовується для зв'язку з ftp-серверами;

mailto – використовується набір протоколів для електронної пошти;

news – використовується протокол для організації телеконференції або передавання розсилання новин;

telnet – використовується протокол для організації діалогової сесії;

Всі згадані тут протоколи є частиною протоколу інтернету TCP/IP:

Host – адреса комп'ютера-сервера, на якому зберігається інформація;

Port – десяткове число, що означає порт (логічну адресу), за котрим здійснюється передача інформації;

шлях_до_файлу/ім'я_файлу – вказує ім'я файлу, який передається, і повний шлях до нього. Якщо параметр ім'я_файлу відсутній, то передається файл, визначений за умовчужанням. Зазвичай файлом за умовчужанням призначається файл index.htm чи main.htm.

Адреса комп'ютера складається з декількох частин:

ім'я_комп'ютера.ім'я_підмережі1.ім'я_підмережі2.

У всесвітній мережі WWW всі комп'ютери-сервери мають одне й те ж саме ім'я WWW. Для зручності класифікації і пошуку інформації вся всесвітня мережа розбита на логічні підмережі (домени). Такими доменами, наприклад, є підмережі, що складаються з комп'ютерів, розташованих на територіях окремих країн, підмережі, що поєднують ресурси учбових закладів, комерційних організацій, державних закладів, армії тощо.

Перелік імен деяких підмереж:

Ua – на території України;

Com – мережа, яка містить інформацію комерційного характеру;

Edu – мережа учбових закладів;

Gov – державні організації;

Org – некомерційні організації;

Net – провайдери Інтернет.

Перелік деяких адрес комп'ютерів всесвітньої мережі:

www.donpu.edu.ua – сервер Донецького національного університету імені Василя Стуса;

www.facebook.com – соціальна мережа;

www.google.com – пошуковий сервер та інші інтернет-сервіси;

www.ukr.net – сервер одного з провайдерів України.

Питання для самоперевірки:

1. Які програми використовуються для перегляду Web-сторінок?
2. Що необхідно для ідентифікації Web-сторінки в мережі?
3. Які протоколи використовують для передачі даних?
4. Якщо в URL-адресі відсутнє ім'я файлу, то який файл передається за умовчужанням?
5. Які домени характеризують тип організації або країну?
6. Наведіть приклади пошукових серверів.

2. Основні синтаксичні елементи HTML

Будь-який гіпертекстовий документ представляє собою текстовий файл. Це означає, що для його створення і редагування може бути використаний такий текстовий редактор, наприклад, як Notepad в системі Windows. Для того, щоб файл, який містить гіпертекст, переглядався одним з браузерів автоматично, він повинен мати розширення htm чи html. Такий файл може містити символи стандартів ISO Latin 1 (кодування ISO 8859-1), ISO 10646 і Unicode.

Синтаксис – це правила, за якими будуються вирази на мові, елемент – це конструкція мови HTML, яка містить дані. HTML містить в собі різні типи елементів, що дозволяють задавати абзаци, гіпертекстові посилання, списки, таблиці, зображення тощо.

Основною синтаксичною одиницею мови гіпертекстової розмітки є **дескриптор** (tag). Дескриптори вміщуються в кутові дужки (<>) – знаки «більше» і «менше». Слід звернути увагу, що кутові дужки, які входять до будь-якого дескриптора, є парними, тобто за відкриваючою дужкою обов'язково повинна бути присутньою закриваюча дужка. В дужках вказується ім'я тегу. Тег може містити один чи декілька атрибутів.

Існує два типи дескрипторів: одиночні дескриптори і контейнери. Контейнер – це дескрипторна пара, яка складається з відкриваючого і закриваючого дескрипторів. Закриваючий дескриптор відрізняється від відкриваючого тим, що перед іменем дескриптора стоїть символ «/».

Приклади одиночних дескрипторів і контейнерів:

<P> – одиночний дескриптор, що означає початок нового абзацу;

 – одиночний дескриптор, що означає початок нового рядка;

<H1> – текст заголовка, </H1> – контейнер, який означає конструкцію, що містить заголовок;

<! коментар > – одиночний дескриптор, що містить коментар;

<!-- текст коментарю --> – дескриптор, що містить коментар, який може займати декілька рядків.

Всі дескриптори записуються як великими, так і малими буквами латинського алфавіту. Тут і надалі для зручності читання всі дескриптори, їхні атрибути і значення атрибутів наведені великими літерами в тексті, та малими в прикладах.

Елементи можуть містити параметри, які називаються **атрибутиами**. Атрибути можуть бути встановлені за умовчужанням або автором веб-сторінки. Пара атрибут/значення вказуються в початковому дескрипторі елементу перед символом >. Наприклад, <BODY bgcolor = «#FF0000»>. Тут вказаний атрибут bgcolor в елементі BODY, завдяки якому колір сторінки буде червоний. В початковому дескрипторі може бути вказано декілька атрибутів, розділених пробілом, причому порядок їх запису не має значення. Важливим моментом HTML є використання вкладених елементів. Елемент, який знаходиться всередині іншого елементу, називається вкладеним. Наприклад, задамо нарис шрифту <I>Курсив-Полужирный курсивКурсив</I>. При використанні вкладеності слід пам'ятати, що вкладені елементи закриватися до того, як будуть закритими зовнішні елементи.

Питання для самоперевірки:

1. Яке розширення мають файли Інтернету?
2. Що є основною синтаксичною одиницею мови HTML?
3. Де можна редагувати Web-сторінку?
4. Які типи дескрипторів існують? Якими є їхні відмінності?
5. Як на мові HTML позначаються коментарі?
6. Чим відрізняються дескриптори
 і
?

3. Основні елементи Web-сторінки

Усі Web-сторінки повинні мати загальну структуру.

Наприклад, нижче описані обов'язкові елементи структури будь-якої Web-сторінки:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title> назва сторінки </title>
</head>
<body>
  вміст сторінки
</body>
</html>
```

З цього прикладу видно, що весь файл з точки зору синтаксису є контейнером HTML, що містить в собі контейнер HEAD, який містить контейнер TITLE і контейнер BODY.

Контейнер TITLE містить назву сторінки, що відображене як заголовок вікна браузера. Ім'я документу в елементі TITLE використовується браузерами й іншими програмами, слугує засобом пошуку, що повертається.

Контейнер HEAD, окрім контейнера TITLE, може містити наступні елементи:

- елемент ISINDEX (використовується рідко);
- елемент BASE (задає точну базову адресу посилань на URL);
- елемент META (задає різну метаінформацію на кшталт дати закінчення терміну документа);
- елемент LINK (задає метаінформацію про зв'язки документа з іншими документами);
- елементи STYLE і SCRIPT.

Контейнер BODY містить в собі власне текст Web-сторінки. Початковий дескриптор цього контейнера може містити ряд атрибутів:

- BGCOLOR - фоновий колір документа;
- TEXT - колір для тексту документа;
- LINK - колір для не відвіданого гіпертекстового посилання;
- VLINK - колір для відвіданого гіпертекстового посилання;
- ALINK - колір для активного гіпертекстового посилання;
- BACKGROUND – URL фонового образу.

В HTML документі дозволений тільки один елемент BODY, і він повинен знаходитися після елементу HEAD.

Питання для самоперевірки.

1. Яку структуру мають Web-сторінки?
2. Яким є призначення дескриптора TITLE?
3. Де розташований текст Web-сторінки?
4. В чому різниця між атрибутами LINK, VLINK, ALINK?
5. Який контейнер є головним і містить решту дескрипторів?

4. Форматування тексту

Створюючи текстовий документ, користувач зазвичай використовує ті чи інші елементи форматування: абзаци, відступи, вирівнювання, різні шрифти тощо. Всі вони досягаються засобами того середовища, в якому власне й набирається текст. Браузер, читаючи створений текст як гіпертекстовий документ, ігнорує будь-які елементи форматування тексту, що не відносяться до HTML. Наприклад, ігноруються зайві пробіли (тобто якщо в тексті підряд стоїть декілька пробілів, то у вікні браузера буде відображений тільки один пробіл), а також символи кінця рядка (натискання клавіші Enter), символи переносу, які вставляються автоматично, виділення абзацу тощо. Для виконання всіх операцій форматування тексту слугують спеціальні дескриптори HTML.

Розмір шрифту для браузера асоціюється з визначеним апаратним способом перетворення його в фізичні розміри, які виражені, наприклад, в пікселях, точках або міліметрах. Слід також пам'ятати, що таке перетворення, як правило, є нелінійним, тобто шрифт розміру 3 необов'язково складає половину шрифту розміром 6.

Слід також пам'ятати, що в різних браузерах одна й та ж сторінка може виглядати по-різному. Крім того, різні версії одного й того ж браузера можуть різним чином відображати вміст Web-сторінок.

4.1. Рядки й абзаци. Дескриптори <P> і

За умовчуванням якщо Web-сторінка містить тільки текст, довжина відображеного рядка тексту у вікні браузера вирівнюється за шириною цього вікна незалежно від того, яка довжина цього рядка у вихідних файлах.

Дескриптор
 використовується для вставки символу початку рядка. Аналогом дії цього дескриптора може слугувати натискання символу Enter в редакторі Notepad.

Дескриптор <P> використовується для вставки символу початку абзацу. В HTML один абзац відокремлюється від іншого більшим міжрядковим інтервалом. Між блоками тексту, якщо розглядати їх з точки зору HTML як абзаци, існує деяка відстань. Такі елементи як відступ чи

виступ, притаманні для виділення конструкції «абзац», наприклад, в Microsoft Word, в HTML не використовуються.

При включенні в код двох дескрипторів
 підряд між блоками тексту виникає більша відстань, ніж при використанні одного дескриптора <P>.

Загалом дескриптори <P> і
 є контейнерами, що вміщують в собі текст, який розглядається як один рядок або один абзац, але кінцеві дескриптори цих контейнерів не є обов'язковими.

При вставленні в текст дескрипторів початку рядка слід пам'ятати, що якщо цей дескриптор буде стояти в кінці дуже довгого рядка, то скоріше за все, рядок при відображенні у вікні браузера буде мати додаткові розриви. Це може виявитися важливим при побудові текстового вмісту сторінки. Оптимальною довжиною рядку є 60-80 символів, хоча цей параметр суттєво залежить від розміру і виду шрифту.

За умовчуванням за горизонталлю текст абзацу вирівнюється за лівим краєм вікна браузера. Для зміни способу вирівнювання в дескрипторі <P> слід використовувати атрибут

ALIGN=вирівнювання

Тут параметр вирівнювання може приймати значення LEFT або CENTER, що призведе до горизонтального вирівнювання за лівим краєм чи за центром вікна браузера.

Нижченаведений приклад ілюструє дескриптори:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title> Приклади використання дескрипторів початку рядка і
початку абзацу </title>
</head>
<body>
  <br> Окремий рядок, вміщений у відповідний контейнер. </br>
  <br> Ще один окремий рядок, вміщений у відповідний контейнер.
</br>
```

В кінці цього рядка стоїть символ початку нового абзацу.

```
<p align=center>
```

Абзац повинний бути вирівняним за центром.

```
</p>
```

```
</body>
```

```
</html>
```

В результаті сторінка буде мати вигляд, зображений на рис. 4.1

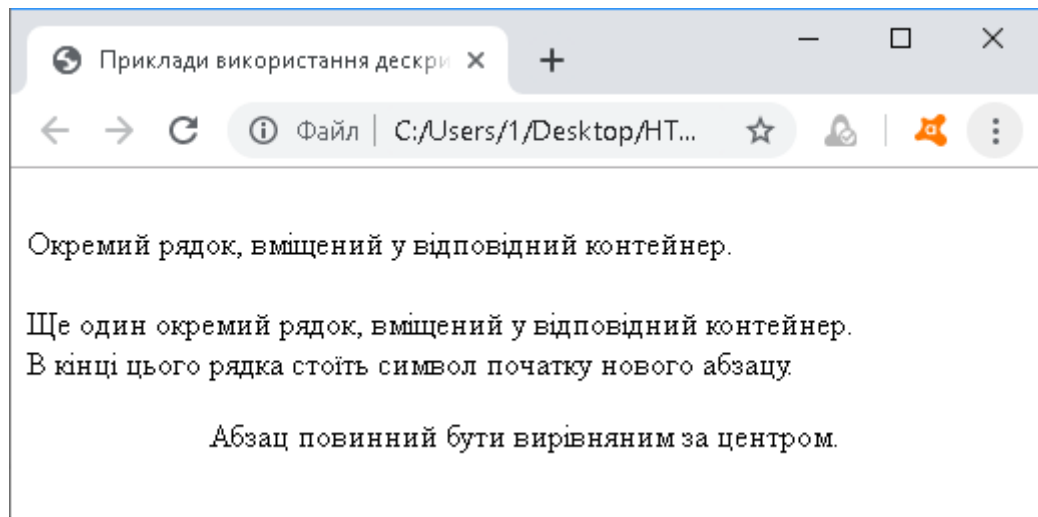


Рисунок 4.1 – Результат використання дескрипторів <P> і

4.2 Заголовки. Контейнер <H>

Конструкція «заголовок» виділяє текст, вміщений в контейнер <H>, використовуючи для цього встановлені для конкретного браузера правила виділення заголовків. В наступний час HTML підтримує шість рівнів заголовків: H1, H2, H3, H4, H5, H6. Заголовок відрізняється від решти тексту видом, розміром і виділенням шрифту.

Нижченаведений приклад містить лістинг вмісту файлу, що ілюструє конструкцію «заголовок»:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title> Приклади заголовків </title>
</head>
<body>
  <h1> Заголовок першого рівня (h1) </h1> <br>
  <h2> Заголовок другого рівня (h2) </h2> <br>
  <h3> Заголовок третього рівня (h3) </h3> <br>
  <h4> Заголовок четвертого рівня (h4) </h4> <br>
  <h5> Заголовок п'ятого рівня (h5) </h5> <BR>
  <h6> Заголовок шостого рівня (h6) </h6> <br>
</body> </html>
```

Наведений приклад демонструє порядок впорядкування заголовків за зростанням розміру шрифту (рис. 4.2).

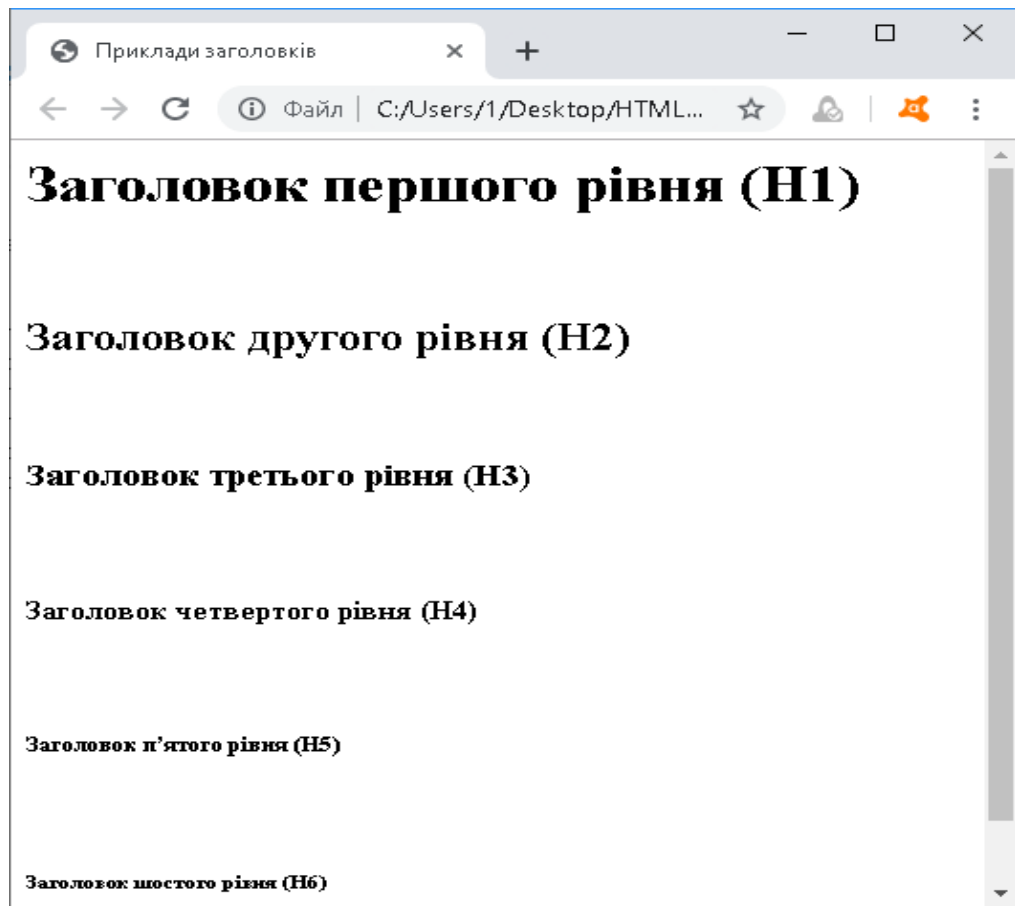


Рисунок 4.2 – Контейнер <H>.

Для горизонтального вирівнювання заголовків відносно вікна браузера в контейнері <H> також використовується атрибут ALIGN= вирівнювання. В попередньому прикладі цей атрибут не використовувався. Це означає, що вирівнювання заголовків відбувалось за лівим краєм (за умовчанням).

Змінимо попередній приклад щодо способу вирівнювання заголовків (рис. 4.3):

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title> Приклади вирівнювання заголовків </title>
</head>
<body>
  <h1 align=justify> Заголовок вирівняний по ширині вікна сторінки
</h1> <br>
  <h2 align=left > Заголовок вирівняний по лівому краю </h2> <br>
  <h3 align=right > Заголовок вирівняний по правому краю </h3>
<br>
  <h4 align= center> Заголовок вирівняний по центру </h4> <br>
</body>
</html>
```

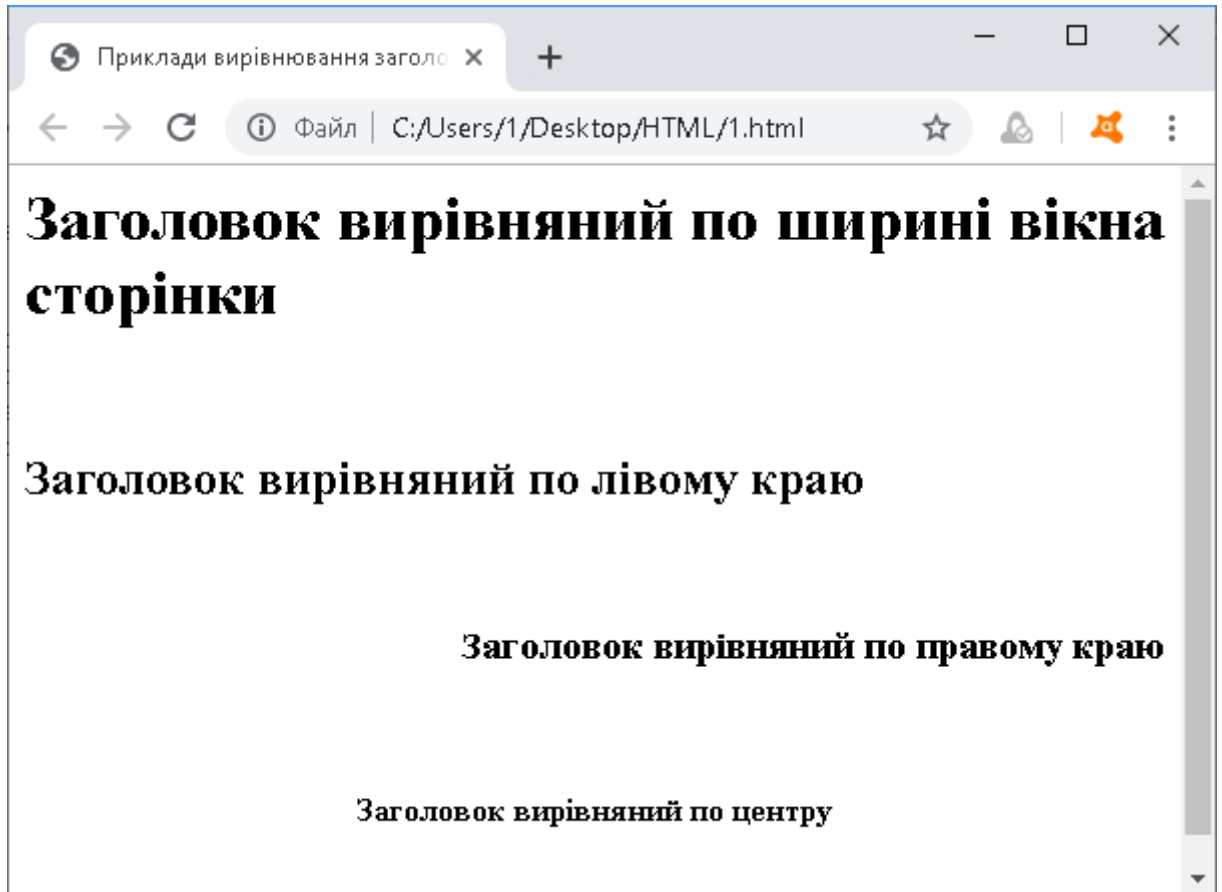


Рисунок 4.3 – Модифікований приклад 4.2.

4.3 Форматування тексту за допомогою контейнерів фізичних стилів

Контейнери фізичних стилів дозволяють виділяти текст напівжирним шрифтом, курсивом, підкреслювати текст тощо. Текст, вміщений в контейнер фізичного стилю всіма браузерами, що підтримують цей стиль, відображається однаково. В табл. 4.1 наведені початкові дескриптори контейнерів фізичних стилів.

Таблиця 4.1 – Початкові дескриптори контейнерів фізичних стилів

Початковий дескриптор контейнера	Стиль
	Напівжирний шрифт
<I>	Курсив
<TT>	Моноширинний шрифт (як у пишучої машинки)
<U>	Підкреслення
<SUB>	Підрядковий шрифт
<SUP>	Надрядковий шрифт
<STRIKE>	Перекреслення

В наступному прикладі наведений лістинг вмісту файлу, що ілюструє використання фізичних стилів (рис. 4.4):

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title> Приклади використання фізичних стилів </title>
</head>
<body>
  Це <b> напівжирний шрифт </b> <br>
  Це <i> курсив </i> <br>
  Це <tt> моноширинний шрифт ( як у пишучої машинки)</tt> <br>
  Це шрифт з <u> підкресленням </u> А це <sub> підрядковий шрифт
  </sub> і <sup> надрядковий шрифт </sup> <br>
  <del> Це приклад шрифту з перекресленням </del>
</body>
</html>

```

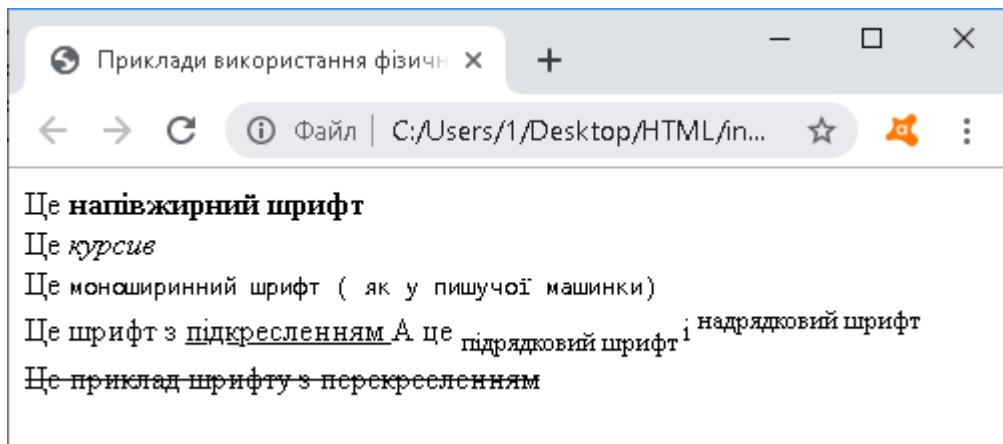


Рисунок 4.4 – Використання фізичних стилів

4.4 Форматування тексту за допомогою контейнерів логічних стилів

Логічні стилі, також, як і фізичні, призначені для різних способів виділення фрагментів тексту. Однак на відміну від чітко визначеного характеру розмітки тексту фізичними стилями, подібна розмітка логічними стилями носить відносний характер, тобто в різних браузерах характер відображення фрагментів тексту, розмічених з використанням логічних стилів, буде різнитися.

В табл. 4.2 приведені початкові дескриптори контейнерів логічних стилів і особливості відображення фрагментів тексту, виділених цими стилями.

Таблиця 4.2 – Початкові дескриптори контейнерів логічних стилів

Початковий дескриптор контейнера	Internet Explorer	Netscape
	Курсив	Курсив
	Напівжирний курсив	Напівжирний
<DFN>	Курсив	Звичайний текст
<CODE>	Малий однорозмірний	Однорозмірний
<SAMP>	Малий однорозмірний	Однорозмірний
<KBD>	Малий однорозмірний	Однорозмірний
<VAR>	Малий однорозмірний	Курсив
<CITE>	Курсив	Курсив
<BIG>	Більший, ніж звичайний	Більший, ніж звичайний
<SMALL>	Трохи менший, ніж звичайний	Менший, ніж звичайний

В наступному прикладі наведений лістинг вмісту файлу, що ілюструє використання логічних стилів (рис. 4.5):

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title> Приклади використання логічних стилів </title>
</head>
<body>
  <em> стиль, заданий тегом EM </em> <br>
  <strong> стиль, заданий тегом STRONG </strong> <br>
  <dfn> стиль, заданий тегом DFN </dfn> <br>
  <code> стиль, заданий тегом CODE </code> <br>
  <samp> стиль, заданий тегом SAMP</samp> <br>
  <kbd> стиль, заданий тегом KBD </kbd> <br>
  <var> стиль, заданий тегом VAR </var> <br>
  <cite> стиль, заданий тегом CITE </cite> <br>
  <big> стиль, заданий тегом BIG </big> <br>
  <small> стиль, заданий тегом SMALL </small> <br>
</body>
</html>
```

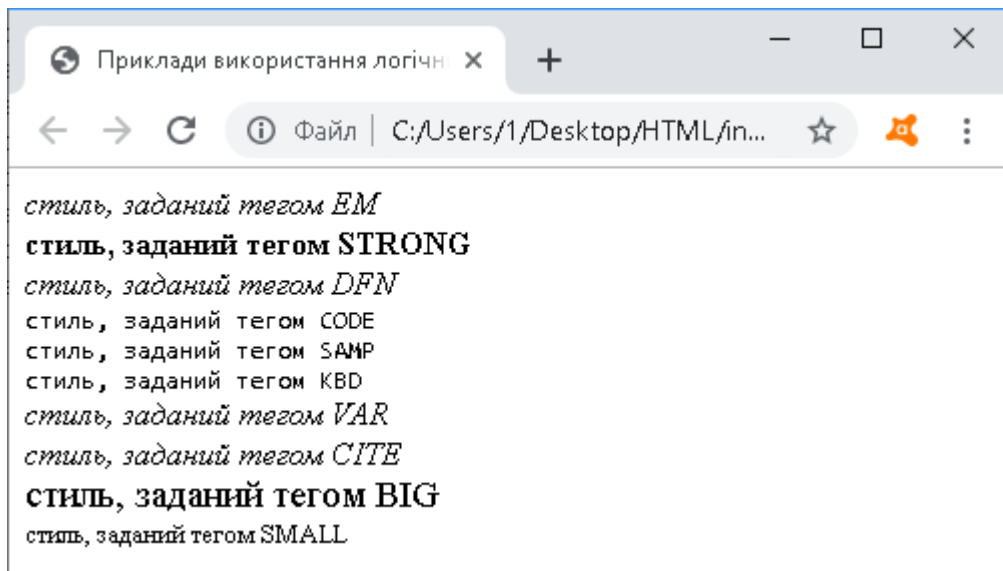


Рисунок 4.5 – Використання логічних стилів

З точки зору розробників стандарту мови HTML, використання логічних стилів має більше переваг, ніж використання фізичних стилів.

4.5 Зміна розміру шрифту. Дескриптори FONT і BASEFONT

Ще одним атрибутом шрифту, крім розглянутих вище, є його розмір (величина). Розмір шрифту в HTML задається цілими числами в діапазоні від 1 до 7. Якщо ніякі атрибути шрифту не вказані явно жодним дескриптором, то для всього тексту встановлюються ті значення атрибутів, котрі прийняті за умовчуванням для даного браузера.

Розмір може бути заданий для всього тексту або для окремого його фрагменту. Окрім визначення абсолютного розміру шрифту можна змінювати його розмір для окремих фрагментів (речень, слів, літер).

Для визначення розміру шрифту всього документу використовується дескриптор

`<BASEFONT SIZE=n>`,

тут *n* означає розмір шрифту, котрий задається цілою цифрою в діапазоні від 1 до 7. Встановлений цим дескриптором основний розмір шрифту застосовується до звичайного і попередньо відформатованого тексту і не застосовується до заголовків, за виключенням тих, котрі модифікуються з використанням елемента FONT з вказуванням відносного розміру шрифту (наприклад, `FONT SIZE=+1`). Слід також відмітити, що в різних браузерах присутнє обмеження на область дії цього дескриптора.

Дескриптор FONT дозволяє задавати чи змінювати заданий абсолютний розмір і колір шрифту. Вказаний дескриптор є контейнером, хоча зона його дії закінчується автоматично, якщо зустрічається наступний дескриптор FONT, котрий задає інший розмір шрифту.

Нижче наведений синтаксис дескриптора FONT:

```
<FONT SIZE=N> текст </FONT>
```

тут N означає розмір шрифту. Задається цілою цифрою в діапазоні від 1 до 7 або цілим числом зі знаком, наприклад, «+1» чи «-2». В цьому випадку розмір шрифту буде змінений на відповідну величину відносно діючого на момент використання дескриптора FONT поточного значення розміру.

Далі наведений лістинг вмісту файлу, що ілюструє використання розглянутих дескрипторів (рис. 4.6):

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title> Приклади завдання розмірів шрифту </title>
</head>
<body>
  <p> Текст має розмір 3, прийнятий за умовчуванням </p>
  <basefont size=5>
  <p>
    Перед початком цього тексту використаний дескриптор
    BASEFONT, який задає розмір 5.
    <font size=1> Текст розміру 1, заданий контейнером FONT.
  </font>
  </p>
  <b> <u>
  <font size=7 color=red>
    Цей текст повинний бути написаним жирним шрифтом з
    підкресленням, шрифтом великого розміру і червоного кольору.
  </font>
  </u> </b>
</body>
</html>
```

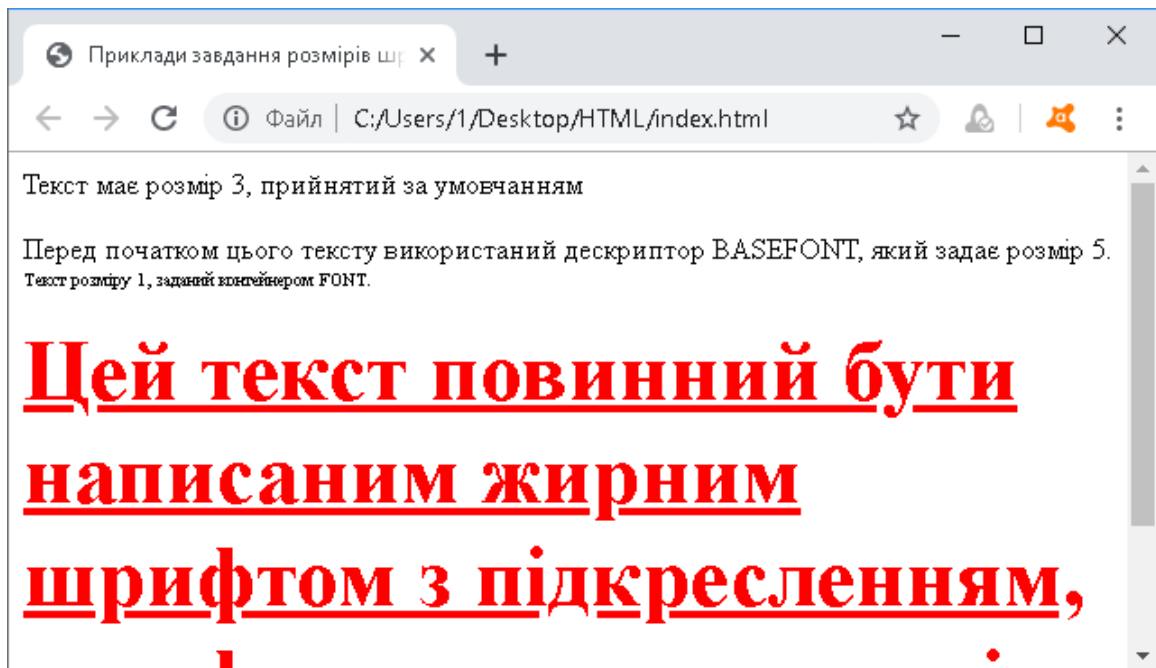


Рисунок 4.6 – Використання дескрипторів FONT і BASEFONT

4.7 Використання попередньо відформатованого тексту. Контейнер <PRE>

Контейнер <PRE> використовується для розміщення попередньо відформатованого тексту, котрий необхідно відобразити на Web-сторінці таким, яким він уже сформований за допомогою будь-якого з допустимих текстових редакторів. Під попередньо відформатованим текстом тут мається на увазі текст з розставленими в ньому пробілами і символами переходу на новий рядок. Нагадаємо, що в звичайному HTML-тексті зайві пробіли і символи переходу на новий рядок, отримані за допомогою інших редакторів, наприклад, блокнотом Notepad, ігноруються. Синтаксис контейнера <PRE>:

<PRE> відформатований текст </PRE>

В наступному прикладі наведений лістинг вмісту файлу, котрий ілюструє використання контейнера <PRE> (рис. 4.8):

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title> Приклад використання контейнера <pre> </title>
</head>
<body>
```

Наведені літери та цифри, котрі можна набрати на клавіатурі:

```
<pre>
0 1 2 3 4 5 6 7 8 9
A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o
```

```
p q r s t u v w x y z
</pre>
```

Повторимо приведений фрагмент тексту, не використовуючи тег PRE:

```
0 1 2 3 4 5 6 7 8 9
A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o
p q r s t u v w x y z
</body>
</html>
```

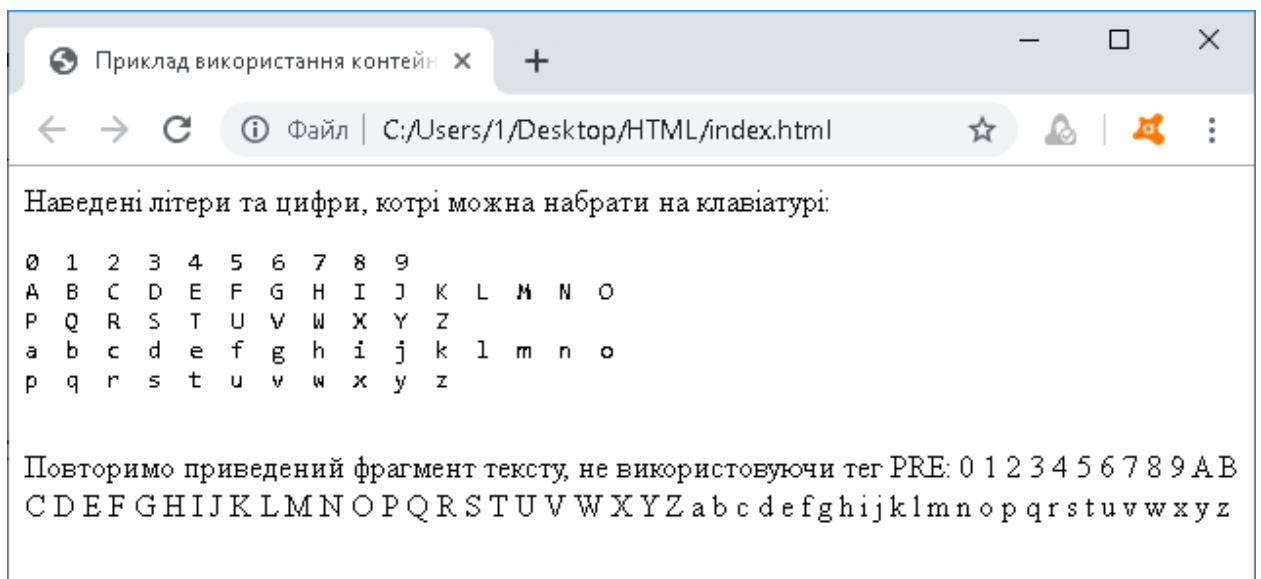


Рисунок 4.8 – Використання контейнера <PRE>

4.8 Створення горизонтальних ліній. Дескриптор <HR>

Горизонтальні лінії, що використовуються в HTML, як правило, виступають візуальними роздільниками змістових розділів Web-сторінки. У текстових фрагментах після горизонтальної лінії браузер автоматично вставляє тег початку абзацу <P>. Дескриптор <HR> може містити ряд атрибутів, які уточнюють положення, розмір і вид лінії, що задається:

```
<HR ALIGN=a, SIZE=m, WIDTH=p, NOSHADE>
```

В табл. 4.3 наведений опис і можливі значення атрибутів тегу <HR>.

Таблиця 4.3 – Опис атрибутів

Атрибут	Можливі значення	Зміст
ALIGN	LEFT, RIGHT, CENTER	Горизонтальне вирівнювання лінії, за умовчужанням - CENTER
NOSHADE	NOSHADE	Задає відображення лінії в одному кольорі (без відтінків)
SIZE	Ціле додатне число	Висота лінії в пікселях
WIDTH	Ціле додатне число, яке означає кількість пікселів або лапки, які містять ціле додатне число зі знаком відсотку, що означає відсоток поточної ширини вікна браузеру.	Довжина лінії
COLOR	Задається по правилам описаних в параграфі 6.1	Колір лінії

У наступному прикладі наведений лістинг вмісту файлу, що ілюструє використання тегу <HR> (рис. 4.9).

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title> Приклад використання тегу <hr> </title>
</head>
<body>
  <p>
    Цей текст підкреслений простою горизонтальною лінією (тег HR)
  </p>
  <hr>
  <p>
```

Зображена нижче горизонтальна лінія вирівняна за правим краєм вікна браузера і має довжину, рівну половині ширини вікна браузера та червоного кольору:

```
</p>
<hr align="right" width="50%" color="red">
<p>
```

Наступна лінія має висоту 5 пікселів, довжину 100 пікселів та голубого кольору:

```
</p>
<hr align="left" size=5 width=100 color="blue">
</body>
</html>
```

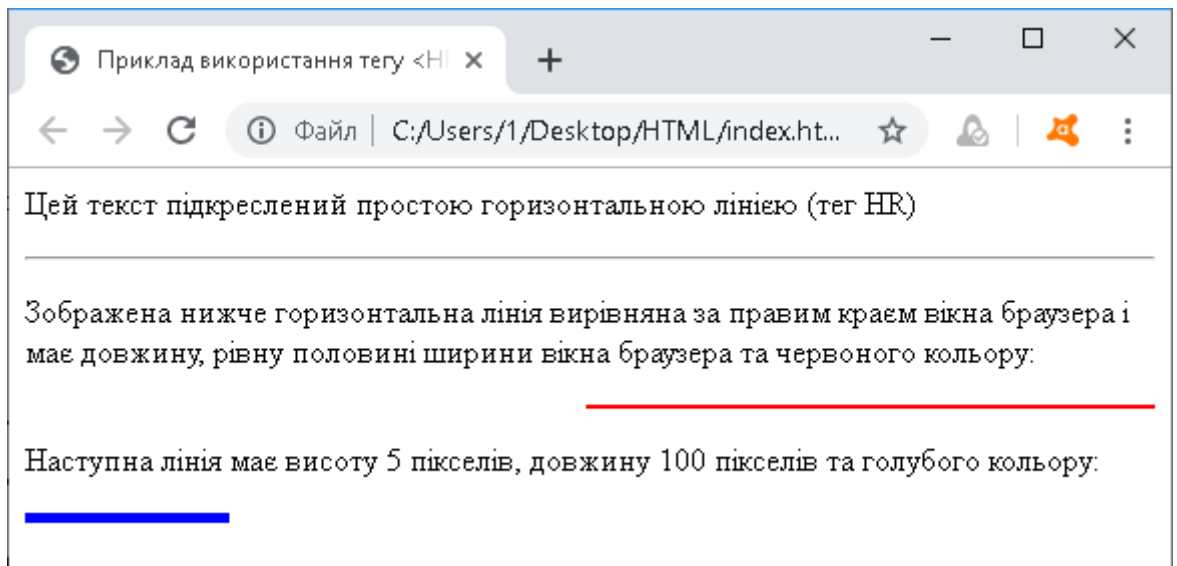


Рисунок 4.9 – Використання тегу <HR>

Питання для самоперевірки:

1. Чи існує різниця між відображенням документу в різних браузерах?
2. Які дескриптори використовуються для завдання рядку і абзацу? Чи є вони контейнерами?
3. Як у файлі HTML записується заголовок?
4. Який атрибут використовується для вирівнювання тексту?
5. Якими є відмінності фізичних і логічних стилів?
6. Що задає дескриптор <VAR>?
7. В чому відмінності дескрипторів і <BASEFONT>?
8. Як задається розмір шрифту?
9. Які параметри форматування тексту залишає дескриптор <PRE>?
10. За допомогою якого атрибута встановлюється товщина ліній?

5. Гіпертекстові посилання (зв'язки). Контейнер <A>

Гіпертекстові посилання (гіперпосилання) встановлюють зв'язки між окремими частинами одного й того ж документу (внутрішні посилання) або різними документами. Будь-яке гіперпосилання можна розглядати як поєднання окремих точок в документі з іншими точками в тому ж самому чи в інших документах. Встановлення такого зв'язку передбачає, що при зверненні до вказаного посилання (клацанні по ньому мишкою), відбувається активація означеного в цьому посиланні об'єкта, тобто у вікні браузера відображається новий фрагмент тексту, запускається аудіо-додаток, завантажується той чи інший файл тощо. Умовно можна сказати, що будь-який такий зв'язок (посилання) з'єднує дві точки: точку розташування початку зв'язку (саме посилання) і деяку іншу точку, названу метою (target) зв'язку. Остання часто є початком іншого документу.

Для організації гіперпосилань в HTML використовується контейнер `<A>` – скорочення від англійського anchors – якір. Нижче наведено опис двох синтаксичних конструкцій цього контейнеру – контейнер, що задає мету, і контейнер, що задає точку переходу:

Контейнер, що задає мету (якір):

```
<A HREF="мета" TITLE="текст"> якір </A>
```

Контейнер, що задає точку переходу:

```
<A NAME="ім'я"></A>
```

Атрибут `TITLE` не є обов'язковим. Текст, що задається в цьому атрибуті, відображається при наведенні на посилання вказівником мишки і зазвичай використовується для додаткового інформування користувачів про призначення посилання. Така інформація може знадобитися, наприклад, якщо замість текстового якоря буде використовуватися зображення.

Всі посилання, що містяться в документі, відображаються у вікні браузера іншим кольором – зазвичай синім, і виділяються підкресленням. При наведенні на посилання вказівника мишки в нижньому куті браузера відображається адреса, яка міститься в атрибуті `HREF`. У вікні браузера буде відображатися тільки вміст контейнера.

Умовно всі посилання можна розділити на внутрішні, відносні (локальні) і зовнішні (віддалені).

5.1 Зовнішні посилання

Зовнішні посилання – це посилання на ресурси інших Web-серверів.

У наступному прикладі наведений лістинг вмісту файлу, що ілюструє найпростіший випадок завдання зовнішнього посилання (рис. 5.1)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title> Приклад використання тегу <a> (завдання зовнішнього
посилання)</title>
</head>
<body>
  Тут можна подивитися різноманітну інформацію про
  <a href="http://www.donnu.edu.ua"> ДонНУ імені Василя Стуса
</a>
</body>
</html>
```

В результаті натискання на посилання в прикладі отримуємо перехід до сайту університету.

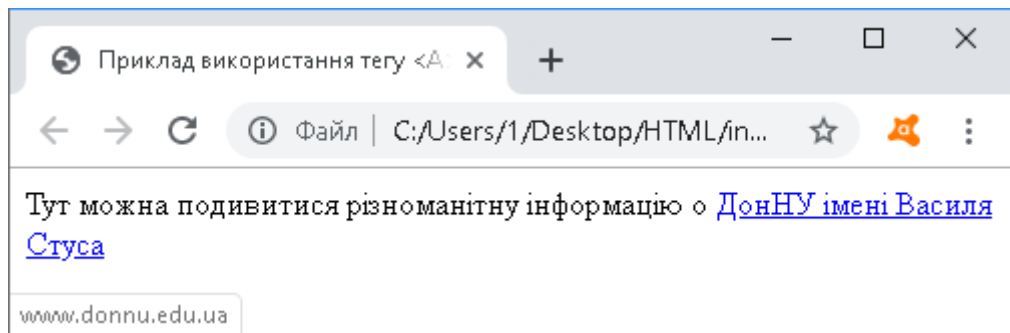


Рисунок 5.1 – Завдання зовнішнього посилання

5.2 Внутрішні посилання

Внутрішні – це посилання на об’єкти в межах одного документа. При використанні таких посилань необхідно задавати точку переходу, тобто використовувати контейнер `<A>` з атрибутом `NAME`. Внутрішні імена задаються за визначеними правилами. По-перше, вони повинні бути унікальними в межах одного Web-документу. По-друге, перед посиланням на внутрішнє ім’я ставиться символ `#`. Наприклад, якщо кінцева точка задається контейнером `` текст ``, то посилання на цю точку буде задаватися контейнером `` якір ``.

В наступному прикладі наведений лістинг вмісту файлу, котрий ілюструє випадок завдання внутрішнього посилання (рис. 5.2):

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title> Приклад завдання внутрішніх посилань</title>
</head>
<body>
  Зміст <br>
  <a href="#розділ 1"> Перший розділ </a> <br>
  <a href="#розділ 2"> Другий розділ </a> <br>
  <a href="#розділ 3"> Третій розділ </a> <br>
  <br>
  <br><br><br><br>
  <a name="розділ 1"> Текст першого розділу </a> <br>
  (Текст розділу)
  <br><br><br><br>
  <a name="розділ 2"> Текст другого розділу </A> <br>
  (Текст розділу)
  <br><br><br><br>
  <a name="розділ 3"> Текст третього розділу </a> <br>
  (Текст розділу)
</body>
```

</html>

В результаті натискання на посилання в прикладі матимемо наступну картину (рис. 5.2)

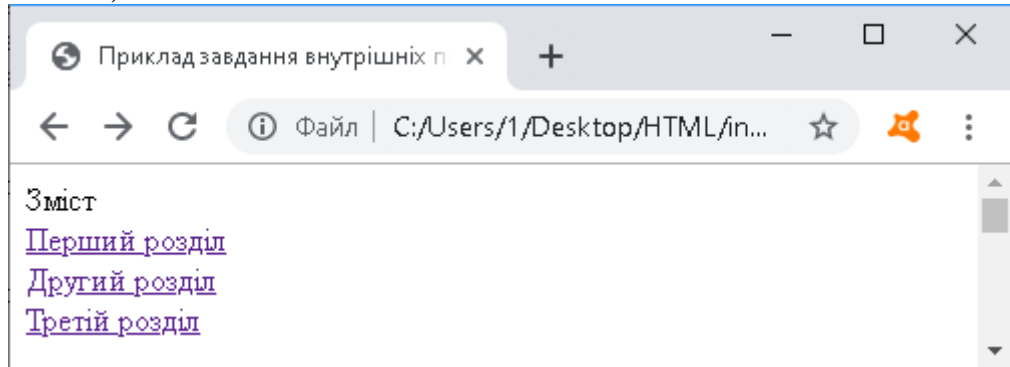


Рисунок 5.2 – Завдання внутрішнього посилання

Численні теги
, присутні в цьому прикладі, вставлені для надання прикладу більшої ілюстративності, їх кількість може бути зменшено або збільшено.

5.3 Відносні посилання

Відносні – це посилання на інші ресурси (Web-сторінки чи інші файли) в межах одного й того ж Web-сервера.

Розглянемо випадок, коли маються окремі файли: page 2.html, page 3.html, page 4.html. Для завдання посилань на ці файли створюється ще один файл, лістинг якого приведений в наступному прикладі. Передбачається, що всі ці файли розташовані в тій же папці, що і файл, текст якого наведений в прикладі (рис. 5.3).

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title> Приклад завдання локальних посилань</title>
</head>
<body>
  Зміст <br>
  <a href="page 2.html" target="_blank"> Сторінка 2 </a> <br>
  <a href="page 3.html" target="_blank"> Сторінка 3 </a> <br>
  <a href="page 4.html" target="_blank"> Сторінка 4 </a> <br>
  <br>
</body>
</html>
```

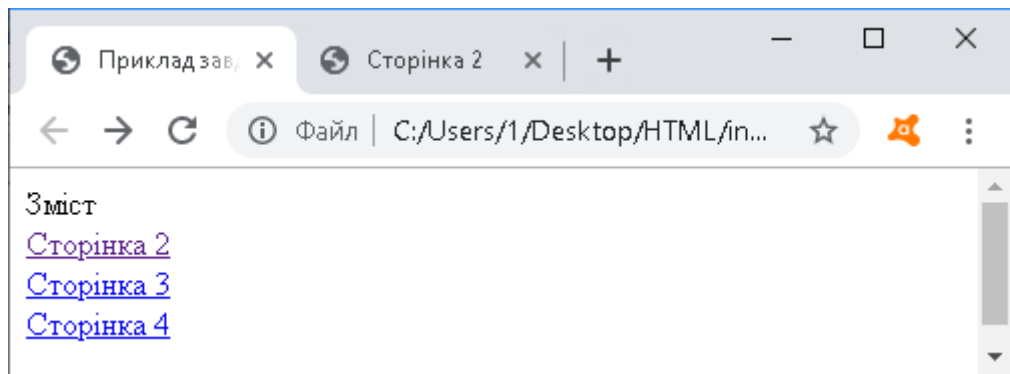


Рисунок 5.3 – Приклад завдання локальних посилань

При натисканні, наприклад, на Сторінка 2, відкривається в новому вікні сторінка з інформацією з -файлу page 2.html, який зберігається в тій же папці, що й основний файл (рис. 5.4).

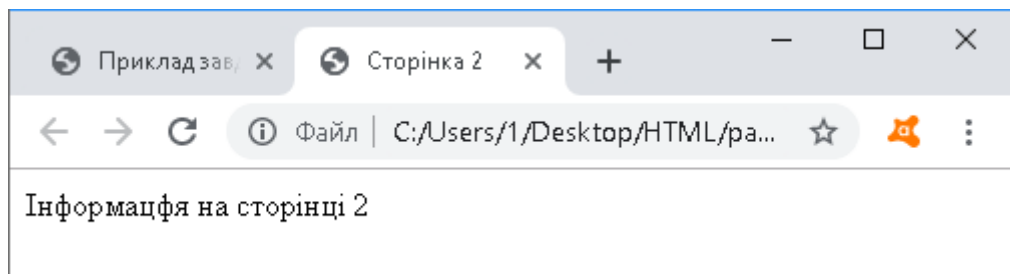


Рисунок 5.4 – Результат локального посилання

Питання для самоперевірки:

1. Яким є призначення гіперпосилань? Який контейнер використовується для їх створення?
2. Що ще можна використовувати замість тексту в гіперпосиланнях?
3. Що таке якір?
4. Які типи посилань використовуються? Якими є їхнє призначення і відмінності?

6. Використання графіки та кольору

Використання графіки в HTML дозволяє користувачеві задавати чи змінювати колір текстового вмісту, текстових посилань, вставляти в текст зображення, задавати і змінювати фоновий колір Web-сторінки, вставляти фонові зображення, задавати якорі посилань у вигляді зображень, формувати карти посилань тощо. В даних методичних рекомендаціях розглядаються можливості HTML, пов'язані з завданням кольору тексту, фону, гіпертекстових посилань, вставки в текст зображень, вставки якорів-зображень.

При роботі з графікою слід враховувати наступні моменти:

- не всі браузері є графічними, тобто можуть відображати графічні файли і кольоровий текст;

- у кожного графічного браузерa існують обмеження в кількості відображених кольорів (звичайно 256 кольорів);
- будь-який новий колір, що не входить до палітри кольорів браузерa, визначається їм у вигляді сполучення близько розташованих пікселів різноманітних кольорів базової палітри;
- в різних операційних системах використовуються різні палітри;
- палітри деяких операційних систем мають частину загальних кольорів;
- в різних графічних браузерaх реалізовані різні алгоритми побудови додаткових кольорів;
- розмір пікселя залежить від можливостей і налаштувань комп'ютеру користувача;
- як правило, будь-який графічний файл займає більше місця, ніж текстовий.

6.1 Способи завдання кольору

Будь-який колір на екрані комп'ютера отримується завдяки суміші трьох кольорів: червоного (red - R), зеленого (green - G) і синього (blue - B). Значення кожного з кольорів задається двозначним шістнадцятиричним кодом, тобто може приймати значення від 00 до FF.

Колір в HTML задається двома способами:

- 1) RGB значенням;
- 2) кольоровим іменем.

В табл. 6.1 наведений 16-кольоровий стандарт, який підтримується VGA палітрою Windows.

Таблиця 6.1 – Палітра кольорів

RGB значення	Українська назва кольору	Англійська назва кольору
1	2	3
"#000000"	чорний	Black
"#000080"	темно-синій	Navy
"#0000FF"	синій	Blue
"#008000"	зелений	Green
"#008080"	чирок	Teal
"#00FF00"	вапно	Lime
"#00FFFF"	аква	Aqua
"#800000"	темно-бордовий	Maroon
"#800080"	фіолетовий	Purple
"#808000"	оливковий	Olive
"#808080"	сірий	Gray
"#C0C0C0"	срібло	Silver

Продовження таблиці 6.1

1	2	3
“#FF0000”	червоний	Red
“#FF00FF”	фуксія	Fuchsia
“#FFFF00”	жовтий	Yellow
“#FFFFFF”	білий	White

Слід звернути увагу на правило запису RGB значення. Нагадуємо, що окрім вказаних кольорів, які складають загальноприйнятий стандарт, кожний графічний браузер підтримує свою, як правило 256-кольорову, палітру. Подробиці про формуванні RGB значення можна подивитися на сайті <http://www.w3.org/pub/WWW/Printing/motta/W3Color.html>.

6.2 Зміна кольорів тексту і фону. Атрибути TEXT, COLOR, BGCOLOR, LINK, ALINK, VLINK, BACKGROUND

До атрибутів, які задають кольорові параметри всієї сторінки, відносяться TEXT, BGCOLOR, LINK, ALINK, VLINK, BACKGROUND. Ці атрибути можуть задаватися в початковому тезі контейнеру <BODY>.

При завданні тих чи інших атрибутів слід враховувати, що колір тексту може бути зміненим, наприклад, для окремого абзацу, а колір фону чи фонове зображення локально не може бути зміненим.

Атрибут BGCOLOR визначає колір фону всієї Web-сторінки. Синтаксис цього атрибута:

BGCOLOR=значення_кольору

Атрибут TEXT задає колір тексту всього документу. Як правило, за умовчуванням в браузері текст всього документа має чорний колір. Синтаксис цього атрибута:

TEXT= значення_кольору

Атрибут BACKGROUND задає URL фонового зображення. Синтаксис цього атрибута:

BACKGROUND="URL"

Атрибути LINK, ALINK, VLINK задають колір невідвідуваного, активного і відвідуваного посилань відповідно. За умовчуванням, наприклад, в браузері Internet Explorer невідвідуване посилання має синій колір. Синтаксис цих атрибутів:

LINK=значення_кольору
ALINK=значення_кольору
VLINK=значення_кольору

Атрибут COLOR використовується для завдання кольору локального фрагмента тексту. Цей атрибут може бути вставлений в початковий тег контейнерів або <P>. Синтаксис цього атрибута:

COLOR= значення_кольору

6.3 Вставка зображень. Дескриптор

Дескриптор використовується для вставки в текст Web-сторінки зображень. Наразі найбільш поширеними графічними форматами, які використовуються у Web-сторінках, є jpeg і gif формати. Синтаксис цього атрибута:

<IMG SRC="URL" ALT="текст" ALIGN=позиціювання
HEIGHT=n WIDTH=m BORDER=p HSPACE=k VSPACE=h>

З усіх наведених атрибутів обов'язковим є тільки атрибут SRC. В табл. 6.2 наведено опис атрибутів дескриптора

Таблиця 6.2 – Опис атрибутів дескриптора

Атрибут	Значення	Тлумачення
SRC	URL	Адреса зображення
ALT	Рядок	Текстовий опис зображення
ALIGN	TOP, MIDDLE, BOTTOM, LEFT, RIGHT	Спосіб розміщення зображення відносно поточного текстового рядка
HEIGHT	Додатне ціле число	Висота зображення в пікселях
WIDTH	Додатне ціле число	Ширина зображення в пікселях
BORDER	Додатне ціле число	Ширина бордюру (рамки) навколо зображення в пікселях
HSPACE	Додатне ціле число	Ширина незаповненого простору безпосередньо ліворуч і праворуч від зображення в пікселях
VSPACE	Додатне ціле число	Висота незаповненого простору вище і нижче зображення в пікселях

Значення ALIGN мають наступне тлумачення:

ALIGN=TOP – верх зображення встановлюється на одному рівні з верхом поточного текстового рядка. Різні браузері інтерпретують це значення по-різному;

ALIGN=MIDDLE – середина зображення вирівнюється за основним текстовим рядком;

ALIGN=BOTTOM – низ зображення вирівнюється за основним рядком (приймається за умовчанням);

ALIGN=LEFT – зображення переміщується до поточного лівого краю. Текст, що слідує за дескриптором , який містить атрибут ALIGN зі вказаним значенням, розташовується вздовж правої сторони зображення;

`ALIGN=RIGHT` – зображення переміщується до поточного правого краю. Текст, що слідує за дескриптором ``, який містить атрибут `ALIGN` зі вказаним значенням, розташовується вздовж лівої сторони зображення.

Атрибути `WIDTH` і `HEIGHT` задають розміри області сторінки, що відводиться для вставки графічного файлу. Розміри області задаються в пікселях. Сумісне використання цих атрибутів дозволяє браузерам зарезервувати місце на екрані для зображення ще до того, як вони будуть переданими мережею. Це дозволяє надавати Web-сторінці реального вигляду до закінчення передачі всіх графічних файлів. Атрибути не змінюють розміри зображення, а тільки резервують місце для їх відображення браузерами. Це означає, що ці атрибути повинні задавати реальні значення розмірів зображення.

Атрибут `BORDER` задає розмір товщини (в пікселях) рамки навколо зображення.

Атрибути `HSPACE` і `VSPACE` задають розміри (в пікселях) незаповненого простору навколо зображення.

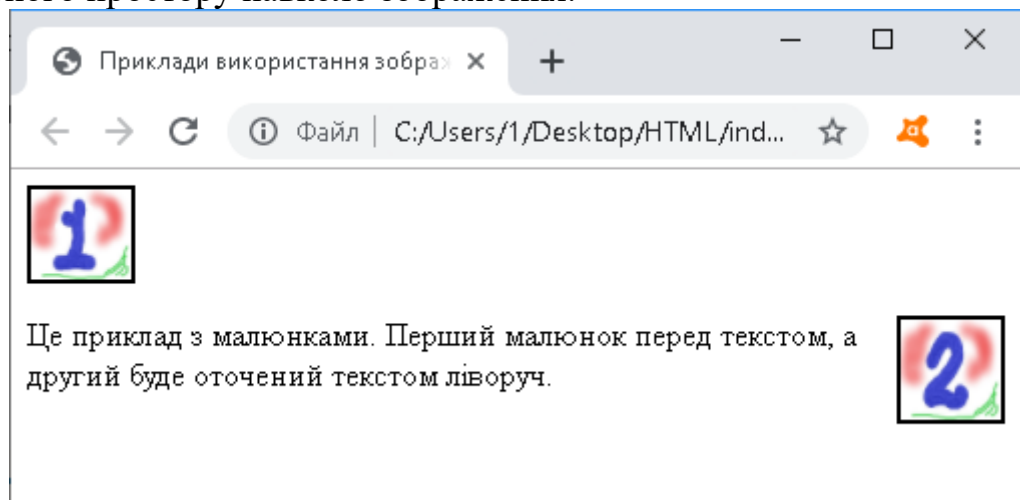


Рисунок 6.2 – Приклади використання зображень

Дескриптор `` можна використовувати для завдання посилань-зображень. Для цього замість тексту якоря вставляється тег ``, тобто контейнер посилання буде мати вигляд:

```
<A HREF="URL"> <IMG SRC="URL"> </A>
```

У наступному прикладі наведений лістинг файлу, що ілюструє різні випадки використання зображень (рис. 6.2).

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title> Приклади використання зображень </title>
</head>
<body>
```

```

<p>

```

Це приклад з малюнками. Перший малюнок перед текстом, а другий буде оточений текстом ліворуч.

```
</p>
</body>
</html>
```

Питання для самоперевірки:

1. Скільки зазвичай кольорів підтримують браузері?
2. Якими способами задається колір в HTML?
3. Яким є призначення атрибуту BACKGROUND?
4. Який дескриптор використовується для вставки зображень?
5. Які атрибути задають розмір області для зображення, що завантажуються?
6. Як задаються посилання-зображення?

7. Списки. Контейнери , , <MENU>, <DIR>, <DL>, <DT>, <DD>

За типом розмітки списки поділяються на нумеровані (невпорядковані, марковані), нумеровані (впорядковані) і списки визначень.

Невпорядковані списки задаються контейнерами , <MENU> і <DIR>. Формально контейнер <MENU> задає список елементів меню, а контейнер <DIR> - список елементів каталогів. Однак списки, що задаються цими трьома контейнерами, виглядають однаково.

Синтаксис контейнера :

```
<UL TYPE маркер>
<LI> елемент списку 1
<LI >елемент списку 2
...
</UL>
```

Значення маркер в атрибуті TYPE може бути таким:

DISC – маркування заповненим кружечком (за умовчужанням);

SQUARE – маркування заповненим квадратом;

CIRCLE – маркування незаповненим кружечком.

Вказана відповідність значень атрибуту TYPE і маркерів наведена для браузера Internet Explorer. Цей атрибут не є обов'язковим.

Синтаксис контейнеру <DIR>:

```
<DIR>
<LI> елемент списку 1
<LI> елемент списку 2
...
</DIR>
```

Синтаксис контейнеру <MENU>:

```
<MENU>
<LI> елемент списку 1
<LI> елемент списку 2
...
</MENU>
```

Списки визначень задаються контейнером <DL>, який, в свою чергу, містить контейнери <DT> і <DD>, використані у визначеному порядку:

```
<DL>
<DT> термін 1 <DD> визначення терміну 1
<DT> термін 2 <DD> визначення терміну 2
...
</DL>
```

Закриваючі теги контейнерів <DT> і <DD> необов'язкові.

Нумеровані списки задаються контейнером :

```
<OL TYPE=тип_нумерації START=ціле_число>
<LI> елемент списку 1
<LI> елемент списку 2
...
</OL>
```

Тут атрибут TYPE, що задає тип нумерації, може приймати наступні значення:

- 1 – звичайні (арабські) числа (за умовчанням);
- a – малі латинські букви (нижній регістр);
- A – великі латинські букви (верхній регістр);
- i – малі римські цифри (нижній регістр);
- I – великі римські цифри (верхній регістр).

Вважається, що як усі цифри, так і всі букви латинського алфавіту впорядковані за зростанням, причому після однобуквених символів слідує двобуквені, трибуквені тощо.

Атрибут START, який задає початкове значення параметру нумерації, може задаватися будь-яким цілим додатним числом. Якщо список нумерується числами, нумерація буде розпочата з відповідної цифри. Якщо ж нумерація задана буквами, слід враховувати кількість букв латинського алфавіту, бо після однобуквеної нумерації починається двобуквена тощо.

Завдання першого тегу викликає появу першого номера, а при завданні кожного наступного тегу автоматично береться наступний за порядком номер, тобто крок нумерації завжди незмінний і дорівнює одиниці. Змінювати крок можна тільки задаючи чергове значення номера власноруч в атрибуті VALUE контейнера .

Контейнер завдання вмісту елемента списку , в залежності від типу списку, має наступний синтаксис:

- у випадку нумерованого списку
 <LI TYPE=маркер> зміст елемента списку
- у випадку нумерованого списку
 <LI TYPE=тип_нумерації VALUE=номер_елемента_списку>
 зміст елемента списку

Нагадаємо, що кінцевий тег завжди може бути відсутнім.

Наведемо лістинг файлу, який містить різні типи списків (рис. 7.1):

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title> Приклади списків </title>
</head>
<body>
  <h3 align=center> В цьому списку наведений приклад різних маркерів
для нумерованих списків</h3>
  <ul type=disc>
    <li> – заповнений кружечок
    <li type=square> – заповнений квадрат
    <li type=circle> – незаповнений кружечок
  </ul>
  <h3 align=center>Приклади вкладених списків </h3>
  <p align=left>
    <ol type=i>
      <li> Перший рядок першого рівня <br>
      <ul>
        <li> Перший рядок другого рівня<br>
        <li> Другий рядок другого рівня <br>
      </ul>
      <li> Другий рядок першого рівня <br>
      <ul type = disc>
        <li> Перший рядок другого рівня<br>
        <li> Другий рядок другого рівня <br>
      </ul>
    </ol>
  </p>
</body>
</html>
```

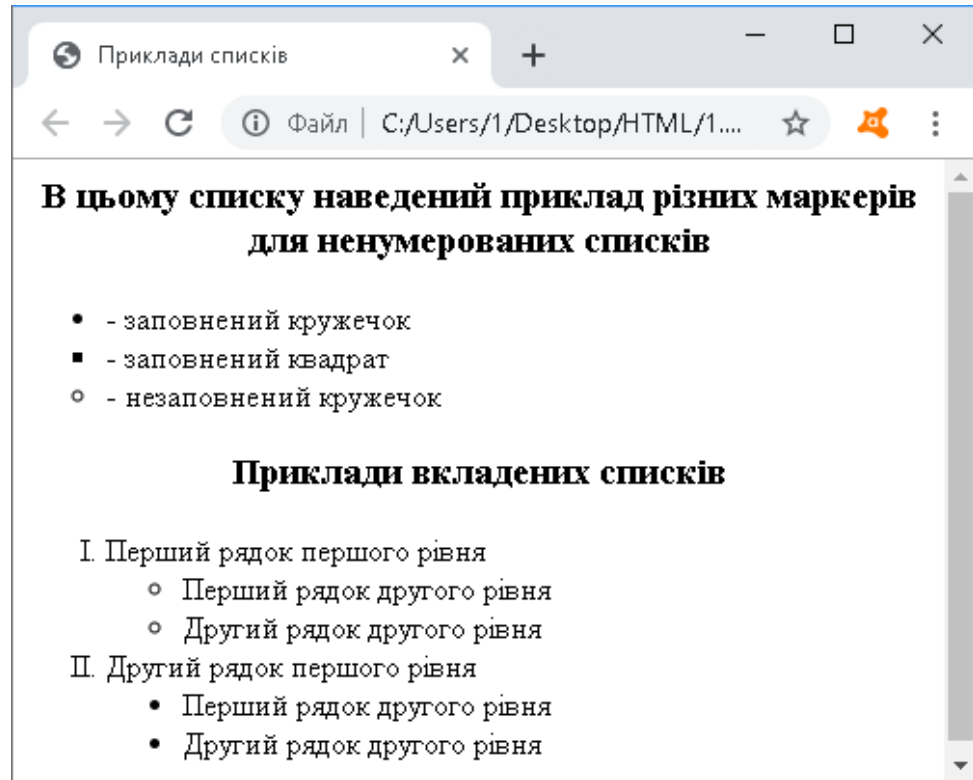


Рисунок 7.1 – Різні типи списків

Питання для самоперевірки.

1. Що таке списки? Якими є їхні типи?
2. В чому відмінність контейнерів ``, `<MENU>`, `<DIR>`?
3. Який контейнер задає нумеровані списки? Яким є його синтаксис?
4. Що може використовуватися в якості нумерації в нумерованому списку?
5. Чи можна починати список не з «одиниці», і як задати довільний крок списку?
6. Чи є обов'язковим тег ``?

8. Таблиці

В HTML таблиця у звичайному варіанті представляється в природному вигляді, тобто рядки і стовпці таблиці розташовані так, що відповідні їм комірки розташовані узгоджено. До елементів таблиці, що задаються тегамі HTML, відносяться власне таблиця (TABLE element), як окремий елемент, рядки і окремі комірки.

При використанні в Web-сторінці такого елемента, як таблиця, слід враховувати наступне:

- таблиці, розмір яких перевищує фізичні розміри екрану, розглядати незручно;
- ширина таблиці розраховується як сума ширин всіх її комірок;

- навколо таблиці і в кожній її комірці можна задавати рамки (borders);
- комірки таблиці можуть містити як текст, так і зображення (вміст графічних файлів);
- комірки можна об'єднувати за визначеними правилами як вздовж рядку, так і вздовж стовпця;
- кількість комірок в усіх рядках має бути однаковою;
- кількість комірок в усіх стовпцях має бути однаковою;
- вміст комірок можна вирівнювати як за горизонталлю, так і за вертикаллю;
- таблиці можуть бути вкладеними;
- всі теги, що описують елементи таблиці, є контейнерами. Якщо закриваючий тег елемента таблиці відсутній, деякі браузері сприймають таку конструкцію як помилку;
- для форматування вмісту таблиці в цілому, окремого рядка чи окремої комірки можуть використовуватися як логічні, так і фізичні стилі. Колір задається за тими ж правилами, що й для решти частини Web-сторінки.

8.1 Структура таблиці

Вся таблиця задається контейнером `<TABLE>`. Наступним за вкладеністю обов'язковим елементом таблиці є контейнер `<TR>`, що задає черговий рядок. Рядки містять комірки, які задаються контейнерами `<TD>` або `<TH>`. Крім цих обов'язкових елементів таблиця може містити ще необов'язковий елемент – контейнер `<CAPTION>`, який задає заголовок таблиці. В загальному вигляді без згадування можливих атрибутів цих контейнерів, структура таблиці виглядає наступним чином:

```

<TABLE>
  <TR>
    <TH>...</TH>
    <TD>...</TD>
  </TR>
  <TR>
    <TH>...</TH>
    <TD>...</TD>
  </TR>
  ...
</TABLE>

```

Кожний контейнер `<TR>...</TR>` задає окремий рядок. Кожний контейнер `<TH>...</TH>` чи `<TD>...</TD>` задає окрему комірку. Для зручності розробників Web-сторінок в HTML присутні комірки двох типів: комірки, що містять заголовки (контейнер `<TH>`), і комірки, що містять дані (контейнер `<TD>`).

Вміст комірок, визначених контейнером `<TD>`, зазвичай (за умовчужанням) представлений браузером нормальним шрифтом. Іноді

браузери можуть використовувати шрифт меншого розміру. За умовчуванням вміст таких комірок вирівнюється за лівим краєм.

Вміст комірок, визначених контейнером <TH>, зазвичай (за умовчуванням) представлений браузером більш виділеним шрифтом, наприклад, напівжирним. За умовчуванням вміст таких комірок вирівнюється за центром.

Розглянемо тепер більш детально контейнери завдання рядків і комірок.

8.2 Контейнер завдання таблиці <TABLE>

Контейнер завдання таблиці може мати наступні атрибути:

<TABLE

ALIGN=горизонтальне_вирівнювання

WIDTH=ширина_всієї_таблиці

BORDER=ширина_рамки_таблиці

CELLSPACING=проміжок_між_комірками

CELLPADDING=проміжок_між_вмістом_комірки_і_рамкою>

</TABLE>

Горизонтальне вирівнювання задається значеннями:

LEFT – вирівнювання за лівим краєм (за умовчуванням);

CENTER – вирівнювання за центром;

RIGHT – вирівнювання за правим краєм.

Ширина таблиці задається в пікселях чи у відсотках від ширини вікна Web-браузера. За умовчуванням ширина кожного стовпця обирається так, щоб в комірці вмістився самий довгий елемент або сама велика картинка.

Ширина рамки навколо таблиці задається в пікселях. За умовчуванням цей параметр дорівнює 0. Це значить, що рамка відсутня.

Проміжок між комірками задається в пікселях. Комірки можуть знаходитися на деякій відстані одна від одної. Ця відстань однакова за горизонталлю і за вертикаллю. Якщо параметр BORDER не дорівнює 0, то параметр CELLSPACING, як правило, встановлюється рівним 1.

Проміжок між вмістом комірки і рамкою навколо цієї комірки також задається в пікселях.

Наведені в цьому контейнері атрибути стосуються параметрів форматування всієї таблиці.

8.3 Контейнер завдання рядку <TR>, <TD>, <TH>

Як уже згадувалося раніше, рядок задається контейнером <TR>. Загальний вигляд цього контейнера такий:

<TR

ALIGN=горизонтальне_вирівнювання

VALIGN =вертикальне_вирівнювання>
 Комірки-заголовки або комірки даних
 </TR>

Горизонтальне вирівнювання задається значеннями:
 LEFT – вирівнювання за лівим краєм (за умовчужанням)
 CENTER – вирівнювання за центром
 RIGHT – вирівнювання за правим краєм.

Вертикальне вирівнювання задається значеннями:
 TOP – за верхнім краєм комірки
 MIDDLE – за центром комірки (за умовчужанням)
 BOTTOM – за нижнім краєм комірки.

8.4 Контейнери завдання комірок <TD>, <TH>

Атрибути контейнеру завдання комірок доповнюють або змінюють значення відповідних атрибутів контейнерів завдання конкретного рядка або всієї таблиці. Контейнер <TD> завдання комірок даних і контейнер <TH> завдання комірок-заголовків мають однакові атрибути:

<TD чи TH
 NOWARP
 ROWSPAN=n
 COLSPAN=m
 ALIGN=горизонтальне_вирівнювання
 VALIGN=вертикальне_вирівнювання>
 WIDTH=ширина_комірки
 HEIGHT=висота_комірки
 </TD чи /TH>

Для відміни переносу слів в комірці використовується атрибут NOWARP. При відсутності цього атрибута браузер зможе переносити слова на нові рядки, використовуючи для визначення місця вставки переносу пробіл між словами.

Значення параметрів горизонтального і вертикального вирівнювання аналогічні значенням відповідних параметрів контейнерів <TABLE> і <TR>.

Ширина і висота комірки задаються в пікселях. Ці значення не повинні конфліктувати з відповідними значеннями сусідніх комірок, інакше вони будуть проігноровані.

Об'єднання n комірок вздовж стовпця задається атрибутом ROWSPAN, а об'єднання m комірок вздовж рядка задається атрибутом COLSPAN.

В наступному прикладі наведений лістинг файлу, якій містить описані вище контейнери:

```
<!DOCTYPE html>
<html>
```

```

<head>
  <meta charset="utf-8">
  <title> Приклад таблиці </title>
</head>
<body>
  <table border="1">
    <tr> <th width="100%"> Заголовок строки та стовбця 1
    </th>
    <th width="100%"> Заголовок стовбця 2 </th>
    <th width="100%">Заголовок стовбця 3 </th>
    <th width="100%"> Заголовок стовбця 4 </th> </tr>
    <tr> <td width="100%"> Назва строки 2 </td>
    <td width="100%"> дані строки 2 стовбця 2 </td>
    <td width="100%"> дані строки 2 стовбця 3 </td>
    <td width="100%"> дані строки 2 стовбця 4 </td> </tr>
    <tr> <td width="100%"> Назва строки 3 </td>
    <td width="100%"> дані строки 3 стовбця 2 </td>
    <td width="100%"> дані строки 3 стовбця 3 </td>
    <td width="100%"> дані строки 3 стовбця 4 </td> </tr>
    <tr> <td width="100%"> Назва строки 4 </td>
    <td width="100%"> дані строки 4 стовбця 2 </TD>
    <td width="100%"> дані строки 4 стовбця 3 </td>
    <td width="100%"> дані строки 4 стовбця 4 </td> </tr>
  </table>
</body>
</HTML>

```

Заголовок строки та стовбця 1	Заголовок стовбця 2	Заголовок стовбця 3	Заголовок стовбця 4
Назва строки 2	дані строки 2 стовбця 2	дані строки 2 стовбця 3	дані строки 2 стовбця 4
Назва строки 3	дані строки 3 стовбця 2	дані строки 3 стовбця 3	дані строки 3 стовбця 4
Назва строки 4	дані строки 4 стовбця 2	дані строки 4 стовбця 3	дані строки 4 стовбця 4

Рисунок 8.3 – Приклад таблиці

Питання для самоперевірки:

1. Які контейнери використовуються для завдання таблиці?

2. Як задати рядок, комірку даних, комірку заголовку?
3. Яке форматування задається за умовчужанням для тексту в комірках даних і заголовку?
4. Якою є товщина рамки за умовчужанням?
5. Як відбувається перенос слів у таблицях?
6. Чи можливо в таблиці задавати об'єднання комірок за горизонталлю, використовуючи параметр ROWSPAN?

9. Використання відеороликів на веб-сторінці

Дескриптор `<video>` додає, відтворює і управляє настройками відеоролика на веб-сторінці. Шлях до файлу задається через атрибут `src` або вкладений дескриптор `<source>`. Список підтримуваних браузерами аудіо та відеокодеків обмежений.

Для універсального відтворення в різних браузерах відео кодують за допомогою різних кодеків і додають файли одночасно.

З усіх атрибутів обов'язковим є тільки атрибут `src`. В табл. 9.1 наведено опис атрибутів дескриптора `<VIDEO >`

Таблиця 9.1 – Опис атрибутів дескриптора `< video >`

Атрибут	Значення	Тлумачення
<code>SRC</code>	URL	Адреса відеоролика
<code>AUTOPLAY</code>		Відео починає відтворюватися автоматично після завантаження сторінки
<code>CONTROLS</code>		Додає панель управління до відеоролика.
<code>HEIGHT</code>	Додатне ціле число	Задає висоту області для відтворення відеоролика
<code>WIDTH</code>	Додатне ціле число	Задає ширину області для відтворення відеоролика
<code>LOOP</code>		Повторює відтворення відео від початку до його завершення
<code>POSTER</code>	URL	Вказує адресу картинки, яка буде відображатися, поки відео недоступне або не чути
<code>PRELOAD</code>	<code>none, metadata, auto</code>	Використовується для завантаження відео разом із завантаженням веб-сторінки

В наступному прикладі наведений лістинг файлу, який ілюструє використання відіоролика:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Приклад використання video</title>
```



```

</head>
<body>
  <video width="400" height="300" controls="controls" >
    <source src="1.ogv" type='video/ogg '>
  </video>
</body>
</html>

```

Питання для самоперевірки.

1. Який дескриптор управляє настройками відеоролика на веб-сторінці?
2. Який атрибут є обов'язковим для дескриптора <VIDEO>?
3. Яким є тлумачення атрибуту POSTER?
4. Яким є тлумачення атрибуту CONTROLS?
5. Яким є тлумачення атрибуту LOOP?

10. Використання аудіозаписів на веб-сторінці

Дескриптор <AUDIO> додає, відтворює і управляє настройками аудіозаписів на веб-сторінці. Шлях до файлу задається через атрибут SRC або вкладений дескриптор <SOURCE>. Всередині контейнера <AUDIO> можна написати текст, який буде виводитися в браузерах, які не працюють з цим дескриптором. Список підтримуваних браузерами кодеків обмежений.

Для універсального відтворення в різних браузерах аудіо кодують за допомогою різних кодеків і додають файли одночасно через дескриптор <SOURCE>.

З усіх атрибутів обов'язковим є тільки атрибут SRC. В табл. 10.1 наведено опис атрибутів дескриптора <AUDIO>

Таблиця 10.1 – Опис атрибутів дескриптора <audio>

Атрибут	Значення	Тлумачення
SRC	URL	Адреса аудіофайла
AUTOPLAY		Звук починає грати відразу після завантаження сторінки
CONTROLS		Додає панель управління до аудіофайлу
LOOP		Повторює відтворення звуку з початку після його завершення
PRELOAD	NONE, METADATA, AUTO	Використовується для завантаження файлу разом із завантаженням веб-сторінки

В наступному прикладі наведений лістинг файлу, який ілюструє використання аудіозаписів:

```

<!DOCTYPE html>
<html>

```

```

<head>
  <meta charset="utf-8">
  <title>audio</title>
</head>
<body>
  <p>Александр Клименков - Четырнадцать</p>
  <audio controls>
    <source src="audio/music.ogg" type="audio/ogg; codecs=vorbis">
    <source src="audio/music.mp3" type="audio/mpeg">
    Тер audio не поддерживается вашим браузером.
    <a href="audio/music.mp3">Скачайте музыку</a>.
  </audio>
</body>
</html>

```

Питання для самоперевірки.

1. Який дескриптор управляє настройками аудіозаписів на веб-сторінці?
2. Який атрибут є обов'язковим для дескриптора <AUDIO>?
3. Яким є тлумачення атрибуту AUTOPLAY?
4. Яким є тлумачення атрибуту PRELOAD?
5. Яким є тлумачення атрибуту LOOP?

11. Використання елементів анімації

Якісна анімація допомагає в перегляді інформації веб-сторінок. Це одна з основ веб-дизайну наших днів. Взаємодія користувачів з сучасними веб-сайтами серйозно пов'язана із анімацією яка такою, що направляє увагу. Також вона допомагає користувачеві побачити результат його дій і може впливати на його поведінку.

Тому, проектуючи сторінки, необхідно з самого початку пам'ятати про інтерактивну природу веб-простору і сприймати її прояви як природну частину дизайну.

11.1 Gif-Анімація

Gif-Анімація – різновид анімації, заснований на можливості стандарту графічних файлів gif зберігати кілька картинок в одному файлі. Ці картини утворюють кадри, при їхній послідовній зміні створюється анімаційний ефект. Створити анімаційний ролик у форматі gif нескладно. Досить часто для цих цілей використовують графічний редактор у комбінації з програмою-аніматором, а також спеціалізовані сайти.

Принцип створення gif-ролика такий же, як і принцип мальованої анімації: спочатку в графічному редакторі створюється зображення, потім у програмі-аніматорі - кадри анімації, які містять змінене зображення. Далі ці кадри вибудовуються в певній послідовності, для кожного кадру задається час показу – і ролик готовий. Для створення плавного й досить реалістичного ефекту руху досить установити час показу кожного кадру в інтервалі від 0,2 до 1 с.

Для вставки в текст Web-сторінки анімації використовується дескриптор як в параграфі 6.3.

11.2 Рухомий рядок

Дескриптор <MARQUEE> створює рухомий рядок на сторінці. Насправді вміст контейнера не обмежується рядками і дозволяє переміщати (скролювати) будь-які елементи веб-сторінки – зображення, текст, таблиці, елементи форм тощо. Переміщення можна задати не тільки за горизонталлю, але і вертикаллю, в цьому випадку вказуються розміри області, в якій буде відбуватися рух.

В табл. 11.1 наведено опис атрибутів дескриптора <MARQUEE>.

Таблиця 11.1 – Опис атрибутів дескриптора <marquee>

Атрибут	Значення	Тлумачення
BEHAVIOR	Alternate, scroll, slide	Задає тип руху вмісту
BGCOLOR	Задається за правилами, описаними в параграфі 6.1	Колір фону
DIRECTION	Down, left, right, up	Вказує напрямок руху вмісту
HEIGHT	Будь-яке ціле додатне число в пікселях чи в відсотках	Висота області прокрутки
HSPACE	Будь-яке ціле додатне число в пікселях	Горизонтальні поля навколо контенту
VSPACE	Будь-яке ціле додатне число в пікселях	Вертикальні поля навколо вмісту
LOOP	Будь-яке ціле додатне число або -1 для нескінченного руху	Задає кількість повторних прокручувань вмісту
SCROLLAMOUNT	Будь-яке ціле додатне число в пікселях	Швидкість руху контенту
SCROLLDELAY	Будь-яке ціле додатне число в мілісекундах	Величина затримки між рухами.
TRUESPEED		Скасовує вбудований обмежувач швидкості
WIDTH	Будь-яке ціле додатне число в пікселях чи в відсотках	Ширина області прокрутки

В наступному прикладі наведений лістинг файлу, який ілюструє використання рухомого рядка (рис. 11.1):

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Приклад рухомого рядка</title>
</head>
<body>
  Приклад рухомого рядку, який рухається між правим і лівим краєм
  <br>
  <marquee behavior="alternate" direction="left" bgcolor="#ffcc00">
  Інформація, яка рухається весь час ...
  </marquee>
</body>
</html>
```

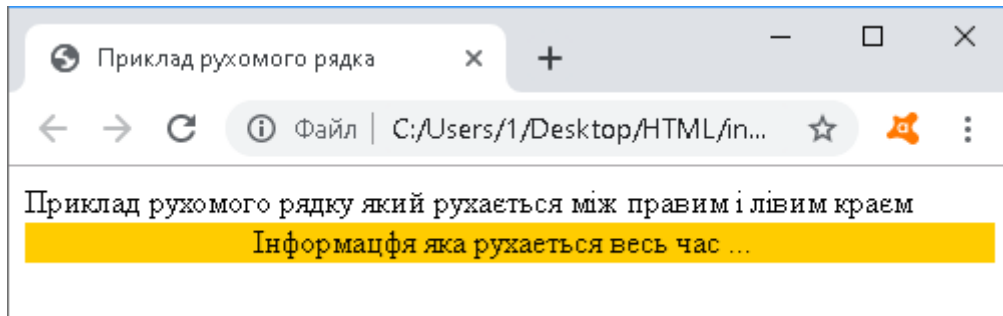


Рисунок 11.1 – Приклад рухомого рядка

Ось ще цікавий приклад, як за допомогою дескриптора `<MARQUEE>` можна створити слайдер в якому кожен малюнок є посиланням:

```
<marquee scrollamount="10">
  <a href="URL сторінки №1"></a>
  <a href="URL сторінки №2"></a>
  <a href="URL сторінки №3"></a>
</marquee>
```

Питання для самоперевірки.

1. Що представляє собою Gif-Анімація?
2. Яким є принцип створення gif-ролика?
3. Яким є тлумачення атрибуту дескриптора `<MARQUEE>` BEHAVIOR?
4. Яким є тлумачення атрибуту дескриптора `<MARQUEE>` SCROLLAMOUNT?
5. Яким є тлумачення атрибуту дескриптора `<MARQUEE>` SCROLLDELAY?

СПИСОК ВИКОРИСТАНИХ ПОСИЛАНЬ

1. Гевлич І.Г. Робоча програма з навчальної дисципліни «Економічна інформатика». Вінниця: ДонНУ імені Василя Стуса, 2019.
2. Бородкіна І. Л., Бородкін Г. О. Web-технології та web-дизайн: застосування мови HTML для створення електронних ресурсів: навч. посіб. для студентів ВНЗ. К.: КНУКіМ, 2017. 329 с.
3. Базурін В. М. WEB-програмування. HTML і CSS: навч. посіб. для студентів. Суми : Цьома С. П., 2016. 91 с.
4. Ємець Є. М. Основи Інтернет: навч. посіб. Полтава: ПУЕТ, 2014. 236 с.
5. Дакетт Д. HTML и CSS. Разработка и дизайн веб-сайтов. М.: Издательство Эксмо, 2018. 480 с.
6. Роббинс Дж. HTML5. Карманный справочник. М.: Издательство Вильямс, 2015. 192 с.
7. <https://www.w3.org>.
8. <https://developer.mozilla.org/uk/#>.
9. <https://w3schools.com>.
10. <https://css.in.ua/>