

2. CSS

Технологія каскадних таблиць стилів (Cascading Style Sheets, CSS) була розроблена як доповнення до HTML з метою визначення зовнішнього вигляду документів і збереження за HTML тільки функції структурної розмітки документів. Система CSS незалежна від HTML, має інший синтаксис і дозволяє задавати параметри графічного (також текстового і звукового) подання дескрипторів HTML.

Таблиці стилів – це набір елементів оформлення, які застосовуються до різних частин документа й описують спосіб їх подання на екрані. Таблиці стилів реалізовані у всіх функціонально-розвинених текстових процесорах.

За допомогою таблиць стилів можна форматувати текст, використовуючи методи, наближені до методів форматування звичайних друкованих сторінок. Створені сторінки будуть коректно функціонувати, враховуючи деякі обмеження, пов'язані із платформами, браузерами, різними розмірами екранів і поданнями моніторів.

Можна виділити три основні переваги застосування таблиці стилів:

1. Можна створити таблицю стилів і застосувати її до одного або декількох документів. Ця універсальність надає змогу змінювати вигляд усіх сторінок, змінюючи лише таблицю стилів у файлі.

2. Надання більшого контролю керування текстом: з'явилися нові властивості, за допомогою яких можна створювати спливаючі меню, розміщати один текст над іншим (причому незалежно від їх послідовності у коді HTML), створювати ефекти тіні для тексту та ін.

3. Таблиці стилів на відміну від інших методів управління зовнішнім виглядом дозволяють зберігати в різних місцях текст та інформацію про те, як він повинен виглядати, що дозволяє зменшити розміри файлів документів.

Прийняття в 1996 році Консорціумом W3C CSS першого рівня як стандарту дозволило відокремити зміст Web-сторінки

(текст, графічні зображення тощо) від її оформлення (макет сторінки й характеристики тексту, наприклад шрифти тощо). Після цього мова HTML стала функціонально-орієнтованою. Стандарт CSS2, прийнятий в 1998 році, заснований на CSS першого рівня й дозволяє розроблювачам здійснювати контроль над Web-сторінками на більш високому рівні. Він включає деякі нові функції, зокрема, можливість точно розташовувати елементи й об'єкти Web-сторінки, застосовувати шрифти, що завантажуються, або використовувати звукові таблиці стилів. Застосування стилів дозволяє також вирішити багато проблем підтримки браузерів, тому що основні компанії-розроблювачі браузерів вмонтували таблиці CSS у свої програмні продукти.

Основні властивості елементів CSS подано у таблиці 1.

Таблиця 1

Список основних властивостей елементів CSS

Властивість	Опис	Приклад
1	2	3
{background: url(картинка)}	Визначає фонову картинку, що знаходиться позаду тексту	TD {background: url(img/logo-3.gif);}
{background-color: значення кольору}	Визначає колір фону	A {background-color: #FFFFFF;}
{border-bottom-color: color} аналогічно border-top-color, border-left-color border-right-color border-color	Визначає колір нижньої (bottom), верхньої (top), лівої (left), правої (right) лінії рамки. border-color – визначає колір одразу всіх ліній	TD {border-bottom-color: #00FFFF; border-top-color: #0000FF; border-left-color: #800080; border-right-color: aqua;}
{border-bottom-style: none або solid або double або groove або ridge або inset або outset}	Визначає стиль нижньої (bottom), верхньої (top), лівої (left), правої (right) лінії рамки. border-style – визначає стиль одразу всіх ліній	TD {border-top-style: solid; }
{border-bottom-width: thin або medium або thick}	Визначає ширину нижньої (bottom), верхньої (top), лівої (left), правої	TD {border-width: 3px; }

або ширина в пікселях (дюймах, пунктах, см, мм)}	(right) лінії рамки. border-width – визначає ширину одразу всіх ліній	
{color: значення кольору}	Визначає колір переднього плану елемента	H1 {color: red;}
{font-family: найменування шрифту}	Визначає тип шрифту	A {font-family: Arial;}
{font-size: розмір}	Визначає розмір шрифту в абсолютних (пікселі, см та ін.) або у відносних величинах (%)	BODY {font-size: 5cm;}
{font-style: normal або italic}	Визначає нормальний або курсивний стиль шрифту	INPUT {font-style: italic;}
{font-weight: normal або bold або bolder або lighter}	Визначає жирність шрифту	INPUT {font-weight: bold;}
{height: число} {width: число}	Встановлює, відповідно, висоту та ширину елемента	TD {height: 20px;}
{margin-left: число} аналогічно: margin-right, margin-top, margin-bottom, margin	Задає лівий (left), правий (right), верхній (top) або нижній (bottom) відступи від краю. margin – задає величину одразу усіх відступів	BODY {margin: 0px;}
{top: число} {left: число}	Задає, відповідно, координату Y та X елемента	DIV.my {top: 10px; left: 100px;}
{padding-left: число} аналогічно: padding-right padding-top padding-bottom padding	Задає розмір лівої (left), правої (right), верхньої (top) або нижньої (bottom) відстані між рамкою та змістом. padding – задає величину одразу усіх відстаней	TD {padding-top: 10px;}
{text-align: left або right або center або justify}	Встановлює вирівнювання тексту відносно лівої та правої межі вікна або елемента	TD {text-align: center;}
{text-decoration: none або [overline або(та) underline]}	Визначає підкреслений або (та) закреслений текст, або їх відсутність	A {text-decoration: none;}

{vertical-align: top або bottom або middle}	Встановлює вирівнювання тексту відносно верхньої та нижньої межі вікна або елемента	TD {vertical-align: middle;}
{z-index: число;}	Встановлює порядок перекривання елементів	<DIV STYLE="z-index:-1">

Існує три способи застосування таблиць стилів у документі HTML.

- **Вбудовування (Inline)**, при якому дескриптори Web-сторінки доповнюються короткими оголошеннями формату. Вбудовування надає контроль над фрагментом, до якого воно застосовується. Атрибут style приєднується до дескриптора HTML і відповідний фрагмент сторінки буде виводитися браузером із застосуванням формату, зазначеного стилем.

- **Впровадження (Embed)** забезпечує контроль над сторінкою HTML. Використання дескриптора <style> у межах розділу <head> дозволяє описати атрибути стилю.

- **Зв'язування (Link)** відоме також як застосування зовнішнього аркуша стилів. У цьому випадку зв'язаний документ використовує еталону таблицю стилів, що застосовується для всього сайту (зберігається у файлі з розширенням .css).

Термін «каскадні» (таблиці стилів) описує, в першу чергу, той факт, що браузер додержується певного порядку (каскаду) при інтерпретації стилів. Можна використовувати всі три типи стилів і браузер інтерпретує їх у наступному порядку: зв'язаний – впроваджений – вбудований.

Другий аспект комбінованого застосування стилів – спадкування. Наприклад, при оголошенні в елементі абзацу деякого кольору шрифту всі елементи, що розташовані в його межах, успадкують цей колір.

Вбудовування стилю

Опис стилю можна вмонтувати в різні елементи HTML, для яких стиль має сенс, наприклад, у дескриптори абзаців, заголовків, гіперпосилань або комірок таблиці.

При вбудовуванні в елемент додається атрибут `style`, значенням якого є властивість і його значення, розділені двокрапкою. Таких пар («властивість: значення») може бути декілька, у цьому випадку вони розділяються крапкою з комою. Значення атрибута `style` беруться в лапки.

Два елементи `div` (розділ) і `span` (інтервал) є універсальними елементами-контейнерами й дозволяють застосовувати стилі до фрагментів тексту, як це показано у прикладах 1–3:

Приклади 1–3.

Приклад 1

```
<p style=«font-size: 2em; color: #ff0000»>Червоний текст із розміром шрифту 2em</p>
<p style=«font-size: 1em; background-color: #ffff00»>Текст із розміром шрифту 1em і жовтим фоном</p>
```

Приклад 2

```
<div style=«font-size: x-small»>Текст із розміром шрифту x-small
<span style=«font-size: large»>Текст із розміром шрифту large</span></div>
<p style=«font-size: 12pt; font-variant: small-caps»>Текст із розміром шрифту 12pt і малими прописними буквами</p>
<p style=«font-size: 16pt; font-family: Garamond, Georgia, serif»>Текст із розміром шрифту 16pt і Garamond або Georgia шрифтом</p>
```

Приклад 3

```
<p style=«font: bold italic 14pt Arial, sans-serif»>Текст зі скороченою формою запису властивостей шрифту</p>
```

На рис.1 показано відображення у браузері сторінки з текстами прикладів 1–3.

Приклад 1

Червоний текст із розміром шрифту 2em

Текст із розміром шрифту 1em і жовтим фоном

Приклад 2

Текст із розміром шрифту x-small Текст із розміром шрифту large

ТЕКСТ ІЗ РОЗМІРОМ ШРИФТУ 12PT І МАЛИМИ ПРОПИСНИМИ БУКВАМИ

Текст із розміром шрифту 16pt і Garamond або Georgia шрифтом

Приклад 3

Текст зі скороченою формою запису властивостей шрифту

Рис. 1. Відображення сторінки з текстами прикладів 1–3

Впровадження стилю

Для впровадження стилю використовується елемент (контейнер) `<style>` у межах розділу `<head>`. Можна сховати таблиці стилів від перегляду застарілими браузерями, якщо вміст контейнера `<style>` укласти в коментарі (`<!--...-->`), як показано у прикладі 4.

У наведеному далі прикладі використані правила, за якими повинні бути представлені елементи документа. Кожне правило має дві основні частини: селектор і блок оголошень. Блок оголошень береться у фігурні дужки й може містити одне й більше оголошень. Кожне оголошення являє собою комбінацію властивості й значення, які розділяються двокрапкою. У середині блоку оголошень може бути задано кілька оголошень, вони відділяються один від одного крапкою з комою.

Приклад 4

```
<html><head>
<title>Приклад 4</title>
<style type=«text/css»>
```

```

<!--
body {
background:#000000;color:#ffffff
}
h1 {
font:14pt Verdana; color:yellow
}
-->
</style>
</head>
<body>
<strong>Приклад 4</strong><br><br>
    Впроваджений стиль для розділу body забезпечує виведення
білих символів на чорному фоні<br>
    <h1> Для виводу заголовка використовується впроваджений
стиль: шрифт Verdana жовтого кольору й розміром 14 пунктів
</h1>
</body></html>

```

У контейнері `<style>` даного прикладу знаходяться описи впроваджених стилів. Кількість стилів може бути довільною. Атрибут `type` указує на тип таблиць стилів, що використовується – CSS.

На рис. 2 показано відображення у браузері сторінки з текстом прикладу 4.

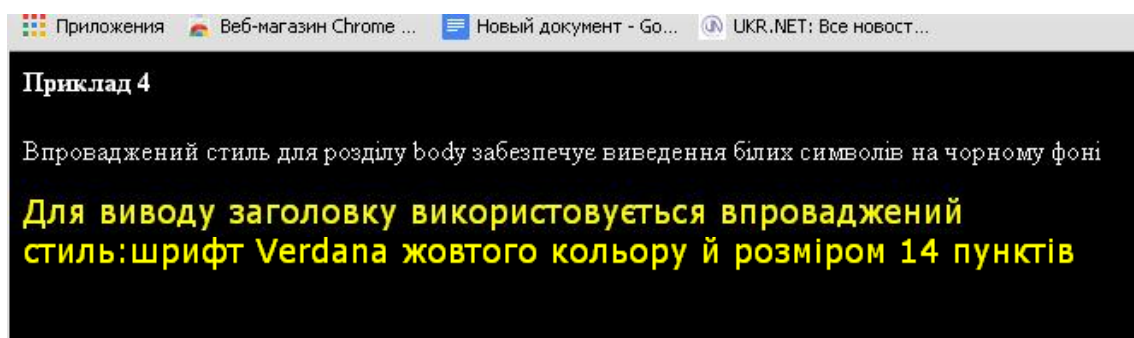


Рис. 2. Відображення в браузері сторінки з текстом прикладу 4

Використання різних селекторів для стильового оформлення.

Селектори елементів

Селектори `body` і `h1` з прикладу 4 є селекторами елементів і визначають область застосування даного стилю. Селектор `body` вказує на те, що цей стиль буде застосований до розділу `body`, а селектор `h1` на те, що стиль буде застосований до тексту усередині всіх елементів `h1` документа. Таким чином, можна оголосити різні стилі для заголовків і для іншого тексту.

У прикладі 4 парами властивість-значення для селектора `body` є `background:#000000` і `color:#ffffff`, а для селектора `h1` – `font:14pt Verdana` і `color:yellow`.

У правилі допускається групування декількох селекторів при одному блоці оголошень стилю або ж групування властивостей і значень в оголошенні стилю.

У прикладі 5 показано, що для того, щоб додати однакові властивості заголовку `h5` і абзацу `p`, використовується групування селекторів.

Приклад 5

```
<style type=«text/css»>
<!--
h5, p {
font-family:Garamond;
font-size:14pt;
color:#660066;}
-->
</style>
```

У прикладі 6 показано групування властивостей і значень, що належать до шрифтового оформлення.

Приклад 6

```
<style type=«text/css»>
<!--
body {
font: italic bold 15pt/18pt Verdana,sans-serif;}
-->
```


-->
</style>

При такому групуванні повинен підтримуватися певний порядок проходження значень властивостей шрифту: нахил (зображення) шрифту, насиченість шрифту, розмір символів, коса риска, міжрядковий інтервал, сімейство шрифтів.

Селектори класів

За необхідності застосування правила стилів незалежно від елементів використовуються селектори класів. Класи дозволяють також створити кілька наборів стилів на базі одного елемента HTML. Назва класу випереджається крапкою.

У прикладі 7 клас, що дозволяє при необхідності застосувати жовтий фон до комірки таблиці (елемент `td`), має селектор виду: `td.yellow`. Клас `black` того ж прикладу не прив'язаний до конкретного елемента й може бути застосований до будь-якого елемента тексту. У даному прикладі клас `black` застосований для створення фонів комірок таблиці в шаховому порядку. Для задання ширини й висоти таблиці застосований селектор елемента `table`.

Для застосування класу використовується атрибут `class` (приклад 7). Усі комірки, яким привласнений клас стилю `yellow`, будуть мати жовтий фон. Аналогічно, комірки, яким привласнений клас `black`, будуть мати чорний фон.

Клас `black` застосований також і до елемента `p`, текст якого розміститься на чорному фоні.

Приклад 7

```
<html><head><title>Приклад 7</title>  
<style>  
td.yellow {background-color:yellow;}  
.black {background-color:black;}  
table {width:45%;height:30%;}  
</style>  
</head>
```

```

<body>
<strong>Приклад 7</strong><br>
<table border=2px>
<tr>
    <td class=«black»>&nbsp;</td>
    <td class=«yellow»>&nbsp;</td>
    <td class=«black»>&nbsp;</td>
</tr>
<tr>
    <td class=«yellow»>&nbsp;</td>
    <td class=«black»>&nbsp;</td>
    <td class=«yellow»>&nbsp;</td>
</tr>
<tr>
    <td class=«black»>&nbsp;</td>
    <td class=«yellow»>&nbsp;</td>
    <td class=«black»>&nbsp;</td>
</tr>
</table>
<p class=«black» style=«color:white;text-align:center»> До абзацу
застосований клас black і вбудований стиль вирівнювання вмісту
по центру з білим кольором символів</p>
</body></html>

```

На рис. 3 показано відображення тексту прикладу 7 у браузері.

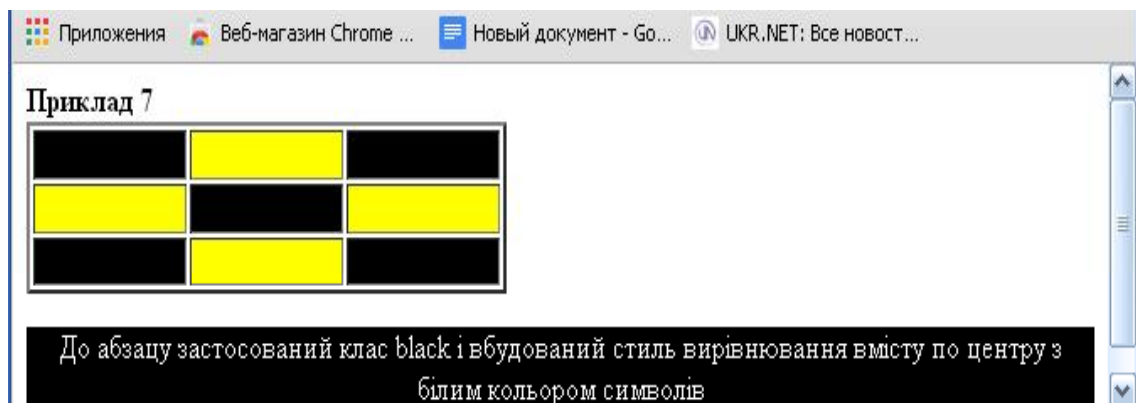


Рис. 3. Відображення тексту у браузері прикладу 7

Селектори ідентифікаторів

Селектори класів можна застосовувати до будь-якої кількості елементів. У правилі стилю можна використовувати селектор ідентифікатора (селектор ID). На відміну від селектора класу селектор ID повинен бути унікальний для даного документа. Селектор ID починається із символу #. Замість атрибута class у відповідних елементах HTML використовується атрибут id. Його застосування показано в прикладі 8. Селектор #pict використовується для позиціонування графічного зображення, яке поміщено в контейнер <div>.

Приклад 8

```
<html><head><title>Приклад 8</title>
<style type=«text/css»>
#pict {
position:absolute;
top:100;
left:50;}
</style>
</head>

<body>
<strong>Приклад 8</strong>
<div id=«pict»>
<img src=«метелик.jpg» alt=«cats» width=100 height=100>
</div></body></html>
```

На рис. 4 показано відображення тексту прикладу 8 у браузері.



Рис. 4. Відображення тексту у браузері прикладу 8

Селектори нащадків

Селектори нащадків записуються через пропуск після батьківського селектора, наприклад, `h1 em {color:red;}`. Це правило зробить червоним контент елемента `em`, який є нащадком елемента `h1`. У селекторі нащадків може бути використано кілька елементів, наприклад, `ul li a {color:red;}`. Правило застосує червоний колір до посилань, що перебувають в елементах `li` маркованого списку.

Селектори атрибутів

Селектори атрибутів можуть застосовуватися для вибору елементів на підставі їх атрибутів і значень цих атрибутів. Селектори атрибутів були введені в стандарті CSS2 і уточнювалися в стандартах CSS2.1 і CSS3.

Для вибору всіх заголовків `h1`, що мають атрибут `class` і фарбування тексту цих заголовків у сірий колір, можна використовувати правило `h1[class] {color:gray}` або для виділення напівжирним шрифтом текст будь-якого гіперпосилання, яке має й атрибут `title` і атрибут `href` можна записати `a[title][href] {font-weight: bold}`.

Вибір на підставі конкретного атрибута проводиться таким чином: `a[title=«Goto GUT»] {font-weight: bold}`. Напівжирним виділенням будуть відключені гіперпосилання, значенням атрибута `title` яких буде `Goto GUT`.

Можливий також вибір за частковими значеннями атрибута, наприклад, `img[title~=«Малюнок»]{border:1px solid gray}`. Це правило забезпечить малювання рамки навколо зображень, в атрибуті `title` яких міститься слово «Рисунок».

Псевдокласи й псевдоелементи

У **CSS2.1** визначені псевдокласи й псевдоелементи. Традиційно псевдокласи використовуються для оформлення гіперпосилань. Вони описують властивості посилань невідвідуваних (**:link**), відвідуваних (**:visited**), над якими перебуває курсор миші (тобто властивості посилань «при наведенні» курсору миші) (**:hover**) і активних посилань (**:active**).

У прикладі 9 наведений код **HTML**, що застосовує псевдокласи для елемента **<a>**. У даному прикладі невідвідувані посилання мають чорний колір, активні посилання фарбуються в синій, відвідувані – у зелений, а при наведенні миші на посилання колір її міняється на червоний і розмір шрифту збільшується до 16 пунктів (рис. 5). Порядок розташування псевдокласів у селекторі важливий, тому що при його порушенні принцип каскадності застосування стилів може привести до ігнорування властивостей окремих псевдокласів.

Приклад 9

```
<html><head><title>Пример 9</title>
<style type=«text/css»>
<!--
a:link {color:black;}
a:visited {color:#00ff00;}
a:hover {color:#ff0000;font-size:16pt;}
a:active {color:#0000ff;}
-->
</style>
</head>
<body>
<strong>Приклад 9</strong><br>
<a href=«класи_1.html» > Перехід до файла «класи_1.
html»</a><br>
<a href=«класи.html» > Перехід до файла «класи.html»</a>
</body></html>
```

На рис. 5 показано відображення у браузері результату дії псевдокласу: **hover** – наведення миші на друге гіперпосилання.

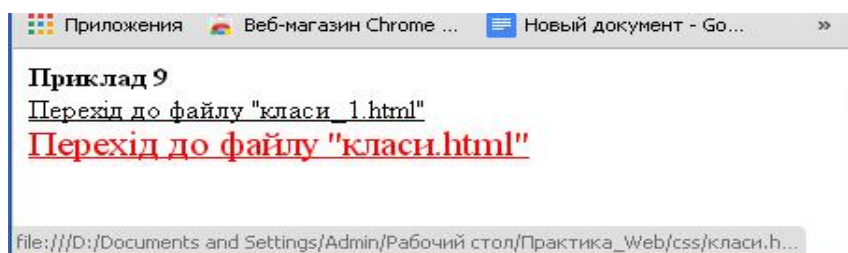


Рис. 5. Відображення в браузері результату дії псевдокласу: **hover**

До псевдокласів відносять також **focus**. Цей псевдоклас ставиться до будь-якого елемента, до якого в даний момент належить фокус вводу, тобто який готовий приймати введення з клавіатури або може бути активований іншим способом.

Псевдоелементи вводять фіктивні елементи в документ, щоб досягти певних ефектів. У **CSS2.1** визначено чотири псевдоелементи, для яких можуть бути застосовані спеціальні стилі: перша буква (**:first-letter**), перший рядок (**:first-line**), до (**:before**) і після (**:after**) елемента.

Наприклад, правило **h3:first-letter {font-size: 150 %}** збільшить першу букву заголовка в півтора раза стосовно іншого тексту. А правило **p:first-line {color:red}** зробить символи першого рядка абзацу червоними.

Приклади 10 і 11 ілюструють використання псевдоелементів **:first-letter** і **:first-line**.

Приклад 10

```
<html><head> <title>Приклад 10</title>
<style type=«text/css»>
p:first-letter{
font-size:3em;
font-weight:bold;
color:#ff0000;}
p {line-height:1em;text-align:justify;}
</style>
</head>

<body>
<strong>Приклад 10</strong>
<p>Псевдоелемент first-letter використовується для створення буквиц в абзаці. Абзац має буквицу розміру 3ем і накреслення bold.</p>
<p style=«text-indent:200px»>Абзац має буквицу розміру 3 ем, накреслення bold і відступ першого рядка 200px. Абзац має буквицу розміру 3 ем, накреслення bold і відступ першого рядка 200px.</p>
</body></html>
```

На рис. 6 показано відображення прикладу 10 у браузері.

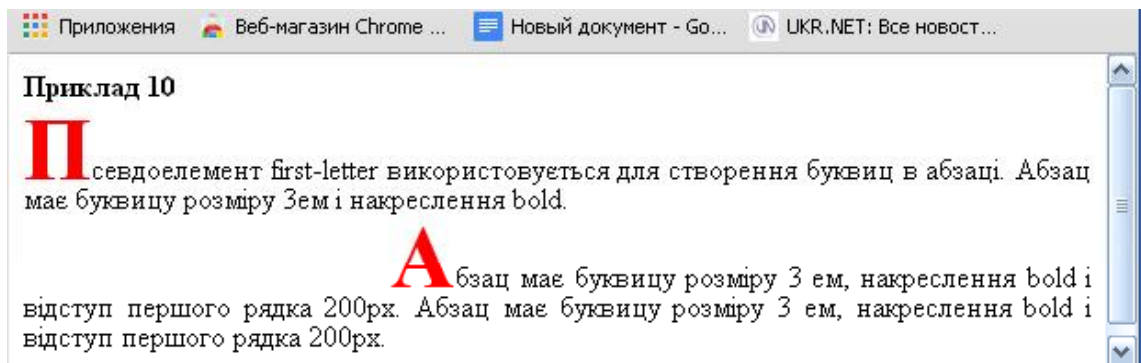


Рис. 6. Відображення у браузері прикладу 10

Приклад 11

```
<html><head> <title>Приклад 11</title>
<style type=«text/css»>
p:first-line{
font-weight:bold;}
p {line-height:1em;text-align:justify}
</style>
</head>
<body>
<strong>Приклад 11</strong>
```

<p>Перші рядки абзаців стилізовані за допомогою псевдоелементу p:first-line. Перші рядки абзаців стилізовані за допомогою псевдокласу p:first-line. </p>

<p > Перші рядки абзаців стилізовані за допомогою псевдоелементу p:first-line. Перші рядки абзаців стилізовані за допомогою псевдокласу p:first-line. </p>

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

На рис. 7 показано відображення прикладу 11 у браузері.

Псевдоелементи: **before** і **:after** дозволяють вставляти згенерований вміст, а потім застосовувати до нього спеціальні стилі. Цей вміст вставляється за допомогою властивості **content**, наприклад, **h2:before {content: «ГУТ»}** додає перед заголовками другого рівня текст «ГУТ» або для **body:after {content: «The end»}** додає в кінець документа текст «The end».

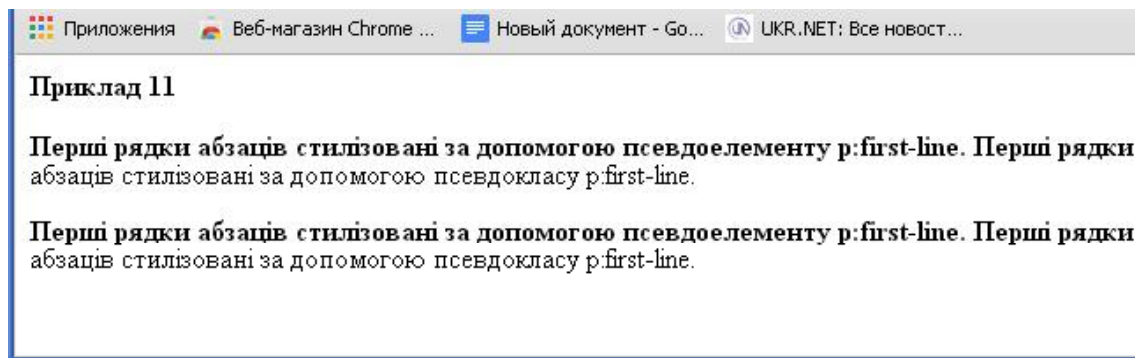


Рис.7. Відображення у браузері прикладу 11

Зв'язування стилю

При зв'язуванні стилю всі стильові описи проводяться за тими ж правилами, що й при впровадженні, але зберігаються вони в окремому файлі, що має розширення `.css`. Сам файл повинен перебувати в кореневому каталозі сайту, а якщо ні, то потрібно коректно вказати зв'язок з ним.

Перевагою зв'язування є те, що стильові описи застосовуються до всіх сторінок сайту й за необхідності зміни оформлення досить внести нововведення у файл, що містить ці описи, замість того, щоб виправляти всі сторінки сайту.

У наведеному нижче прикладі ілюструється використання зовнішньої таблиці стилів.

Приклад. Файл із іменем `my_style.css` містить *тільки* правила стилів:

```
body {  
background: #000000;  
color: #ffffff;  
}  
a {  
color: #ff0000; text-decoration:none;  
}
```

Документ **HTML**, який посилається на цей файл, повинен містити в частині **head** наступне посилання, що використовує елемент **link**:

```
<html><head>
```



```
<link rel=stylesheet href=«my_style.css» type=«text/css»>
</link>
```

Аналогом елемента **link** є директива **@import url (url)**. Директива **@import** також вказує браузеру на необхідність завантаження зовнішньої таблиці стилів. Розміщається директива в контейнері **<style>** перед іншими правилами **CSS**. Приклад використання директив **@import** :

```
<style>
@import url («sheets22.css»);
@import url («http://example.org/library/layout.css»);
@import url («printer.css») print;
</style>
```

Із прикладу видно, що може бути використано кілька директив **@import** для приєднання декількох файлів з таблицями стилів. В останній директиві зазначене обладнання, для якого створена приєднана таблиця.

Текстові властивості CSS

1. **text-align** – вирівнювання тексту із значеннями: **left** (по лівому краю), **right** (по правому краю), **center** (по центру), **justify** (по обох краях).

2. **text-indent** – відступ у першому рядку блоку (абзацу, розділу) зі стандартними значеннями довжини (**pt**, **px**, **cm**, **mm**).

3. **text-decoration** – оформлення тексту підкресленням зі значеннями: **none** (відсутнє – за замовчуванням), **underline** (підкреслення), **overline** (лінія над текстом), **line-through** (підкреслення), **blink** (мерехтіння).

4. **text-transform** – переклад букв у верхній або нижній реєстр зі значеннями: **none** (відсутнє – за замовчуванням), **capitalize** (перша буква кожного слова стає прописною), **uppercase** (переводить усі букви у верхній реєстр), **lowercase** (переводить усі букви в нижній реєстр).

5. **text-shadow** – установка ефекту затемнення тексту зі значеннями: **none** (відсутнє – за замовчуванням), **color left top radius** (колір затемнення з відстанями ліворуч (праворуч), униз (нагору) від тексту й радіусом нерізкості).

6. **letter-spacing** – відстань між символами тексту зі стандартними значеннями довжини (**pt, px, cm, mm, em,ex**).

7. **word-spacing** – відстань між словами зі стандартними значеннями довжини (**pt, px, cm, mm**).

Крім стандартних одиниць виміру довжини **px, pt** (дорівнює 0,35 мм), **mm, cm**, може використовуватися **em** (**1em** дорівнює ширині букви **m** у шрифті заданого розміру), **ex** (**1ex** дорівнює висоті шрифту), **px** (**1px** дорівнює ширині пікселя).

Приклад 12

```
<html><head><title>Приклад 12</title>
<style type=«text/css»>
<!--
p.class1 {
text-align: justify;
text-indent: 20pt;
color: #c0c0c0;
background-color: #000000;
}
.class2 {
text-decoration: underline;
}
.class3 {
letter-spacing: 0.5em;
}
.class4 {
text-transform: uppercase;
}
-->
</style>
</head>
<body>
<strong>Приклад 12<b/>
```

```

<p class=«class1»>Текст абзацу вирівнюється по ширині,
має відступ першого рядка 20 пунктів і використовує
білий колір символів на чорному фоні</p>
<p>У тексті абзацу <span class=«class2»>використовується
підкреслення, </span>
<span class=«class3»> збільшення відстані між символами</span>
<span class=«class4»>, а також відображення символів
у верхньому регістрі</span></p>
</body></html>

```

На рис. 8 показано відображення тексту прикладу 12 у браузері.

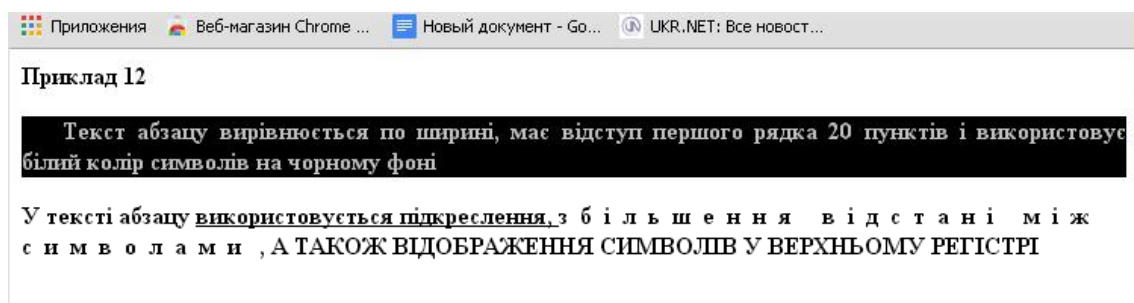


Рис. 8. Відображення тексту у браузері прикладу 12

Колір і фон

1. **color** – колір тексту – будь-яке значення, що відповідає стандарту.

2. **background-color** – колір фону – будь-яке значення, що відповідає стандарту.

3. **background-image:URL(«URL»)** – фонове зображення – будь-яке значення **URL**, що відповідає стандарту.

4. **background-repeat** – напрям повторення фонового зображення зі значеннями: **repeat** (повторення по горизонталі й вертикалі – за замовчуванням), **repeat-x** (повторення тільки по горизонталі), **repeat-y** (повторення тільки по вертикалі), **no-repeat** (відсутність повторення).

5. **background-position** – положення фонового зображення щодо верхнього лівого кута утримуючого його елемента зі значеннями: відстань по горизонталі, відстань по вертикалі в стандартних одиницях довжини або в процентному співвідношенні.

Значення 50 % по горизонталі й 50 % по вертикалі дає розташування фонового зображення по центру. Розміщення фонового зображення може бути також **top** (по верхньому краю), **center** (по центру), **bottom** (по нижньому краю), **left** (по лівому) і **right** (правому) краях.

6. **background-attachment** – прокручування фонового зображення разом з документом зі значеннями: **scroll** (прокручування – за замовчуванням) або **fixed** (фіксація) зображення у вікні браузера.

7. Для опису одразу всіх параметрів фону можна використувати властивість `background: background-color background-image background-repeat background-attachment background-position`

Приклад 13_1

```
<html><head><title>Приклад 13_1 </title>
<style>
p {background-image: url(fox.jpg);
  background-position:center;
  border: 1px dotted gray;}
p.c1 {background-repeat: repeat-y;}
p.c2 {background-repeat: repeat-x;}
</style>
</head>

<body><strong>Приклад 13_1</strong>
  <p class=«c1»> У даному абзаці фонове зображення позиціо-
  новане по центру і поширюється по вертикалі. </p>
  <p class=«c2»> У даному абзаці фонове зображення позиці-
  оноване по центру і поширюється по горизонталі. </p>
</body></html>
```

На рис. 9 показано відображення тексту прикладу 13_1 у браузері.

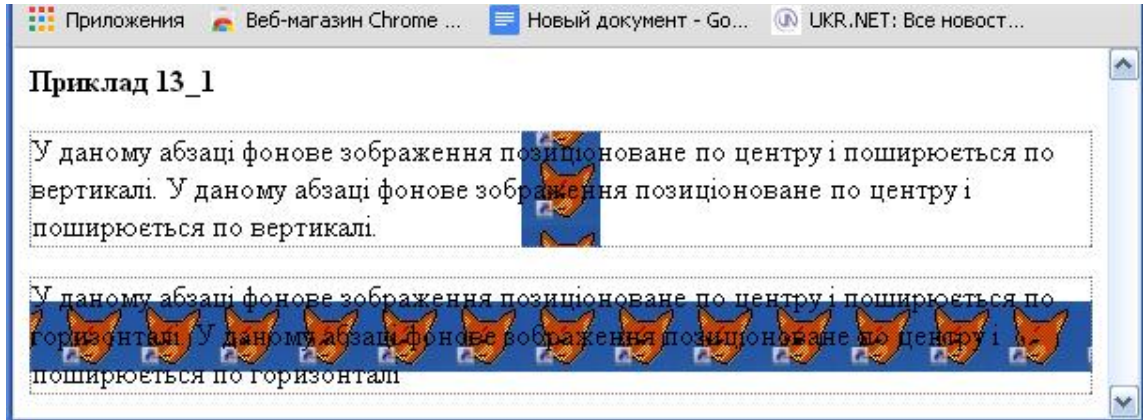


Рис. 9. Відображення тексту у браузері прикладу 13_1

У прикладі 13_2 фонове зображення позиціоноване на 25 % по горизонталі й, за замовчуванням, по центру по вертикалі.

Приклад 13_2

```
<html><head><title>Приклад 13_2</title>
<style>
p {background-image: url(fox.jpg);
  background-position:25%;
  background-repeat: no-repeat;
  border: 2px dotted gray;}
</style>
</head>
<body>
<strong>Приклад 13_2</strong><br>
  <p>У даному абзаці фонове зображення позиціоноване на
25 % по горизонталі і по центру по вертикалі. </p>
</body></html>
```

На рис. 10 показано відображення тексту прикладу 13_2 у браузері.

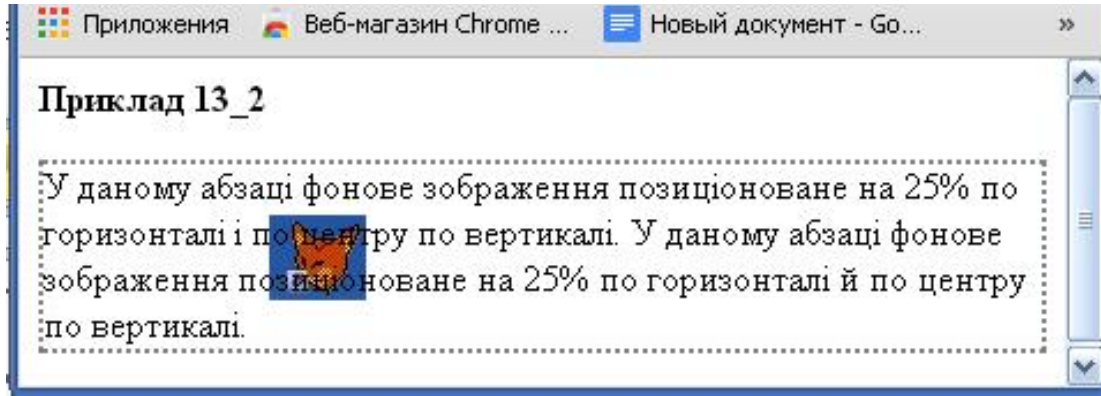


Рис. 10. Відображення тексту у браузері прикладу 13_2

Шрифти

1. **font-family** – сімейство шрифтів. Може бути кілька сімейств, відділених один від одного комами. Пріоритет визначається порядком у списку. Значенням може бути ім'я типового шрифту (**serif/sans-serif/cursive/fantasy/monospace**), а також шрифту, що належить одному з типів.

2. **font-style** – зображення шрифту зі значеннями: **normal** (звичайне за замовчуванням), курсив (**italic**), похиле зображення (**oblique**).

3. **font-variant** – у вигляді малих прописних букв зі значеннями: **normal** (звичайне – за замовчуванням) або **small-caps** (у вигляді малих прописних букв).

4. **font-weight** – товщина виведеного шрифту зі значеннями: **normal** (звичайна – за замовчуванням), **bold** (напівжирний), **bolder** (жирний), **lighter** (світлий) або числовими значеннями: 100-світлий, 400-звичайний, 700-напівжирний, 900-жирний.

5. **font-size** – висота символів (кегель). Значенням є будь-яка відповідна до стандартів висота або процентне значення, що позначає зменшення або збільшення у відсотках від кегля батьківського елемента. Значення абсолютного розміру можуть бути записані у вигляді ключових слів: **xx-small** (занадто малий), **small** (малий), **medium** (середній – за замовчуванням), **large** (великий), **x-large** (дуже великий), **xx-large** (занадто великий).

Значення відносного розміру можуть бути записані у вигляді ключових слів: **larger** (більше) і **smaller** (менше).

6. Для опису одразу всіх параметрів шрифту можна використувати властивість **font: font-style font-variant font-weight font-size font-family**

Використання різних властивостей шрифтів наведено в прикладі 14.

Приклад 14

```
<html><head><title>Приклад 14 </title>
<style type=«text/css»>
p {
font: oblique 18pt Impact;}
p1 {font: large Verdana, sans serif;
color:#0000ff;}
</style>
</head>
<body>
<strong>Приклад14</strong>
<p> Для абзацу застосовано шрифт Impact стилю oblique,
18 пунктів</p>
<p class=«p1»> Для абзацу застосовано шрифт Verdana, large</p>
</body></html>
```

На рис. 11 показано відображення тексту прикладу 14 у браузері.

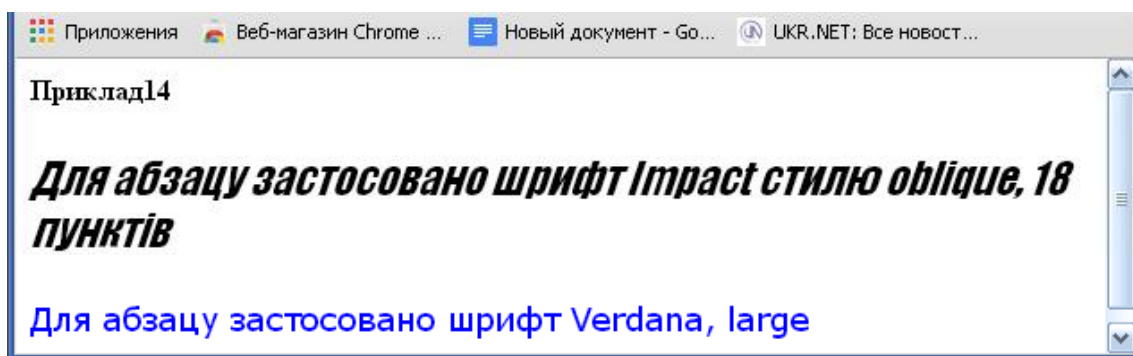


Рис. 11. Відображення тексту у браузері прикладу 14

Блокова модель

1. **margin-top, margin-right, margin-bottom, margin-left** – ширина верхнього, правого, нижнього й лівого поля. Значення за замовчуванням – 0, значеннями є довжини, відповідні до стандарту, або процентне значення, що визначає відношення ширини поля до ширини елемента.

2. **margin** – ширина полів для всіх сторін елемента. У цієї властивості може бути від 1 до 4 значень. Одне значення привласнюється усім полям, із двох значень перше привласнюється *верхньому й нижньому* полю, а друге – *лівому й правому*. При трьох: перше – верхньому полю, друге – лівому й правому, а третє – нижньому.

3. **padding-top, padding-right, padding-bottom, padding-left** – ширина проміжку між вмістом елемента й певною ділянкою його границі. Значення за замовчуванням – 0, значеннями є довжини, відповідні до стандарту, або процентне значення, що визначає відношення ширини проміжку до ширини елемента.

4. **padding** – ширина проміжку для всіх сторін елемента. У цієї властивості може бути від 1 до 4 значень. Одне значення привласнюється всім проміжкам, із двох значень перше привласнюється *верхньому й нижньому* проміжку, а друге – *лівому й правому*. При трьох: перше – верхньому полю, друге – лівому й правому, а третє – нижньому.

5. **border** – рамка. Для опису всіх властивостей рамки можна використовувати конструкцію: **border: border-width border-style border-color.**

Властивостями є **border-width** (ширина рамки), **border-style** (стиль рамки) і **border-color** (колір рамки).

5.1. **border-width** – ширина рамки. Значеннями є: **thin** (тонка лінія), **medium** (середня – за замовчуванням), **thick** (товста), а також стандартні значення ширини.

Можна також встановлювати значення ширини рамки для певної сторони. Для цього використовуються властивості **border-top-width, border-right-width, border-bottom-width, border-left-width** з аналогічними значеннями.

5.2. **border-style** – стиль рамки. Значеннями є: **none** (відсутність – за замовчуванням), **hidden** (схована), **dotted** (пунктир), **solid** (суцільна), **double** (подвійна), **dashed** (штрих-пунктир), **groove** (подвійна борозна), **ridge** (гребінь), **inset** (врізання), **outset** (орнамент).

Можна також установлювати значення стилю рамки для певної сторони. Для цього використовуються властивості **border-top-style**, **border-right-style**, **border-bottom-style**, **border-left-style**.

5.3. **border-color** – колір рамки. Значенням є будь-яке, що відповідає стандарту.

Можна також установлювати значення кольору рамки для певної сторони. Для цього використовуються властивості **border-top-color**, **border-right-color**, **border-bottom-color**, **border-left-color**.

Приклад 15

```
<html><head><title>Приклад 15</title>
<style type=«text/css»>
p {
border: 10px double red;
margin: 50px 100px 100px 50px;
padding:20px }
</style>
</head>
<body>
<strong>Приклад15</strong>
<p>Текст абзацу поміщений у подвійну рамку червоного
кольору, товщиною 10 пікселів. Установлені поля шириною 100
пікселів праворуч і знизу, зверху й ліворуч – 50 пікселів , а про-
міжок між текстом абзацу і рамкою встановлено 20 пікселів з усіх
сторін</p>
</body> </html>
```

На рис. 12 показано відображення тексту прикладу 15 у браузері.

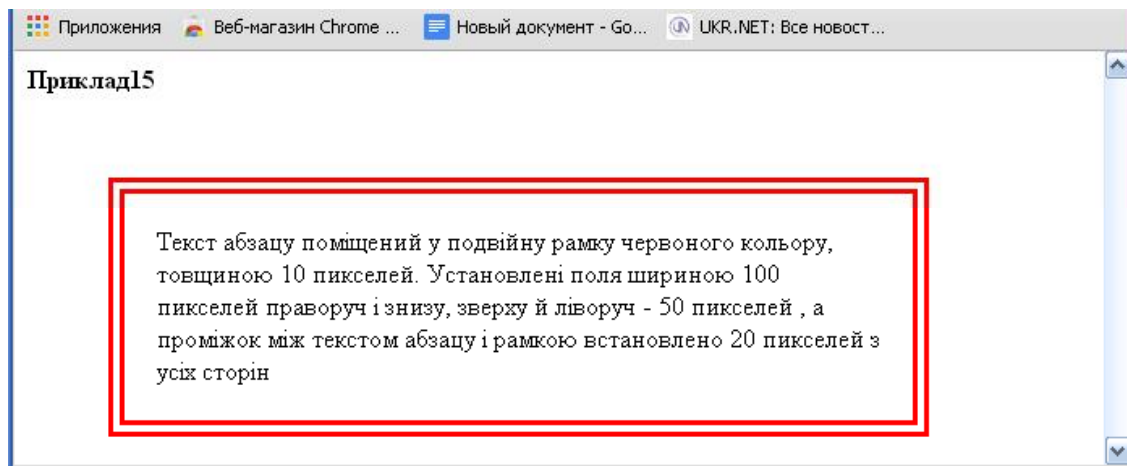


Рис. 12. Відображення тексту у браузері прикладу 15

Плаваюча модель для розміщення зображень й інших елементів

До появи останніх стандартів **CSS** для обтікання текстом зображень користувалися атрибутом **align** елемента **img** зі значеннями **left** і **right**. Зараз стандартизована властивість **float** зі значеннями **left** і **right**, яка може бути застосована не тільки до зображень, але й до інших елементів **HTML**. У комбінації із властивістю **float** часто застосовують властивість **clear** зі значеннями **left** і **right** або **both** для того, щоб звільнити місце ліворуч або (і) праворуч від інших елементів Web-сторінки.

У прикладі 16_1 показано використання плаваючої моделі для зображення.

Приклад 16_1

```
<html><head> <title>Приклад 16_1</title>
  <style>
    .leftfloat {float:left}
    .rightfloat {float:right}
  </style>
</head>
<body>
<strong>Приклад 16_1</strong><br>
<img src=«метелик.jpg» class=«leftfloat» alt=«flax» width= «80px»
height=«80px» border=«1px»>
```

<p> Для обтікання текстом зображення ліворуч замість атрибута align тегу img використовується властивість float зі значенням left. Для обтікання текстом зображення ліворуч атрибута align тегу img використовується властивість float зі значенням left. Для обтікання текстом зображення ліворуч замість атрибута align тегу img використовується властивість float зі значенням left. </p>

```
<img src=«метелик.jpg» class=«rightfloat» alt=«flax» width=«80px» height=«80px» border=«1px»>
```

<p > Для обтікання текстом зображення праворуч замість атрибута align тегу img використовується властивість float зі значенням right. Для обтікання текстом зображення праворуч атрибута align тегу img використовується властивість float зі значенням right. Для обтікання текстом зображення праворуч атрибута align тегу img використовується властивість float зі значенням right.</p>

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

На рис. 13 показано відображення тексту прикладу 16_1 у браузері.



Рис. 13. Відображення тексту у браузері прикладу 16_1

У прикладі 16_2 показано використання плаваючої моделі для абзацу із цитатою.

Приклад 16_2

```
<html><head><title>Приклад 16_2</title>
```

```

<style>
p {font-family:Garamond;
text-align:justify;
margin-left:0.25em;}
.clearp {clear:left;}
p.Floatl {float:left;
width:45%;
text-align:center;
margin:0.5em 0.25em;
padding:0.25em;
border-top:1.3em solid #999 ;
border-bottom:1.3em solid #999 ;
background-color:black;
color:white;
font-size:1.3em;
font-family:Garamond;
}
</style>
</head>
<body>
<strong>Приклад 16_2</strong>
<p>У прикладі використовується обтікання текстом абзацу цитати. Властивості цитати записана в класі .Floatl</p>
<p class=«Floatl»>«Фантазія важливіше знань» А.Ейнштейн </p>
    <p>Обтікання цитати, укладеної в абзац, забезпечується властивістю float. Для елемента обтікання повинна бути обов'язково задана його ширина, крім того, в прикладі зазначені властивості margin і padding, рамки сірого кольору зверху і знизу, а також властивості шрифту: сімейство, колір символів, колір фону і розмір. Обтікання цитати, укладеної в абзац, забезпечується властивістю float. Для елемента обтікання повинна бути обов'язково задана його ширина, крім того, в прикладі зазначені властивості margin і padding, рамки сірого кольору зверху і знизу, а також властивості шрифту: сімейство, колір символів, колір фону і розмір</p>
</body></html>

```

На рис. 14 показано відображення тексту прикладу 16_2 у браузері.

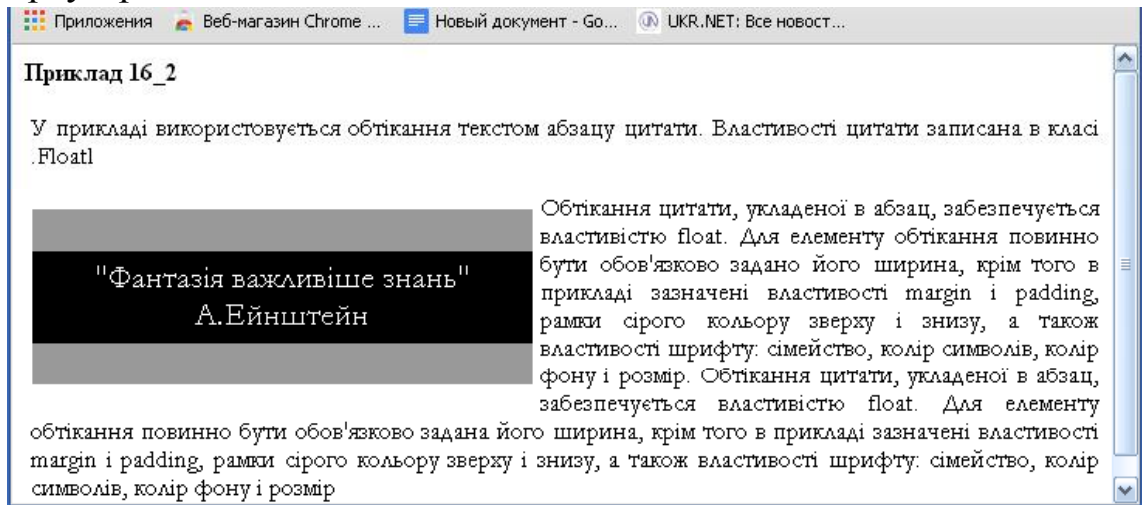


Рис. 14. Відображення тексту у браузері прикладу 16_2

У прикладі 16_3 показано використання обтікання для створення триколонного документа із вставленою в одному зі стовпчиків цитатою.

Приклад 16_3

```
<html><head><title>Приклад 16_3</title>
<style type=«text/css»>
#header {color: white;
background-color: #666;
border-bottom:1px solid #333;
text-align: center;
padding: 0.1em 0;
margin: 0 0 1em 0}

#leftcol {width: 25%;
float:left;
background: white;
padding-bottom: 1em;
text-align: left;
}

#rightcol {width: 25%;
```

```

float:left;
background: white;
padding-bottom: 1em;
text-align: left;
}

#centercol {
width:43%;
float: left;
background: white;
padding: 0 1em;
border-left: 1px solid #333;
border-right: 1px solid #333
}

#cite {
float:right;
width:90%;
border: 1px solid red;
padding:0.1em;
margin: 0 0.1em;
font-size:larger;
text-align:center;
}

#footer {
clear: both;
padding-bottom: 1em;
border-top: 1px solid #333;
text-align: center;}
</style>
</head>
<body>

```

```

<div id=«header»>
<h3>Технологія каскадних таблиць стилів (Cascading Style Sheets,
CSS)</h3>
</div>
<div id=«leftcol»>

```

<p>Таблиці стилів – це набір елементів оформлення, які застосовуються до різних частин документа і описують способи їх представлення на екрані. Таблиці стилів реалізовані у всіх функціонально-розвинених текстових процесорах</p></div>

<div id=«centercol»>

<p>Прийняття в 1996 році Консорціумом W3C CSS першого рівня як стандарту дозволило відокремити зміст Web-сторінки (текст, графічні зображення тощо) від її оформлення (макет сторінки і характеристики тексту, наприклад, шрифти, колірне оформлення тощо). Після цього мова HTML знову стала функціонально-орієнтованою, а не орієнтованою на форму. Стандарт CSS2, прийнятий в 1998 році, заснований на CSS першого рівня дозволяє розробникам здійснювати контроль над Web-сторінками на більш високому рівні. Він включає деякі нові функції, зокрема, можливість точно розташовувати елементи й об'єкти Web-сторінки, застосовувати шрифти, що завантажуються, або використовувати звукові таблиці стилів. Застосування стилів дозволяє також розв'язати багато проблем підтримки браузерів, тому що основні компанії-розроблювачі браузерів вмонтували таблиці CSS у свої програмні продукти.

</p></div>

<div id=«rightcol»>

<p> За допомогою таблиць стилів можна формувати текст, використовуючи методи, наближені до методів форматування звичайних друкованих сторінок. Створені сторінки будуть коректно функціонувати, враховуючи деякі обмеження, пов'язані із платформами, браузерами, різними розмірами екранів і дозволами моніторів. Процес розробки Web-стандартів незабаром обіцяє розв'язати ці проблеми</p>

<p id=«cite»><«Мистецтво – це відображення світу в образах, а наука – відображення світу в поняттях» Н. К. </p></div>

<div id=«footer»>Консорціум W3C CSS</div>

</body></html>

На рис. 15 показано відображення тексту прикладу 16_3 у браузері.

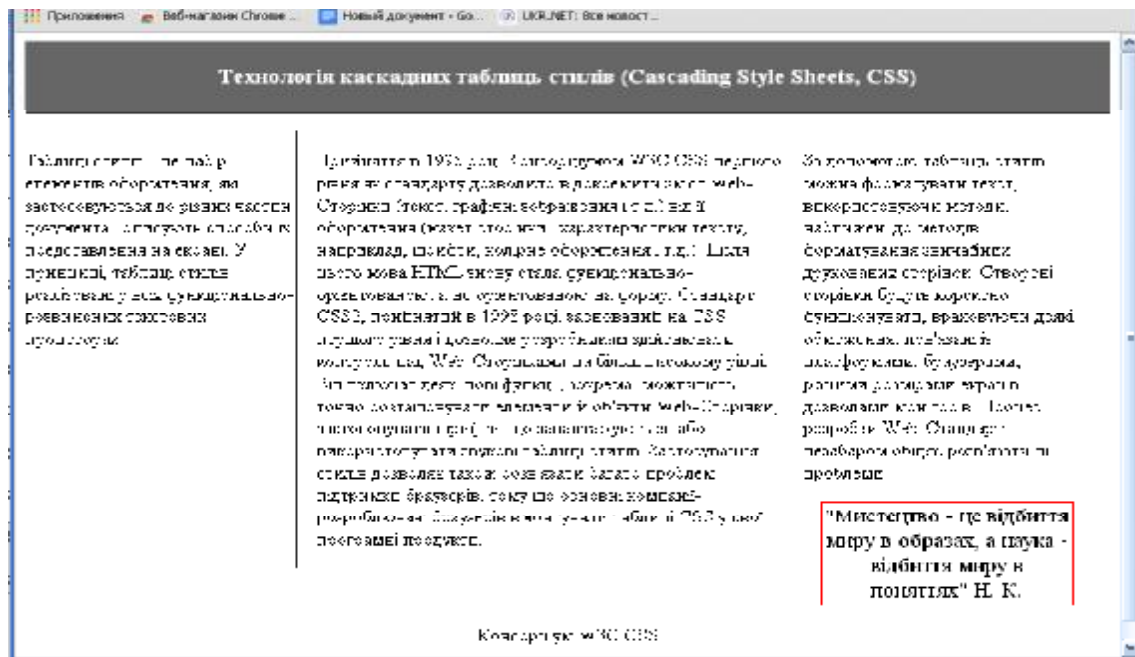


Рис. 15. Відображення тексту у браузері прикладу 16_3

Позиціонування й візуалізація

Застосування позиціонування припускає використання ряду понять:

Обмежена область (блок-контейнер) – невидима прямокутна область, що визначена браузером. Таблиці стилів дозволяють управляти цією областю, установлюючи її положення на сторінці з використанням абсолютних або відносних значень позиціонування.

Абсолютне позиціонування – технологія, що дозволяє задавати координати блоку щодо його блок-контейнера, яким може бути інший (батьківський) елемент документа або початковий блок-контейнер – прямокутник, відповідно до вікна перегляду браузера. При цьому блок елемента повністю видаляється з потоку документа й позиціонується щодо його блок-контейнера.

Відносне позиціонування – технологія, що дозволяє задавати координати блоку щодо його незміщеного положення в потоці документа. При відносному позиціонуванні елемент зрушується зі свого звичайного місця, але простір, який він повинен був займати, не зникає.

Фіксоване позиціонування – технологія, що дозволяє задавати координати блоку щодо початкового блоку-контейнера (вікна перегляду браузера).

1. **position** – метод позиціонування блоку за замовчуванням має значення **static**, іншими значеннями є **absolute** (абсолютне) і **relative** (відносне) позиціонування, а також значення **fixed**, при якому позиціонування блоку є зсувом, як у випадку абсолютного позиціонування, але блок фіксується у вікні браузера й не переміщується при прокручуванні вікна.

2. **top** – величина зсуву вниз від верху його блок-контейнера, **bottom** – нагору від нижньої сторони блок-контейнера, **left** – відповідно, уліво, а **right** – вправо від лівої й правої сторін блок-контейнера. Значеннями є будь-які відповідні до стандарту довжини, а також процентне значення: відношення у відсотках довжини зсуву до ширини (висоти) блоку, а також припустимі негативні значення зазначених властивостей зсуву.

3. **width** – ширина блоку. Значеннями є будь-які відповідні до стандарту довжини, а також процентне значення: відношення у відсотках довжини зсуву до ширини вікна.

4. **height** – висота блоку. Значеннями є будь-які відповідні до стандарту довжини, а також процентне значення: відношення у відсотках довжини зсуву до висоти вікна.

5. **z-index** – **z-індекс** визначає порядок розташування блоків. Значеннями є цілі числа (позитивні й негативні), причому блоки з більшими значеннями z-індексу будуть з'являтися над блоками з меншими значеннями.

6. **visibility** – видимість. Визначає, чи є елемент видимим – **visible** або прихованим – **hidden**.

7. **overflow** – керування переповненням. Має три значення: **visible** (елемент видний), **hidden** (частина, що перекривається, відтинається), **scroll** (використовується механізм прокручування для візуалізації елемента).

8. **clip: rect (top right bottom left)** – відсікання. Визначає області, що вирізаються.

9. **display** – подання елемента зі значеннями **none**, **inline**, **block**, **list-item**, **table**, **inline-table**.

Приклад 17_1 ілюструє основні варіанти позиціонування

```
<html><head><title>Приклад 17_1</title>
<style type=«text/css»>
<!--
p, h5 {margin:0;padding:0}
-->
</style>
</head>
<body>
<strong>Приклад 17_1</strong>
<div style=«background-
color:gray;position:absolute;top:70;left:50;color:blue;»>
<h5>relative</h5>
<p>Екб екб екб екб</p>
<p style=«position:relative;top:70;left:50;color:red»>
Екб екб екб екб</p>
</div>
<div style=«background-
color:gray;position:absolute;top:70;left:250;color:blue»>
<h5>normal stream</h5>
<p>Екб екб екб екб</p>
<p style=«color:red»>Екб екб екб екб</p>
</div>

<div style=«background-color:gray;position: absolute;top: 70;left:450;
color:blue»;>
<h5>absolute</h5>
<p>Екб екб екб екб</p>
<p style=«position:absolute;top:70;left:50;color:red»>Екб екб екб
екб</p>
</div>
</body></html>
```

На рис. 16 показано відображення тексту прикладу 17_1 у браузері.

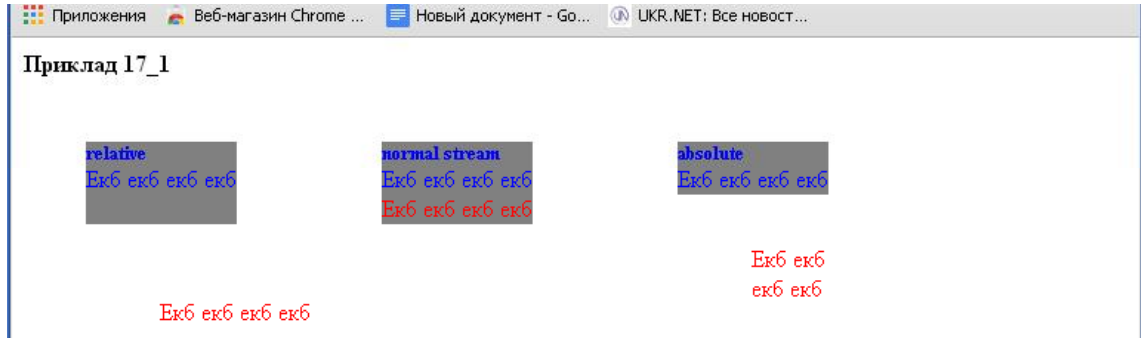


Рис. 16. Відображення тексту у браузері прикладу 17_1

У випадку відносного позиціонування текст червоного кольору на **top:70** і **left:50** від свого незміщеного положення в потоці елементів, а у випадку абсолютного – від батьківського елемента, яким є прошарок **div**.

У прикладі 17_2 показано використання **абсолютного** позиціонування для відображення двох прошарів. Вертикальний прошарок кольору **deeppink** містить логотип деякого ЗАТ, який позиціонований щодо його нормального положення в потоці елементів сторінки, а також текст «ЗАТ Повітряна куля», що має оформлення, описане в класі «**revers**».

Горизонтальний прошарок кольору **mistyrose** розташований поверх вертикального й містить текст, оформлений згідно зі стилем **bluetext**.

Приклад 17_2

```
<html><head><title>Приклад 17_2</title>
<style type=«text/css»>
.revers {
font-weight:bold;
color:white;
text-align: center;}
.bluetext {
font-weight:bold;
color:darkblue;
text-align: justify;
padding:0.5em;
```

```

margin:0}
</style>
</head>

<body>
<strong>Приклад 17_2</strong>
<!--вертикальний шар-->
<div style=«position: absolute; top:0;left: 250;width:200;
height:300;background:gray»>
<img src=«метелик1.jpg» style=«position:relative; top:20;
left:25;» >
<p class=«revers»>МЕТЕЛИК</p>
</div>
<div style=«position: absolute; top:210;
left:25;width:400;
height:90;background :white»>
    <p class=«bluetext»>Метелик з абзацем лежить поверх пра-
вого стовпчика (шару кольору deerpink) з логотипом. Зображення
логотипу позиціоноване відносно (relative) його нормального по-
ложення в потоці елементів сторінки</p>
</div>
</body>
</html>

```

На рис. 17 показано відображення тексту прикладу 17_2 у браузері.

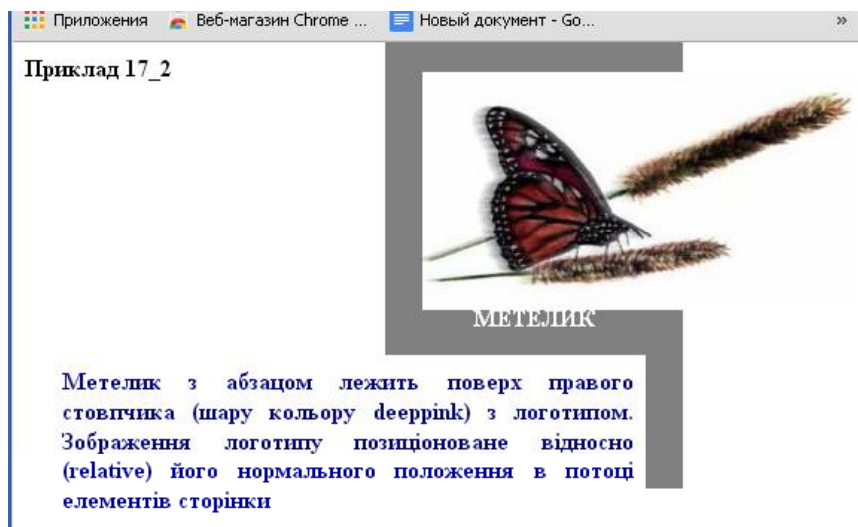


Рис. 17. Відображення тексту у браузері прикладу 17_2

Приклад 17_3 демонструє використання властивості **z-index**, яке дозволяє задати порядок (рівень) розташування елемента. Елемент **<div>** використовується для визначення властивостей прошарку. Він є контейнерним елементом і дозволяє застосувати позиціонування й **z-index** для одного або декількох елементів, що вміщені в ньому.

У прикладі 17_3 на Web-сторінці виводиться текст і прошарок **<div>** сірого кольору.

Приклад 17_3

```
<html><head><title>Приклад 17_3</title>
<style type=«text/css»>
<!--
.div_layer {
background-color:#c0c0c0;
position:absolute;left:80;
width:150; height:100;
visibility:visible;}
div_text {
color:blue;
position:absolute;left:50;
width:350; height:100;
font-size:24pt;}
-->
</style>
</head>

<body>
<strong>Приклад17_3</strong>
<div class=«div_layer» style=«top:50;z-index:1;»>
</div>
<div class=«div_text» style=«top:50;»> z-індекс шару більше,
ніж тексту. Шар розташований «ближче» до користувача.</div>

<div class=«div_layer» style=«top:200;z-index:-1;»>
</div>
```

```
<div class=«div_text» style=«top:200;»> z-індекс тексту біль-  
ше, ніж шару. Текст розташований «ближче» до користува-  
ча.</div>  
</body></html>
```

На рис. 18 показано відображення тексту прикладу 17_3 у браузері.

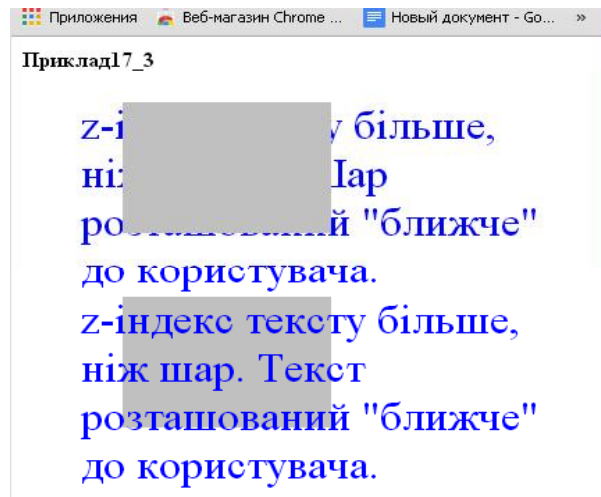


Рис. 18. Відображення тексту у браузері прикладу 17_3

Властивості прошарку сірого кольору й прошарку тексту описані за допомогою відповідних класів у розділі опису стилів (`<style>`). У верхній частині прикладу **z-index** сірого прошарку позитивний (рівний 1), тоді як текст має за замовчуванням **z-index** рівний 0. У результаті прошарок тексту перебуває на задньому плані, а сірий прошарок – на передньому плані.

У нижній частині прикладу **z-index** сірого прошарку негативний (рівний -1), а текст лежить у нульовому прошарку. При цьому текст виявляється на передньому плані, а сірий прошарок – за текстом.

У прикладі 18 демонструється ефект, завдяки якому можна управляти поданням інформації у межах обмеженої області. Обмежена область менша, ніж зображення, а переповнення сховано. Таким чином, зображення вписується в обмежену область. Область із переповненням розміщена нижче звичайного зображення, яке оточено тонкою рамкою, а область із переповненням – рамкою з більшою товщиною.

Приклад 18

```
<html>
<html><head><title>Приклад 18</title>
<style type=«text/css»>
<!--
.overflow {
position: absolute; top:190;left: 50;
width:80;
height:80;
border:2px solid black;
overflow:hidden;
}
-->
</style>
</head>
<body>
<strong>Приклад18</strong><br>
<img src=«roza.jpg» width=150 height=150 alt= «flower» border=1>
<img class=«overflow» src=«roza.jpg» width=150 height=150 alt=
«flower»>
</body>
</html>
```

На рис. 19 показано відображення тексту прикладу 18 у браузері.



Рис. 19. Відображення тексту у браузері прикладу 18

У прикладі 19 використаний клас **clip1**, який описує текст, поміщений у квадратну область, описану дескриптором **<div>**. Клас **clip** забезпечує застосування до цієї області відсікання (зверху й ліворуч на 25 пікселів, а знизу й праворуч на 125 пікселів). На рис. 20 текст у квадратній області розташований праворуч, а відсічений – ліворуч.

Приклад 19

```
<html><head><title>Приклад 19</title>
<style type=«text/css»>
<!--
.clip {
position: absolute; top:50;
left: 230;
width:150;
height:150;
color:yellow;
background-color:black;
clip:rect(25px 125px 125px 25px);}
.clip1 {
position: absolute; top:50;
left: 30;
width:150;
height:150;
color:yellow;
background-color:black;
}
-->
</style>
</head>
<body>
<strong>Приклад19</strong>
<div class=«clip1»>Текст абзацу буде обрізаний за допомогою відсікання, застосованого до даної квадратної області розміром 150 на 150 пікселів.</div>
```



```
<div class=«clip»>Текст абзацу буде обрізаний за допомогою відсікання, застосованого до даної квадратної області розміром 150 на 150 пікселів.</div>
```

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

На рис. 20 показано відображення тексту прикладу 19 у браузері.

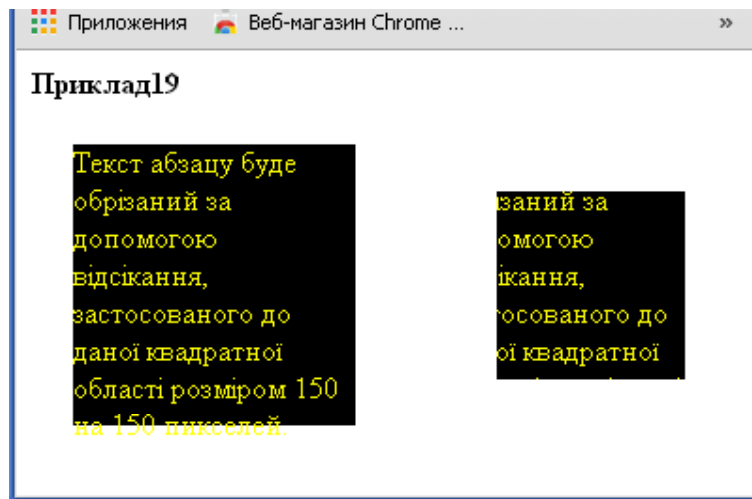


Рис. 20. Відображення тексту у браузері прикладу 19

У прикладі 19_1 показано застосування властивості **display** для створення горизонтального меню зі списку посилань. *Блоковий* елемент, яким є список посилань, перетворений у *рядковий* з оформленням, відповідним до елементів меню.

Приклад 19_1

```
<html><head><title>Пример 19_1</title>
<style>
#nav h5 {text-align:center;font-size:16pt}
#nav ul {padding: 3px 0;
border-bottom:1px solid #778;
font :bold 12px Verdana, sans-serif;
}

#nav ul li {
display: inline;
}
```

```
#nav ul li a {
padding: 3px 0.5em;
margin-left: 3px;
border: 1px solid #778;
border-bottom: none;
background: #dde;
text-decoration: none;
}
```

```
#nav ul li a:link {
color: #0000ff;
}
```

```
#nav ul li a:visited {
color: #667;
}
```

```
#nav ul li a:link:hover, #nav ul li a:visited:hover {
color: #000;
background: #aae;
border-color: #227;}
}
```

```
#nav ul li a#current {
background: white;
border-bottom: 1px solid white;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div id=«nav»>
```

```
<h5> navigation bar </h5>
```

```
<ul>
```

```
<li><a href=«bg1.html»>home</a></li>
```

```
<li><a href=«bg1.html»>about</a></li>
```

```
<li><a href=«bg1.html»>archives</a></li>
```

```
<li><a href=«bg1.html» id=«current»>contact</a></li>
```

```
</ul>
```

</div>

</body></html>

На рис. 21 показано відображення тексту прикладу 19_1 у браузері.



Рис. 21. Відображення тексту у браузері прикладу 19_1

Фільтри

Є два типи спеціальних ефектів для зміни виду тексту й графіки, які називаються **Visual Filter** (візуальні або статичні фільтри) і **Transitions Filter** (динамічні фільтри).

Ефекти фільтрів – від дзеркального відображення тексту й графіки до зникнення їх на шаховій дошці.

Статичні фільтри

Статичні фільтри можуть бути застосовані до тем елементів **HTML**, які розглядаються як керуючі. Це елементи, що створюють на Web-сторінці прямокутну область. До таких елементів відносять: **body, div, img, span, table, tr, td, input**.

Статичні фільтри міняють зовнішній вигляд об'єкта, і ця картина залишається нерухливою. За властивістю стилю **filter** ставиться двокрапка, потім – ім'я фільтра, за яким у круглих дужках перелічуються атрибути з їхніми значеннями, розділені комами. Відмінність у кодуванні різних фільтрів полягає у назві фільтра, а також у кількості й типі використовуваних атрибутів. У таблиці 2 наведений список статичних фільтрів, їх атрибути й функції.

Таблиця 2

Фільтр	Необхідні атрибути	Дія
alpha	opacity (0-100%), finish opacity, style (0,1 –зміна прозорості уздовж лінії, 2 – радіально від центру), startx, starty, finishx, finishy (координати крапок для style=1)	Задає рівень прозорості
Blur	add(=1), direction (0 – 360 – напрямокразмытия), strength (величина зсуву)	Створює ефект руху
chroma	color	Створює одноколірний транспарант
dropshadow	color (колір тіні), offx (зсув по осі x), offy (те ж по y), positive (0–1, тіні немає або є)	Створює тінь (на відстані)
flipH		Робить горизонтальне відображення об'єкта
flipV		Робить вертикальне відображення об'єкта
glow	color, strength	Створює ефект сяйва
Grayscale		Перетворить об'єкт у чорно-білий
Invert		Інвертує кольори, відтінки і яскравість
Light		Висвітлює об'єкт від джерела
Mask	color	Робить із об'єкта маску-транспарант
Shadow	color, direction	Створює тінь
Wave	add, freq (число максимумів), lightstrength, phase, strength	Створює хвилясте викривлення об'єкта
Xray		Показує контур об'єкта

У прикладі 20 наведений приклад дії на текст «сяючого» фільтра й фільтра горизонтального відображення.

Приклад 20

```
<html><head ><title>Приклад 20</title><style type=«text/css»>
<!--
.effect_glow { filter:glow(color=gray,strength=5);}

```

```

.effect_flip { filter:flipv;}
-->
</style>
</head>
<body>
<strong>Приклад 20</strong><br><br>
<div class=«effect_glow»
style=«position:absolute;width:200;height:100;top:50;left:100;font-
family:verdana;font-style:bold;font-size:20;text-align:center;»>
Текст із ефектом «сяючого» фільтра!</div>
<div class=«effect_flip»
style=«position:absolute;width:250;height:150;top:130;left:70;font-
family:verdana;font-style:bold;text-align:center;»>
Текст і графіка з ефектом вертикального відображення.
<img src=«roza.jpg» width=60 height=70 alt=«roza»>
</div>
</body>
</html>

```

На рис. 22 показано відображення тексту прикладу 20 у браузері.

Приклад 20



Рис. 22. Відображення тексту у браузері прикладу 20

Динамічні фільтри

Динамічні фільтри дозволяють спостерігати плавний перехід від одного стану об'єкта до іншого зі швидкістю, що задається розроблювачем.

Для установки динамічного фільтра використовується дескриптор **meta**. За допомогою цього дескриптора можна управляти завантаженням сторінки в браузер і її відходом з екрана.

При вході на сторінку застосовується ефект «кола, що розширюється» (номер фільтра-3), а при виході зі сторінки – «шаховий порядок униз» (номер фільтра – 11). Html-код сторінки прикладу 21.1 містить гіперпосилання (на сторінку прикладу 21.2) і при активізації цього гіперпосилання відбувається вихід зі сторінки й спостерігається ефект – «шаховий порядок униз», пов'язаний з виходом зі сторінки. При завантаженні сторінки або поверненні на неї за допомогою кнопки браузера «назад» спостерігається вхідний ефект - «коло, що розширюється».

Для перегляду всіх ефектів можна встановити номер фільтра, рівний 23, що відповідає випадковому вибору типу фільтра.

Приклад 21.1

```
<html>
<head><title>Приклад 21.1</title>
<meta http-equiv=«Page-Enter»
content=«Revealtrans(Duration=6,Transition=3)»>
<meta http-equiv=«Page-Exit»
content=«Revealtrans(Duration=6,Transition=11)»>
</head>
<body bgcolor=#d3d3d3 text=yellow>
<strong>Приклад 21.1</strong><br><br>
<a href=«pr21_2.html»> Перехід до прикладу 21.2</a>
</body>
</html>
```

Приклад 21.2

```
<html>
<head><title>Приклад 21.2</title>
```

```

</head>
<body bgcolor=darkcyan text=yellow>
<strong>Приклад 21.2</strong><br><br>
<a href=«pr21_1.html»> Перехід до прикладу 21.1</a>
</body>

```

</html> У таблиці 3 наведені номери фільтрів і ефекти переходу, відповідні до цих фільтрів.

Таблиця 3

Ефект переходу	Номер фільтра	Ефект переходу	Номер фільтра
Прямокутник, що стягається	0	Випадковий наплив	12
Прямокутник, що розширюється	1	Вертикальний розподіл усередину	13
Коло, що стягається	2	Вертикальний розподіл назовні	14
Коло, що розширюється	3	Горизонтальний розподіл усередину	15
Стирання нагору	4	Горизонтальний розподіл назовні	16
Стирання вниз	5	Стирання вліво вниз	17
Стирання праворуч	6	Стирання вліво нагору	18
Стирання ліворуч	7	Стирання вправо вниз	19
Вертикальні жалюзі	8	Стирання вправо нагору	20
Горизонтальні жалюзі	9	Випадкові горизонтальні лінії	21
Шаховий порядок поперечний	10	Випадкові вертикальні лінії	22
Шаховий порядок униз	11	Випадковий вибір з попередніх	23

На рис. 23 показаний ефект, що виникає при завантаженні сторінки прикладу 21.1 у браузер.

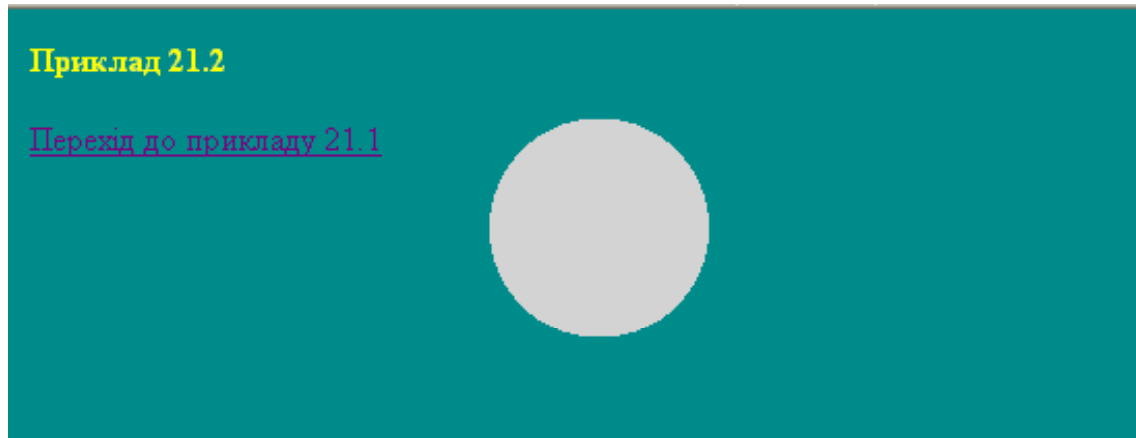


Рис. 23. Ефект, що виникає при завантаженні сторінки прикладу 21.1 у браузер

На рис. 24 показаний ефект, що виникає при виході зі сторінки прикладу 21.1 за допомогою гіперпосилання на сторінку прикладу 21.2.

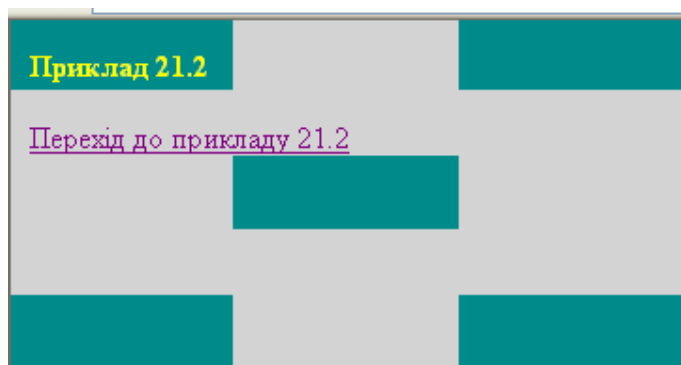


Рис. 24. Ефект, що виникає при виході зі сторінки прикладу 21.1 за допомогою гіперпосилання на сторінку прикладу 21.2