

Лекційне заняття №2

з навчальної дисципліни WEB-Дизайн

на тему:

Основи HTML

Частина 1

План заняття

1. Коротка характеристика мови HTML
2. Форматування тексту
3. Теги заголовків
4. Форматування HTML-документу
5. Коментарі та спеціальні символи
6. Списки
7. Гіперпосилання
8. Рисунки

Коротка характеристика мови HTML

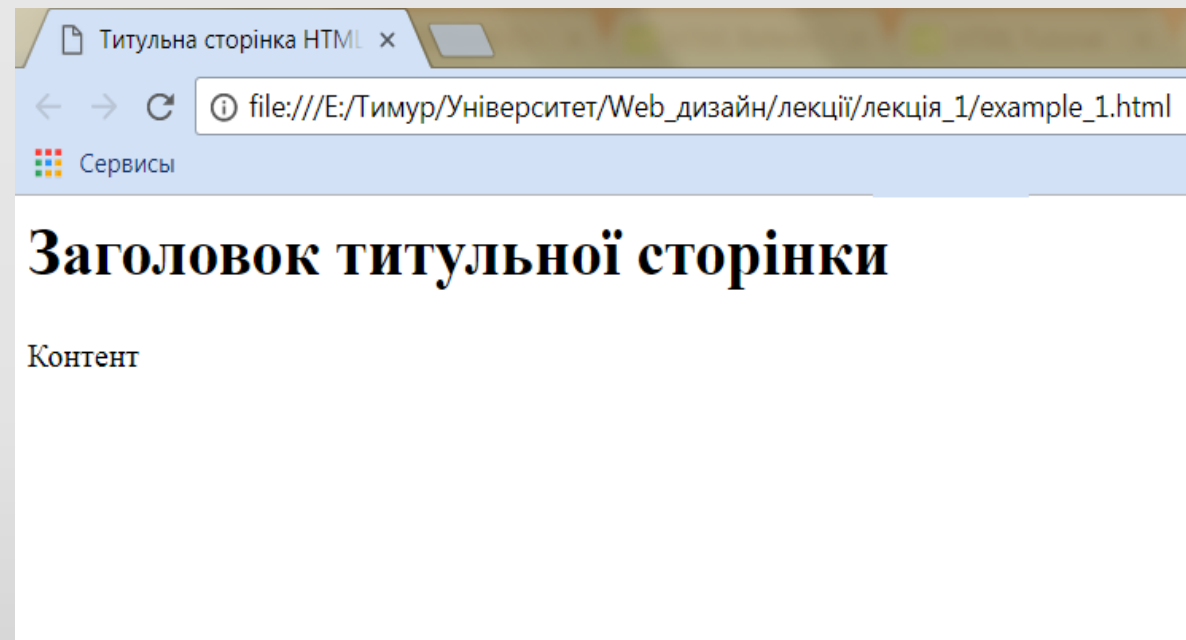
```
<!DOCTYPE html>
<html>
  <head>
    <title>Титульна сторінка HTML-документа
  </title>
  </head>
  <body>

    <h1> Заголовок титульної сторінки </h1>

    <p> Контент </p>

  </body>
</html>
```

Код HTML-документа



Приклад ВЕБ-сторінки в браузері

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv=Content-Type content="text/html;
charset=windows-1251" >
    <title>Титульна сторінка HTML-документа </title>
  </head>
</body>
```

Контент

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

Першим тегом, з якого слід починати опис HTML-документу, є тег `<html>`. Він завжди повинен починати опис документу, а завершати опис повинен тег `</html>`. Тобто тег `<html>` є тегом контейнером для всієї HTML-сторінки. Сам документ є звичайним текстовим ASCII-файлом і може складатися із двох розділів - заголовка та змістовної частини.

Розділ заголовка відкривається тегом **`<head>`** і закривається тегом **`</head>`**. Задачею заголовку є надання браузеру інформації, необхідної для коректного відображення HTML-сторінки. Теги, що знаходяться всередині розділу заголовку (крім назви документу), на екрані не відображаються. Назву документу визначають за допомогою теги контейнеру **`<title>`**.

Змістова частина документу, що не містить фреймів, визначається тегом контейнером **`<body>`**. Тобто весь текст та всі теги змістовної частини документу знаходиться між тегами `<body>` та `</body>`.

Тег meta

- Визначає кодування тексту HTML-сторінки

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

- Визначає ключові слова, які використовуються пошуковими системами

```
<meta name="keywords" content="економіка, торгівля">
```

- Визначення опису HTML-сторінки в пошуковій системі.

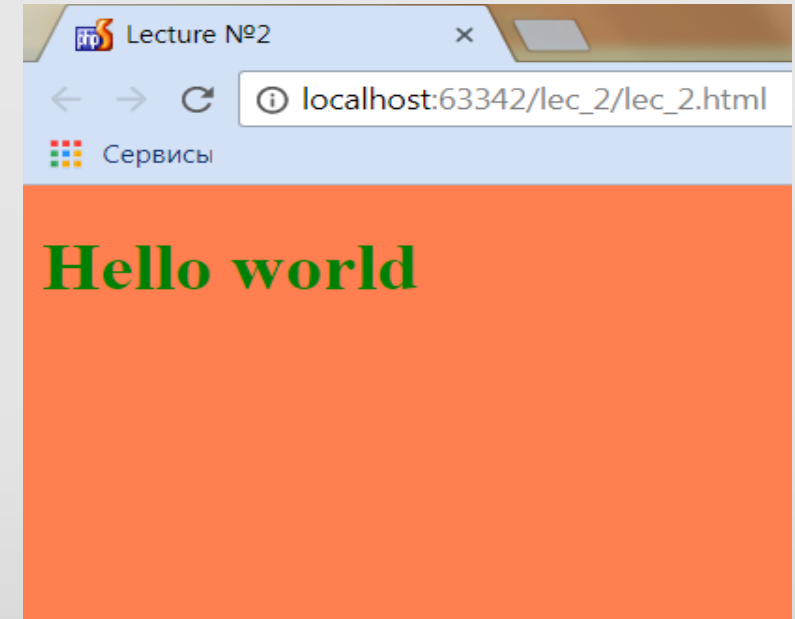
```
<meta name="description" content = "сайт, присвячений економіці та торгівлі">
```

- Відображення в браузері визначеної HTML-сторінки

```
<meta http-equiv="refresh" content="20;URL=untitled1.html">
```

Параметри тега <body>

Параметр	Призначення
alink	Визначає колір активного гіперпосилання
background	Вказує на адресу фонового зображення
bottommargin	Визначає межу нижнього поля документу
bgcolor	Визначає колір фону документу
bgproperties	Дозволяє або забороняє прокрутку фонового зображення
leftmargin	Визначає межу лівого поля документу
link	Визначає колір не переглянутого гіперпосилання
rightmargin	Визначає межу правого поля документу
scroll	Встановлює або забороняє смуги прокрутки вікна браузера
text	Визначає колір тексту
topmargin	Визначає межу верхнього поля документу
vlink	Визначає колір переглянутого гіперпосилання



```
<!DOCTYPE html>
<html>
<head lang="en">
  <meta charset="UTF-8">
  <title>Lecture №2</title>
</head>
<body text="green" bgcolor="#ff7f50">

  <h1>Hello world</h1>

</body>
</html>
```

Форматування тексту

Форматування тексту засобами HTML реалізується за допомогою спеціальних тегів. Всі теги форматування тексту використовуються тільки в розділі `<body>` і є парними тегами. Найчастіше використовуються такі теги:

Тег `` - призначений для відображення тексту напівжирним шрифтом. Наприклад:

`Це напівжирний шрифт`

Тег `<i>` - призначений для відображення тексту курсивом. Наприклад:

`<i>Це курсив</i>`

Тег `<u>` - призначений для відображення тексту підкресленим. Наприклад:

`<u>Це підкреслений текст</u>`

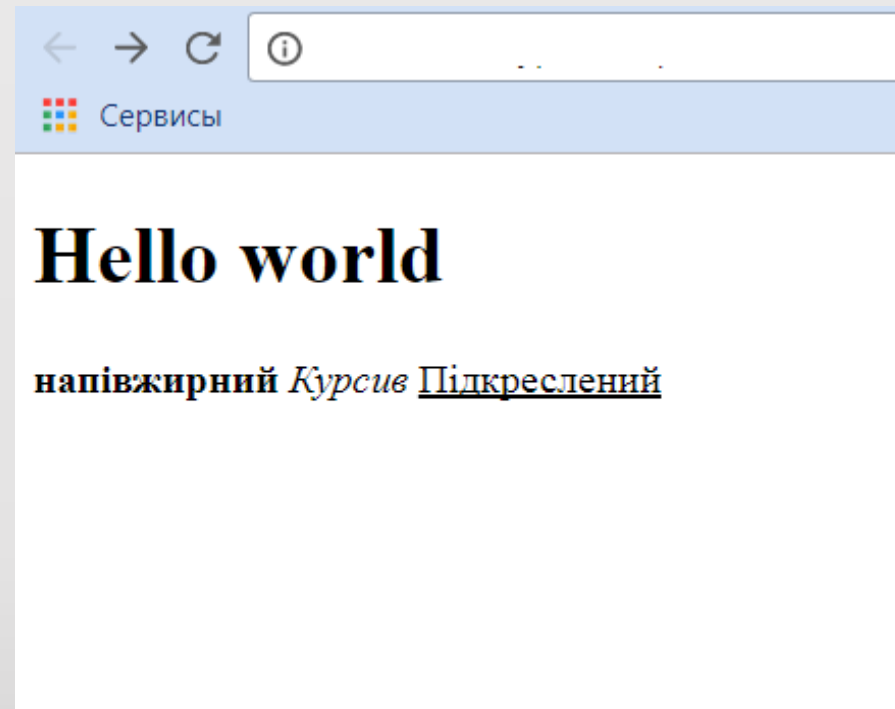


Рис. 2.2 Використання тегів форматування тексту

Тег **<s>** - призначений для відображення тексту перекресленими символами.

Наприклад:

<s>Це перекреслені символи</s>

Тег **<sub>** - призначений для відображення тексту в вигляді верхнього індексу.

Наприклад:

_{Верхній індекс}

Тег **<sup>** - призначений для відображення тексту в вигляді нижнього індексу.

Наприклад:

^{Нижній індекс}

Теги **<sub>** та **<sup>** зручно використовувати для виводу формул.

Всі теги форматування тексту можуть бути вкладені один в інший. Це дозволяє проводити комбінований вивід тексту. Наприклад, для того, щоб текст був виведений курсивом та був підкресленим необхідно:

<i><u>Приклад підкресленого курсиву</u></i>

Порядок вкладення тегів один в інший значення не має. Наприклад, попередній приклад коректно записати і так:

<u><i>Приклад підкресленого курсиву</i></u>

Зміна зовнішнього вигляду тексту в вікні браузера при використанні наведених тегів форматування показана на рис. 2.2.

Тег **** - призначений для визначення параметрів шрифту. За допомогою цього тегу можливо визначити колір символів, їх розмір та тип шрифту. Зміна параметрів шрифту відбувається за допомогою змін параметрів тегу color, size та face.

Параметр color. Дозволяє змінювати колір символів.

Наприклад, визначення зеленого кольору символів можливо реалізувати так:

Цей текст зеленого кольору

Відзначимо, що колір тексту, визначений за допомогою цього тегу, має пріоритет над кольором, визначеним за допомогою тегу **<body>**. Тобто, якщо в тегу **<body text="red">** визначено, що колір тексту червоний, а в тегу **** заданий жовтий колір, то на всій HTML-сторінці текст буде червоний, але блок, обмежений тегом , буде жовтого кольору.

Параметр face. Дозволяє задати тип шрифту, яким браузер буде виводити текст. Для цього в якості значення параметру необхідно вказати назву шрифту.

Наприклад, вивід тексту шрифтом Courier New, можливо реалізувати так:

** Використовуємо шрифт Courier New**

Відзначимо, якщо на комп'ютері користувача такого шрифту немає, то браузер буде використовувати стандартний шрифт. Крім того можливо задати не одну, а декілька назв шрифтів, розділених комами. Якщо на комп'ютері користувача не буде першого шрифту, то браузер намагатиметься знайти шрифт за номером два і так далі.

Наприклад:

`Використовуємо шрифт Arial, якщо такого нема, то Helvetica і так далі.`

Параметр size. Дозволяє визначити розмір символів. Розмір задається в умовних одиницях від 1 до 7. Фактичний розмір умовних одиниць залежить від характеристик екрану комп'ютера користувача та налаштувань браузера. Цифра один відповідає найменшим символам, а 7 - найбільшим. Розмір символів при стандартних налаштуваннях 3. Розмір символів можливо вказувати як в абсолютних, так і в відносних величинах.

Синтаксис запису наступний:

`Визначення символів в абсолютних величинах`

` Визначення символів в відносних величинах`

В першому випадку x - це цифра від 1 до 7, тобто величина символів в умовних одиницях. В другому випадку x може бути як додатнім, так і від'ємним числом. Причому знак + або – треба вказувати обов'язково. Величина x буде показувати на скільки умовних одиниць повинна змінитись величина символів відносно поточного значення.

Наприклад, вивід тексту з розміром символів 7 та 1 можливо реалізувати так:

`Визначення символів розміром 7`

`Визначення символів розміром 1`

Зменшення символів на одну умовну одиницю відносно поточного значення:

** Зменшення символів на 1 умовну одиницю **

Збільшення символів на одну умовну одиницю відносно поточного значення:

** Збільшення символів на 1 умовну одиницю **

Приклади використання параметрів face та size показані на рис.2.3.

Тег **<basefont>** використовується для визначення параметрів шрифту, які будуть використовуватись в якості стандартних при перегляді HTML-сторінки.

Параметри тегу **<basefont>** ті ж самі, що і для тегу ****. Зміна параметрів **<basefont>** розповсюджується на ту частину HTML-сторінки, яка розміщена нижче тегу **<basefont>**. При сумісному використанні тегу **<basefont>** з **** пріоритет має тег ****. На відміну від **** тег **<basefont>** не має закриваючого тегу. Його дія розповсюджуються до кінця HTML-сторінки або до ще одного тегу **<basefont>**. Наприклад, визначення шрифту розміром 4 за допомогою тегу **<basefont>** можна здійснити так:

<basefont size=4>Використання тегу basefont

Відзначимо, що теги **** та **<basefont>** відносяться до послідовних тегів, а тому не можуть включати в себе теги рівня блоку, наприклад, **<table>** або **<tr>**.

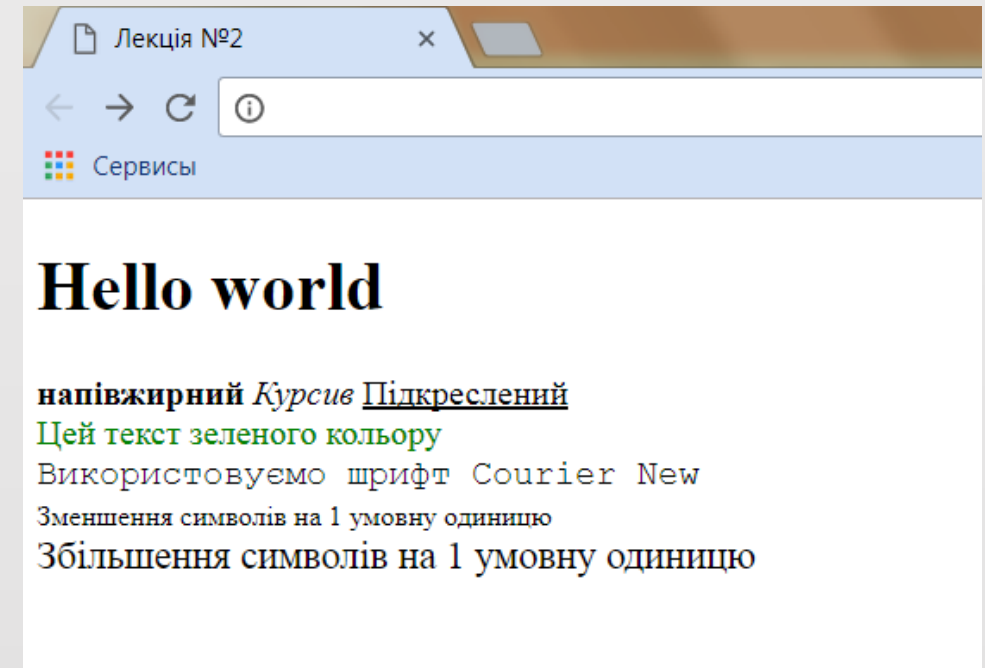


Рис. 2.3 Приклади використання параметрів face та size тегу

Теги заголовків

Теги заголовків (`<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`) призначені для виділення на екрані користувача певних фрагментів тексту HTML-сторінки. Виділення тексту реалізується за рахунок зміни розмірів та "жирності" тексту.

Тегу `h1` відповідає найбільший розмір тексту, а `h6` — найменший розмір. Особливістю даних тегів є вставка порожнього рядка до і після виділеного фрагменту тексту.

Наприклад, для виділення в тексті HTML-сторінки трьох фрагментів тексту як заголовків 1, 3 та 6 рівнів необхідно:

`<body>`

Звичайний текст. `<h1> Заголовок 1 рівня </h1>`

Звичайний текст. `<h3> Заголовок 3 рівня </h3>`

Звичайний текст. `<h6> Заголовок 6 рівня </h6>`

Звичайний текст.

`</body>`

Відповідне відображення HTML-сторінки в вікні браузера показано на рис. 2.4.

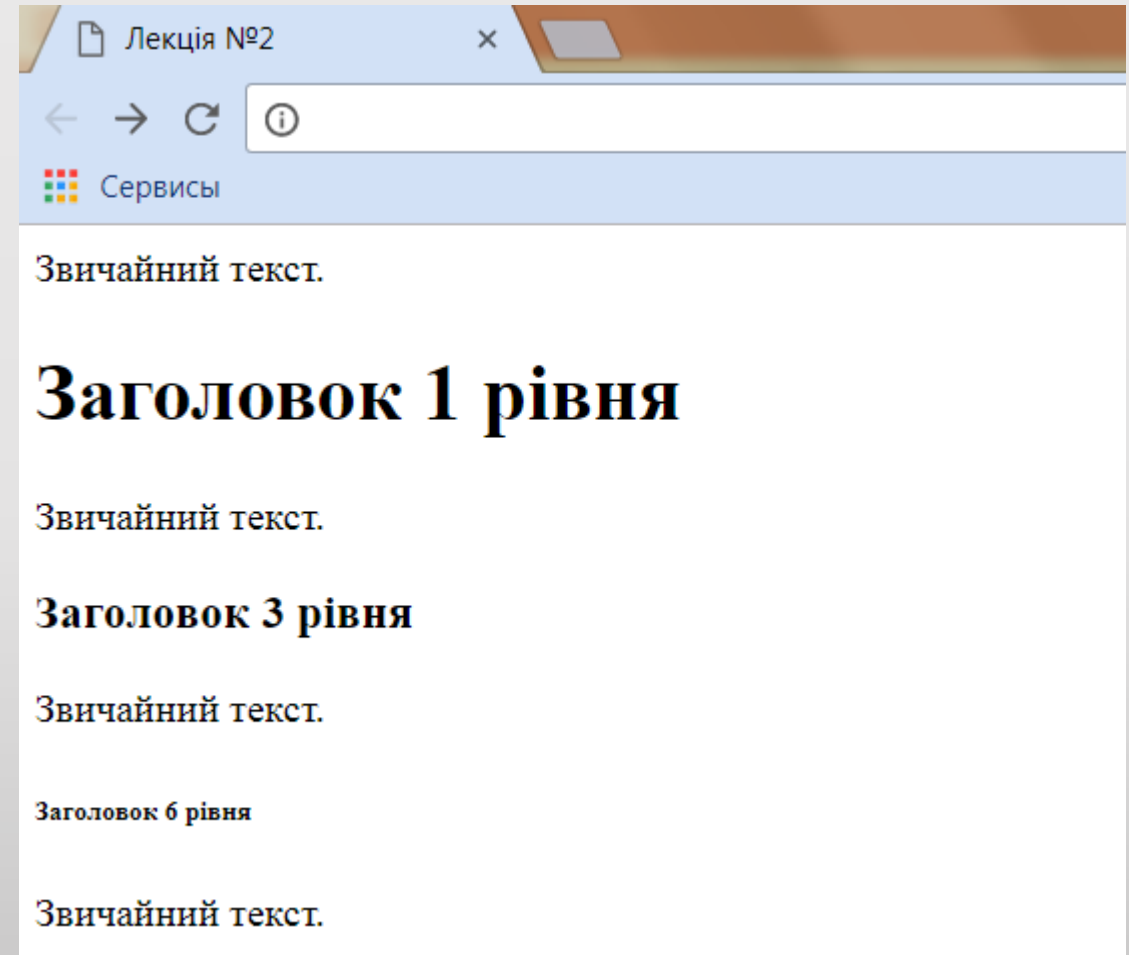


Рис. 2.4. Використання тегів заголовків

Всі вказані теги є парними та можуть використовуватись з необов’язковим параметром align, який служить для горизонтального вирівнювання тексту. Можливі значення параметру align показані в табл. 2.2.

Таблиця 2.2

Значення параметру align

Значення параметру	Наслідки використання
left	Текст вирівнюється по лівій стороні вікна браузера
center	Текст вирівнюється по центру вікна браузера
right	Текст вирівнюється по правій стороні вікна браузера
justify	Текст вирівнюється по ширині вікна браузера. Візуально відрізняється від left тільки в тому випадку, коли текст заголовку містить більш ніж два рядки тексту.

Відзначимо, якщо параметр **align** не заданий, то використовується вирівнювання заголовку по центру вікна браузера.

Розглянемо приклад визначення на HTML-сторінці чотирьох заголовків третього рівня з різними параметрами вирівнювання. Для цього в розділі необхідно записати:

<body> Звичайний текст.

<h3 align="center"> Заголовок вирівнюється по лівій стороні вікна браузера</h3>

Звичайний текст.

<h3 align="center"> Заголовок вирівнюється по центру вікна браузера</h3>

Звичайний текст.

<h3 align="right"> Заголовок вирівнюється по правій стороні вікна браузера</h3>

Звичайний текст.

<h3 align="justify"> Заголовок вирівнюється по ширині вікна браузера.

Візуально відрізняється від left тільки в тому випадку, коли текст заголовку містить більш ніж два рядки тексту. </h3>

Звичайний текст. </body>

Відповідне відображення в вікні браузера показане на рис. 2.5.

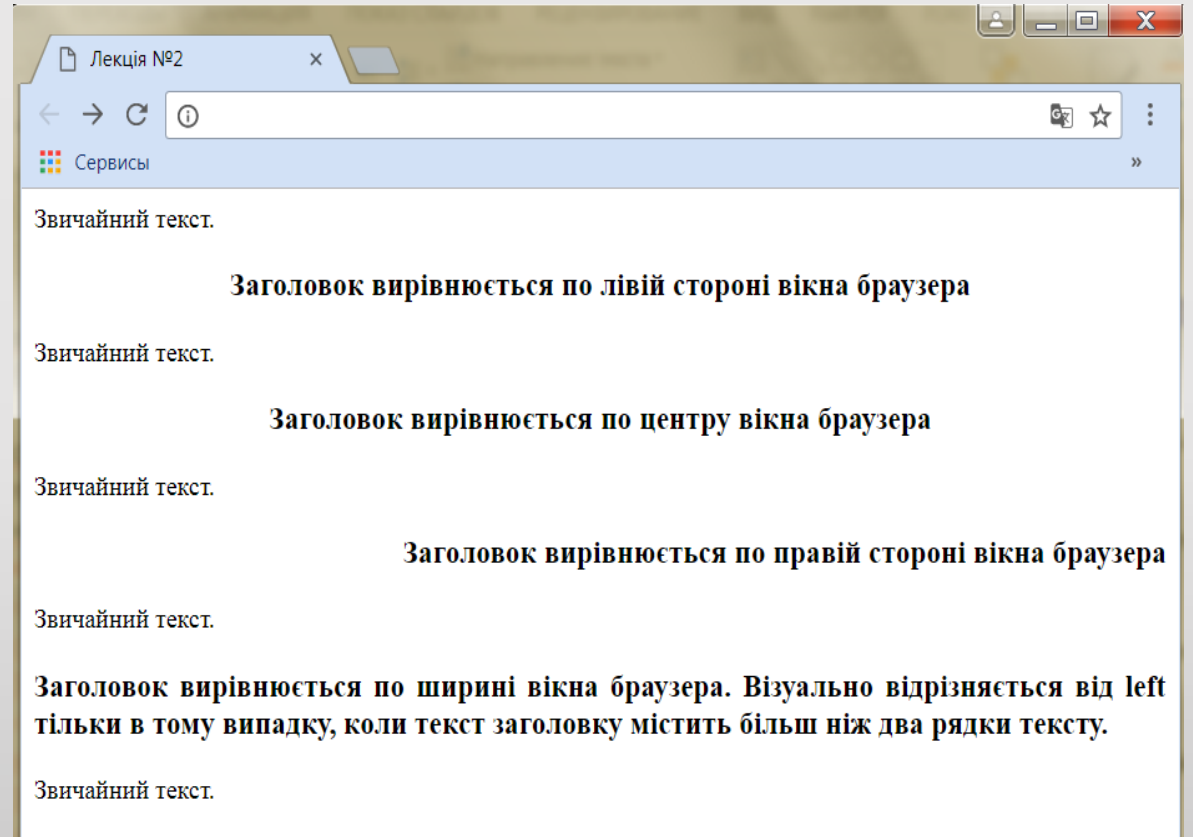


Рис. 2.5 Використання тегу заголовку третього рівня з різними параметрами вирівнювання.

Форматування HTML-документу

Тег `
` використовується для примусового переведу рядка. Необхідність використання даного тегу пояснюється тим, що при відображенні текстових документів перенесення тексту із одного рядка в інший відбувається автоматично та залежить в основному від розмірів шрифту та розмірів вікна браузера. При цьому переведення рядка може здійснюватись тільки по символам, що розділяють окремі слова.

Розглянемо приклад запису виразів "Перший рядок" та "Другий рядок" в двох різних рядках. Відповідний HTML-код такий:

```
<body>  
Перший рядок <br> Другий рядок  
</body>
```

Якщо даний текст не буде вміщуватись в одному рядку, то в вікні браузера з'явиться горизонтальна смуга прокрутки. Тег `<p>` призначений для розділу тексту HTML-сторінки на окремі абзаци. Даний тег є тегом-контейнером та може використовуватись з необов'язковим параметром `align`. Значення параметру `align` та наслідки його використання для абзців ті ж самі, що і для заголовків, див. табл. 2.2. Відзначимо, що візуально в вікні браузера абзац буде відділятися від іншого тексту за допомогою порожніх рядків та можливої зміни горизонтального вирівнювання. Наведемо приклад використання тегу `<p>`:

```
<body>  
Звичайний текст  
<p align="center"> Абзац з горизонтальним вирівнюванням - "по центру" </p>  
</body>
```

Відповідне вікно браузера показане на рис. 2.6.

Парний тег **<center>** призначений для горизонтального вирівнювання елементів HTML-сторінки посередні вікна браузера.

Наведемо приклад його використання:

<body>

Звичайний текст

<center> Текст з горизонтальним вирівнюванням - "по центру" </center>

Звичайний текст </body>

Відзначимо, що в специфікаціях діючих стандартів HTML тег **<center>** позначений як не бажаний для використання, хоча і підтримується практично всіма розповсюдженими браузерами.

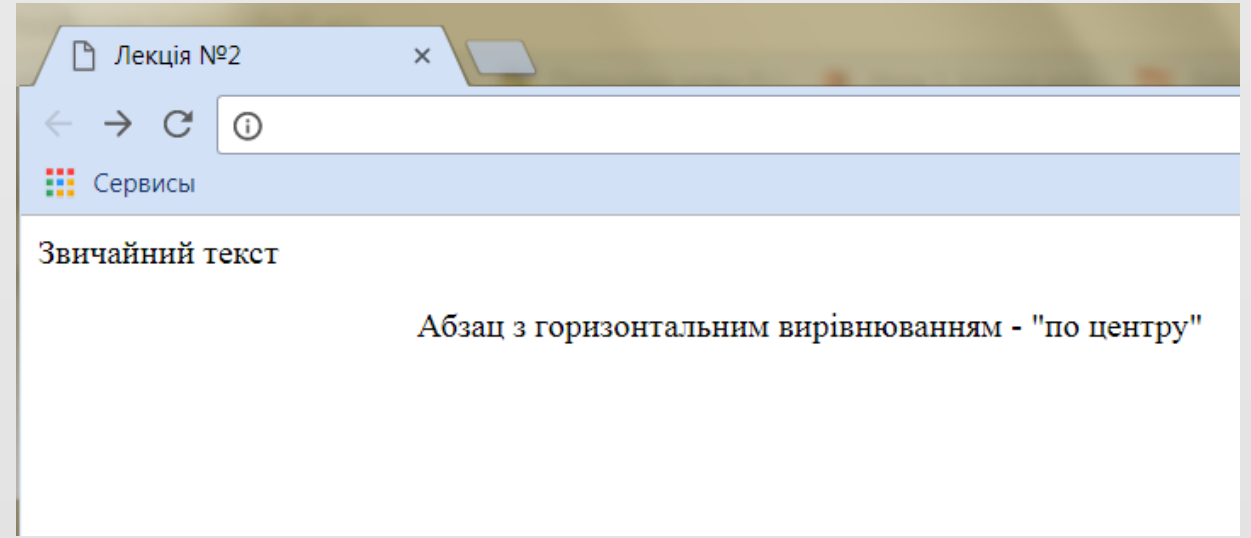


Рис. 2.6 Використання тегу абзацу

Тег-контейнер **<pre>** використовується для визначення попередньо відформатованого тексту. Текст, розміщений між тегам **<pre>** та **</pre>**, буде виведений у вікні браузера із збереженням деяких параметрів форматування, виконаних за допомогою текстових редакторів. Відзначимо, що в середині контейнеру **<pre>** не можна використовувати деякі важливі та широко вживані теги. Тому використання цього тегу досить обмежене.

Тег **<hr>** призначений для виведення в вікні браузера горизонтальної лінії, до і після якої вставляються порожні рядки. Необов'язкові параметри тегу **<hr>** та їх призначення наведені в табл. 2.3.

Параметри тегу <hr>

Параметр	Призначення
align	Визначення горизонтального вирівнювання лінії. Можливі значення: left ("по лівому краю"), center ("по центру"), right ("по правому краю"). Якщо параметр align не використаний то застосовується вирівнювання "по центру"
color	Встановлює колір лінії
noshed	Відміняє рельєфність лінії
size	Визначає товщину лінії
width	Встановлює довжину лінії

Наприклад, для визначення лінії червоного кольору, товщиною 2 пікселі, довжиною 400 пікселів, без рельєфу необхідно:

<hr color="red" size=2 width=400 noshade>

Коментарі та спеціальні символи

Таблиця 2.4

Коди спеціальних символів

Спеціальний символ	HTML-код
Амперсанд (&)	&
Знак copyright (©)	©
Знак більше (>)	>
Знак менше (<)	<
Пробіл	
Знак плюс/мінус (\pm)	±
Знак одна друга ($\frac{1}{2}$)	½
Знак одна друга ($\frac{1}{4}$)	¼
Знак одна друга ($\frac{3}{4}$)	¾
Знак "»"	»
Знак " § "	§
Знак "«"	«

В деяких випадках необхідно не показувати у вікні браузера частину HTML-коду сторінки, тобто "за коментувати" цей код. Для цього використовуються теги коментарів `<!--` та `-->`. Наприклад:

`<!-- Цей текст не буде відображатись у вікні браузера -->`

Як відомо, при створенні HTML-сторінки повинні застосовуватись тільки символи, що входять до базової частини таблиці кодів ASCII. Проте деякі спеціальні символи (наприклад, пробіл) не входять до цієї частини таблиці.

В випадку використання вказаних спеціальних символів необхідно застосовувати спеціальні HTML-коди, див. табл. 2.4.

Списки

Списки - широко розповсюджена форма показу даних як в електронних так і в друкованих документах. Мовою HTML передбачено використання трьох стандартних видів списків: маркірованого, нумерованого та списку визначень. Відзначимо, що від інших елементів на HTML-сторінці стандартні списки відділяються порожніми рядками.

Маркіровані списки

Для визначення маркірованого списку використовується тег-контейнер ``, в середині якого розміщуються всі елементи списку. При цьому кожен пункт списку повинен починатись тегом ``. Наведемо приклад HTML-коду маркірованого списку:

`<body>` Звичайний текст

``

`` Перший пункт маркірованого списку

`` Другий пункт маркірованого списку

`` Третій пункт маркірованого списку

``

Звичайний текст

`</body>`

Відображення маркірованого списку у вікні браузера показане на рис. 2.7.

Як в тезі ``, так і в тезі `` можливо використовувати необов'язковий параметр `type`, за допомогою якого визначається тип маркеру списку.

Можливі значення цього параметру:

- `disc` - маркери відображаються заповненими колами;
- `circle` - маркери відображаються не заповненими колами;
- `square` - маркери відображаються заповненими квадратами.

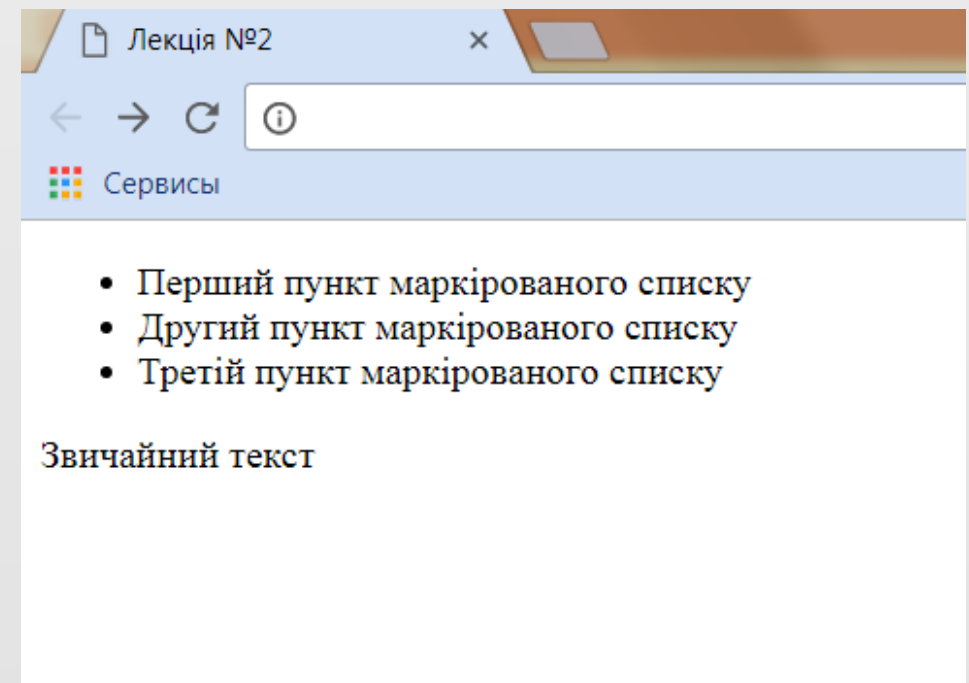


Рис. 2.7 Відображення маркірованого списку

Приклад використання різних значень параметру `type` в тезі `` маркірованого списку:

```
<ul>  
<li type="disc"> Перший пункт маркірованого списку. type="disc"  
<li type="square"> Другий пункт маркірованого списку. type="square"  
<li type="circle"> Третій пункт маркірованого списку. type="circle"  
</ul>
```

Відповідне відображення у вікні браузера показане на рис. 2.8.

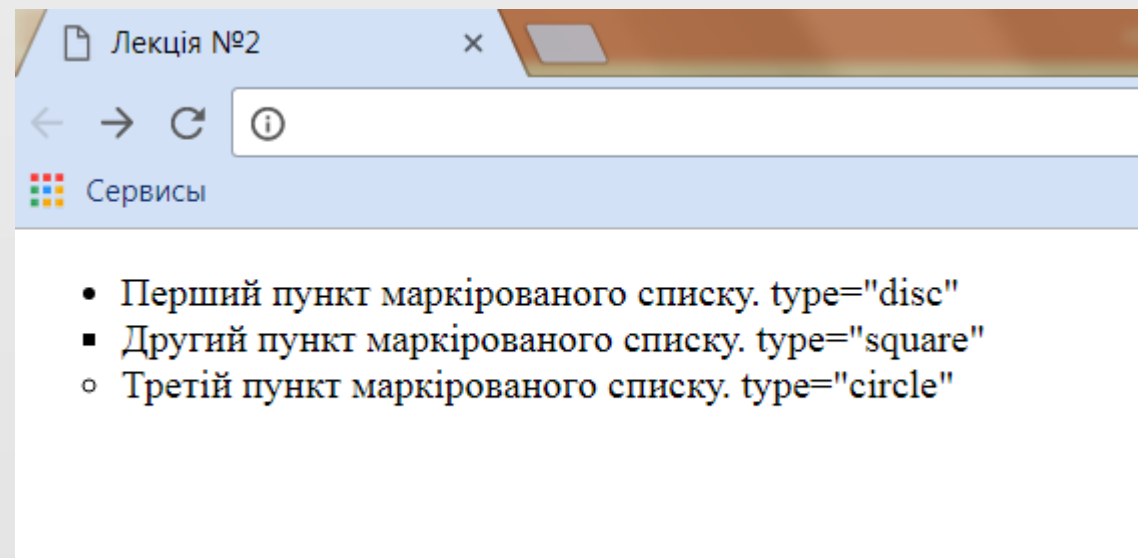


Рис. 2.8 Використання різних значень параметру type в тезі

Якщо параметр **type** в тегах **** та **** не використовується, то для всього списку використовується маркер типу **"disc"**.

Нумеровані списки

Для визначення нумерованого списку використовується тег-контейнер ****, в середині якого розміщуються всі елементи списку. При цьому кожен пункт списку повинен починатись тегом ****. Наведемо приклад HTML-коду нумерованого списку:

```
<ol>
<li> Перший пункт нумерованого списку
<li> Другий пункт нумерованого списку
<li> Третій пункт нумерованого списку
</ol>
```

Відображення такого нумерованого списку у вікні браузера показано на рис. 2.9.

Досить часто тег `` використовується з необов'язковими параметрами `type` та `start`. При цьому параметр `type` використовується також в тезі ``. Його призначенням є визначення одного з стандартних типів маркера. Відзначимо, що для нумерованого списку існує п'ять стандартних типів маркерів. Використання цього параметру в тезі `` розповсюджується на весь список, а використання в тезі `` розповсюджується тільки на поточний пункт списку. Можливі значення параметру `type` та відповідні цим значенням стандартні типи маркерів показані в табл. 3.2.

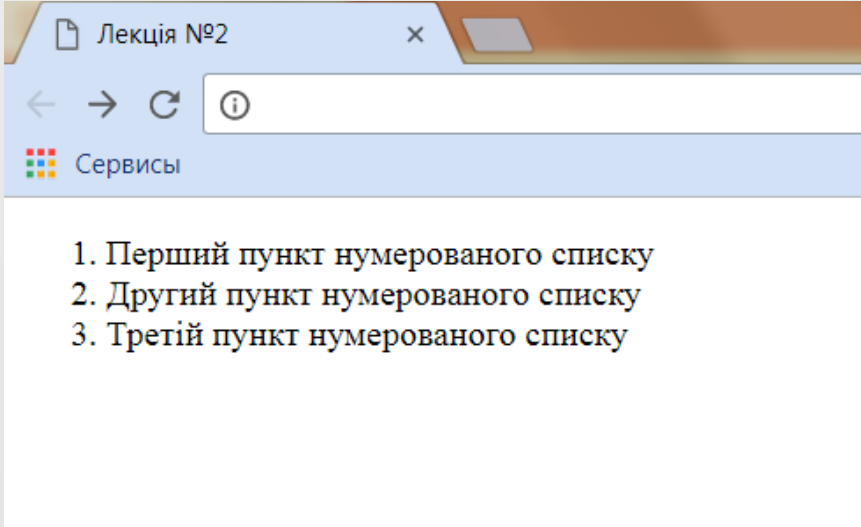


Рис. 2.9 Відображення нумерованого списку
Таблиця 3.2

Використання параметру `type`

Значення параметру <code>type</code>	Тип маркеру
1	у вигляді арабських цифр
A	у вигляді великих букв латинського алфавіту
a	у вигляді маленьких букв латинського алфавіту
I	у вигляді великих римських цифр
i	у вигляді маленьких римських цифр

Наведемо приклад нумерованого списку з маркерами у вигляді великих римських цифр. При цьому в другому пункті списку змінимо тип маркеру на "великі латинські букви".

```
<ol type="I">
```

```
<li> Перший пункт нумерованого списку.
```

```
<li type="A"> Другий пункт нумерованого списку. type="A"
```

```
<li> Третій пункт нумерованого списку.
```

```
</ol>
```

Відображення цього списку у вікні браузера показано на рис. 2.10.

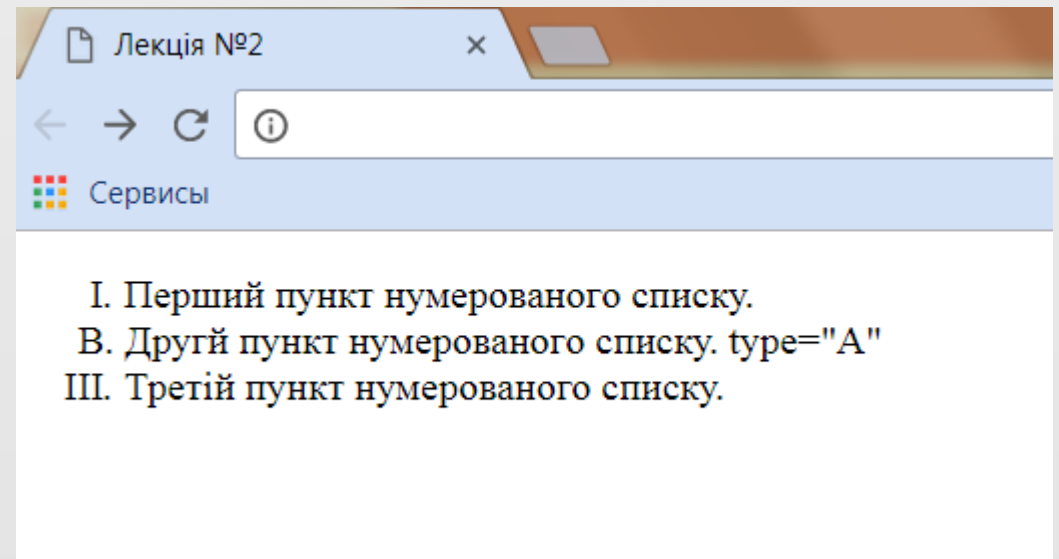


Рис. 2.10. Зміна типу маркерів в нумерованому списку

Відзначимо, що нумерація пунктів нумерованого списку не залежить від типу маркеру. Якщо параметр **type** в тегах **** та **** не зазначено, то для всього списку використовується маркер типу "1".

Параметр **start** призначений для зміни початку нумерації пунктів списку і може використовуватись тільки в тезі ****. Для зміни нумерації пунктів в середині списку в тезі **** використовується параметр **value**. Значеннями параметрів **start** та **value** можуть бути тільки натуральні числа. Наведемо приклад визначення нумерованого списку, в якому нумерація пунктів починається з п'яти, а в третьому рядку списку початок нумерації змінюється на десять:

```
<ol start="5">
```

```
<li> Перший рядок нумерованого списку.
```

```
<li> Другий рядок нумерованого списку.
```

```
<li value="10"> Третій рядок нумерованого списку.
```

```
<li> Четвертий рядок нумерованого списку.
```

```
</ol>
```

Відображення цього нумерованого списку показано на рис. 2.11.

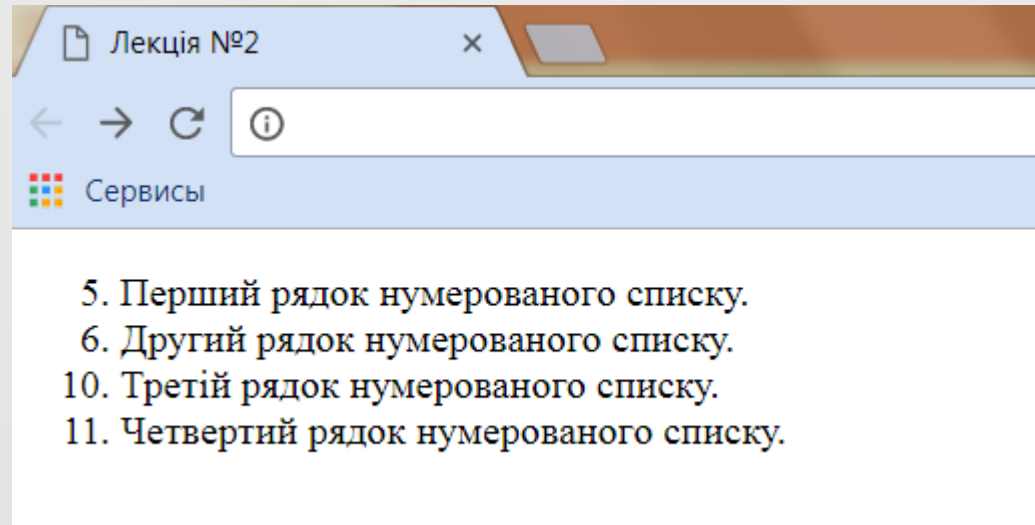


Рис. 2.11 Зміна початку нумерації пунктів нумерованого списку

Списки визначень

Список визначень використовується для розміщення на HTML-сторінці тексту, за змістом подібного до енциклопедії або словника. Тобто тексту, в якому потрібно навести термін та пояснення цього терміну, що може займати декілька рядків. На відміну від інших типів списків кожен пункт списку визначень складається із двох частин. В першій частині пункту записується термін, що потребує визначення. В другій частині пункту записується текст, що пояснює зміст терміну.

Список визначень задається за допомогою тегу-контейнеру `<dl>`. Для визначення терміну використовується тег `<dt>`, а для визначення пояснення використовується тег `<dd>`. Наведемо приклад списку визначень для розміщення на HTML-сторінці деяких стандартних термін в галузі Web-програмування:

`<dl>`

`<dt>`Інтернет (мережа Інтернет)

`<dd>`Сукупність мереж та обчислювальних засобів, які використовують стек протоколів TCP/IP (Transport Control Protocol/Internet Protocol), спільний простір імен та адрес для забезпечення доступу до інформаційних ресурсів мережі будь-якій особі;

`<dt>`Гіпертекстове посилання

`<dd>`Адреса іншого мережевого інформаційного ресурсу у форматі URL, який тематично, логічно або будь-яким іншим способом пов'язаний з документом, у якому це посилання визначене.

`</dl>`

Відображення цього списку визначень у вікні браузера показано на рис. 2.12.

Як видно з рисунка 2.12, у вікні браузера термін відділяється від свого пояснення за допомогою переводу рядка та табуляції.

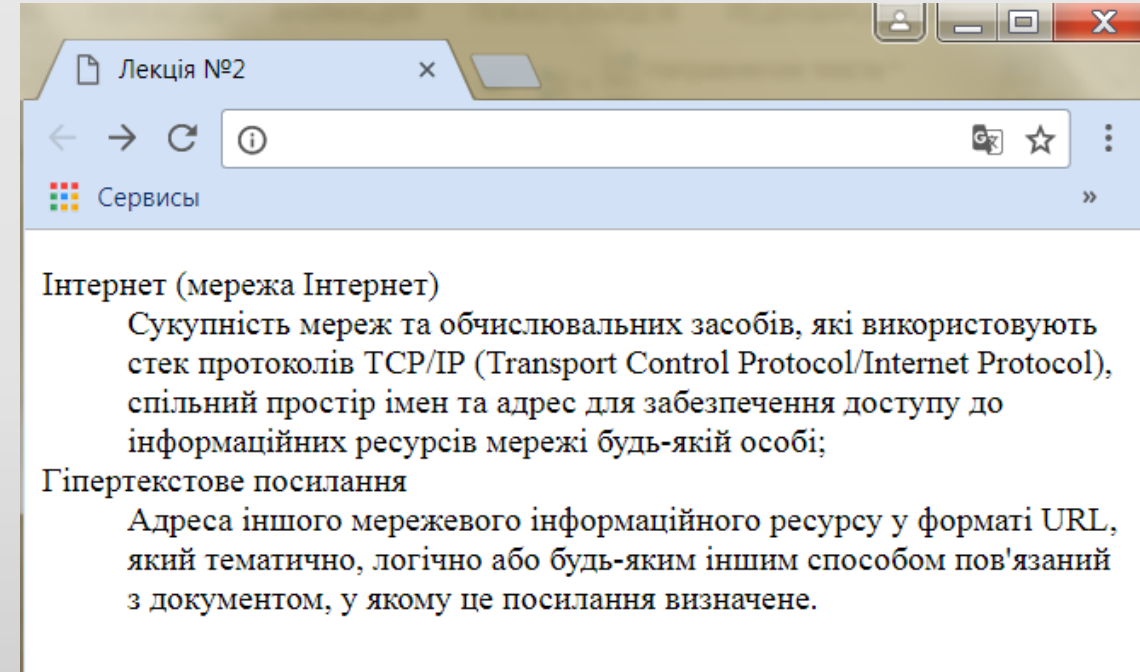


Рис. 2.12. Список визначень

Гіперпосилання

Як відомо, гіперпосилання призначені для зв'язування одного Web-ресурсу з іншим. Гіперпосилання складається із двох частин. Перша частина – це об'єкт, який користувач бачить у вікні браузера, і вибір якого призводить до переходу на цільовий Web-ресурс, URL-адреса якого вказана в другій частині гіперпосилання. В якості цього об'єкту можуть бути текст або/та зображення. В залежності від цього класифікують текстові та графічні гіперпосилання. При типових настройках браузера гіперпосилання виділяються на HTML-сторінці за допомогою кольору. Крім того текст гіперпосилання виділений за допомогою підкреслення. Для визначення гіперпосилання в коді HTML-сторінки використовують парний тег `<a>`. Основним параметром цього тегу є параметр `href`, що задає URL-адресу гіперпосилання. Синтаксис запису гіперпосилання такий:

` Текстовий або графічний об'єкт `

Приклади запису текстового та графічного гіперпосилання показані нижче:

` Тексто́ве гіперпоси́лання `

` `

Вказана URL-адреса може бути відносна або абсолютна. При використанні відносної адреси повний шлях до цільового ресурсу не заданий. В цьому випадку визначення місцезнаходження цільового ресурсу відбувається з урахуванням місцезнаходження HTML-сторінки, в якій використане гіперпосилання. Наприклад, якщо в файлі 1.html визначене гіперпосилання:

` Текстовий або графічний об'єкт `

то мається на увазі, що цільова HTML-сторінка 2.html знаходиться в тому ж каталозі того ж комп'ютера, в якому знаходиться і HTML-сторінка 1.html.

Абсолютною називається URL-адреса, в якій повністю визначено комп'ютер, каталог та файл цільового ресурсу. Наприклад:

` Абсолютне гіперпосилання`

Важливим параметром тегу `<a>` є параметр `target`, за допомогою якого можливо вказати вікно браузеру, в яке повинен бути завантажений цільовий ресурс. Для завантаження цільового ресурсу в нове вікно браузера значення цього параметру має бути `"_blank"`. Наприклад, для визначення текстового гіперпосилання, вибір якого призводить до відображення в новому вікні браузера файлу 2.html, необхідно:

` Текст `

Відзначимо, що при відсутності параметру `target` цільовий ресурс буде завантажуватись в вікно браузера, в якому відображається HTML-сторінка з гіперпосиланням.

Цікавою є можливість гіперпосилань задати в якості цільового ресурсу певну частину власної HTML-сторінки. Наприклад, зручно читати великий документ, коли він має зміст з гіперпосиланнями на різні розділи. В цьому випадку перехід по гіперпосиланню до деякого розділу (який може знаходитись в кінці документу) призведе до відображення цього розділу на екрані. Такі гіперпосилання називаються внутрішніми. Для їх побудови необхідно за допомогою тегу `<a>` вказати місце переходу (покажчик) та за допомогою параметру `name` тега `<a>` присвоїти йому ім'я.

Наприклад, для переходу на 12 розділ документу на початку тексту цього розділу розмістимо покажчик з іменем `"go12"`:

`Текст розділу 12`

Відзначимо, що в цьому випадку параметр `href` не використовується, а ім'я покажчика повинно бути записане за допомогою літер англійського алфавіту та цифр.

Особливістю гіперпосилання для переходу в середині HTML-сторінки є використання в параметрі `href` імені покажчика з перфіксом `#`. Наприклад:

`Перехід до розділу 12`

Рисунки

Для вставки рисунка в HTML-сторінку використовується тег ****. Основним параметром цього тегу є параметр **src**, за допомогою якого вказується адреса файлу, в якому і зберігається рисунок. Цей же тег також можливо використовувати для показу відео-роликів. В цьому випадку використовується параметр **dynsrc**, за допомогою якого вказується адреса відео файлу. Найпоширенішими форматами файлів-рисунків є формати jpg, gif, png, а відео файлів - avi.

Відзначимо, що адреса файлу - рисунка може бути вказана в абсолютному та відносному форматі. При використанні абсолютного формату в адресі повністю визначено комп'ютер, каталог та файл рисунку.

Наприклад, для вставки рисунка, що зберігається в файлі logo.gif за адресою www.picture.com.ua/picture необхідно:

```

```

При використанні відносного формату визначення місцезнаходження файлу-рисунку відбувається з урахуванням місцезнаходження даної HTML-сторінки.

Наприклад, для вставки в HTML-сторінку prim.html рисунка, який зберігається в файлі logo.gif, що знаходиться в одному каталозі з файлом prim.html необхідно:

```

```

Графічні файли зручно зберігати в окремому каталозі, наприклад, в каталозі з іменем img. В цьому випадку для вставки в HTML-сторінку prim.html рисунка logo.gif необхідно:

```

```

Відзначимо, що каталог img та файл prim.html повинні знаходитись в одному каталозі. Крім обов'язкового параметру **src** тег **** має декілька необов'язкових параметрів, див. табл. 3.1.

**Параметри тегу **

Назва параметру	Призначення
alt	Визначення альтернативного тексту
width	Ширину рисунку в пікселях
height	Висоту рисунку в пікселях
hspace	Відступ від рисунку до інших об'єктів на HTML-сторінці по вертикалі
vspace	Відступ від рисунку до інших об'єктів на HTML-сторінці по горизонталі
border	Товщина рамки навколо рисунку
lowsrc	Задає файл з альтернативним зображенням
align	Вирівнювання рисунку відносно інших об'єктів на HTML-сторінці
src	Адреса графічного файлу
dynsrc	Адреса відео файлу
start	Задає момент початку прокрутки відео ролика. Можливі значення: fileopen - ролик починається відразу після завантаження відео файлу, mouseover - ролик починається при наведенні миші на зображення.
loop	Задає кількість повторів відео - ролика. Якщо значення дорівнює -1, то програвання відбувається безперервно

Розглянемо призначення вказаних параметрів докладніше. Досить часто при перегляді HTML-сторінок користувачі використовують браузер, що працює в режимі відключення завантаження зображень. Як відомо, цей режим дозволяє зменшити час відображення HTML-сторінки в вікні браузера. В цьому випадку замість зображення на екрані з'являється альтернативний текст, заданий в параметрі **alt**. Крім цього альтернативний текст відображається в якості підказки при знаходженні курсору "миші" в області рисунку.

Приклад використання параметру **alt**:

```

```

Параметри **width** та **height** призначені для визначення розмірів рисунку у вікні браузера. Відзначимо, що при зміні розмірів зображення розмір файлу не змінюється, тобто час завантаження рисунку не змінюється.

Наведемо приклад застосування параметрів **alt**, **width** та **height** для одного і того рисунку:

```

```

```

```

```

```

Відповідне вікно браузера показано на рис. 2.7.

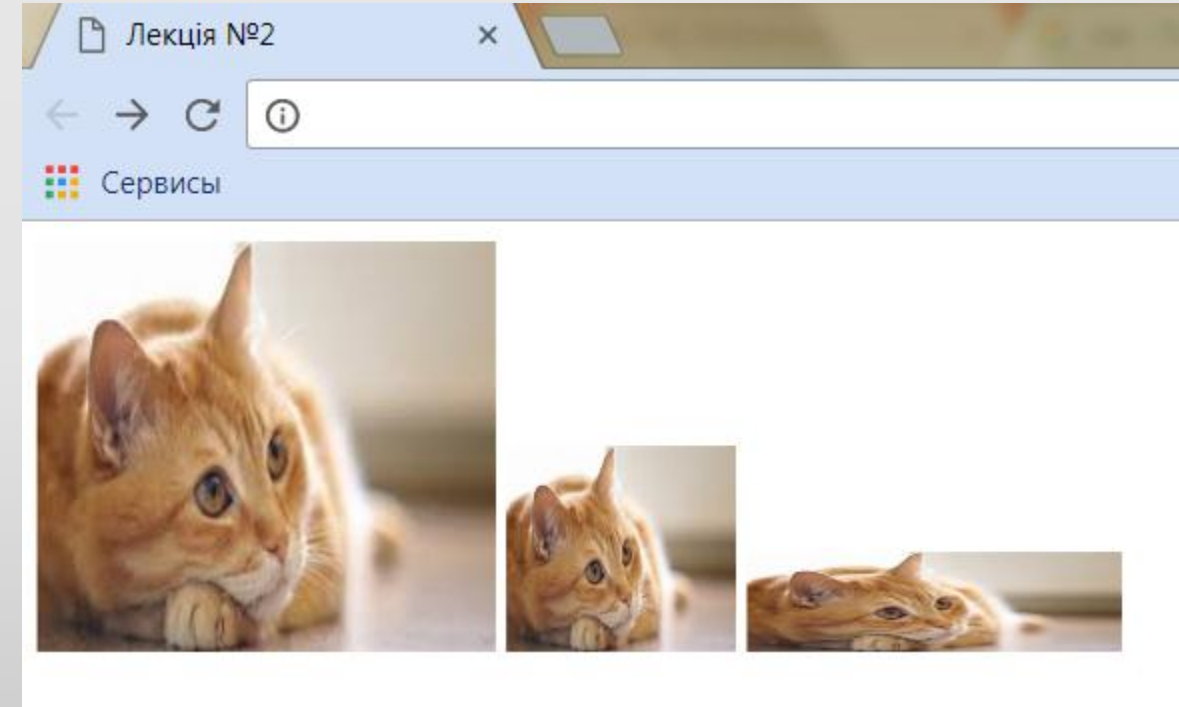


Рис. 2.7 Використання параметрів alt, width та height тегу

Слід зазначити, що зміна розмірів рисунку може призвести до втрати якості зображення, особливо за рахунок втрати пропорцій між початковою висотою та шириною.

Практичний досвід показує, що навіть при дотриманні пропорцій збільшення початкових розмірів зображень-фотографій більш ніж на 10-15% призводить до чуттєвої втрати якості.

При цьому зменшення розмірів зображень позначається на якості їх відображення набагато менше, але їх використання не оптимальне з точки зору часу завантаження.

Параметр `border` дозволяє визначити товщину рамки навколо рисунку.

Застосуємо цей параметр із значенням 2 для рисунків попереднього прикладу:

```

```

```

```

```

```

Відповідне вікно браузера показано на рис. 2.8.

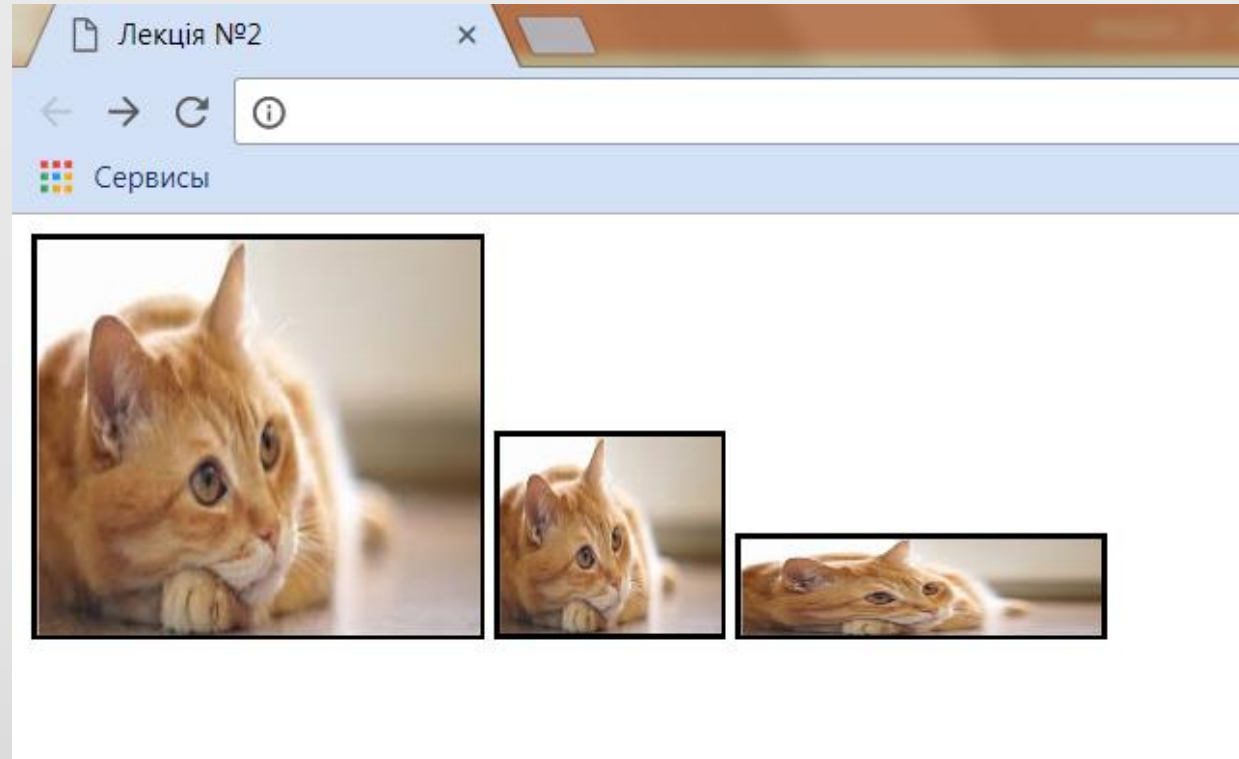


Рис. 2.8. Використання параметрів `border`, `width` та `height` тегу ``

Параметри `hspace` та `vspace` дозволяють в пікселях задавати відступи від рисунку до інших об'єктів на HTML-сторінці.

Покажемо застосування цих параметрів із значеннями 20 та 20 для попереднього прикладу:

```



```

Результат застосування показаний на рис. 2.9.

Параметр `lowsrc` дозволяє визначити файл з альтернативним зображенням, яке з'являється у вікні браузера до появи основного зображення. Альтернативне зображення, як правило, це рисунок того ж змісту, що і основний, але менш якісний. По цій причині файл альтернативного зображення менший за розміром, а тому завантажується набагато швидше ніж основний. Це дозволяє швидко показати загальні риси зображення.

Приклад використання:

```

```

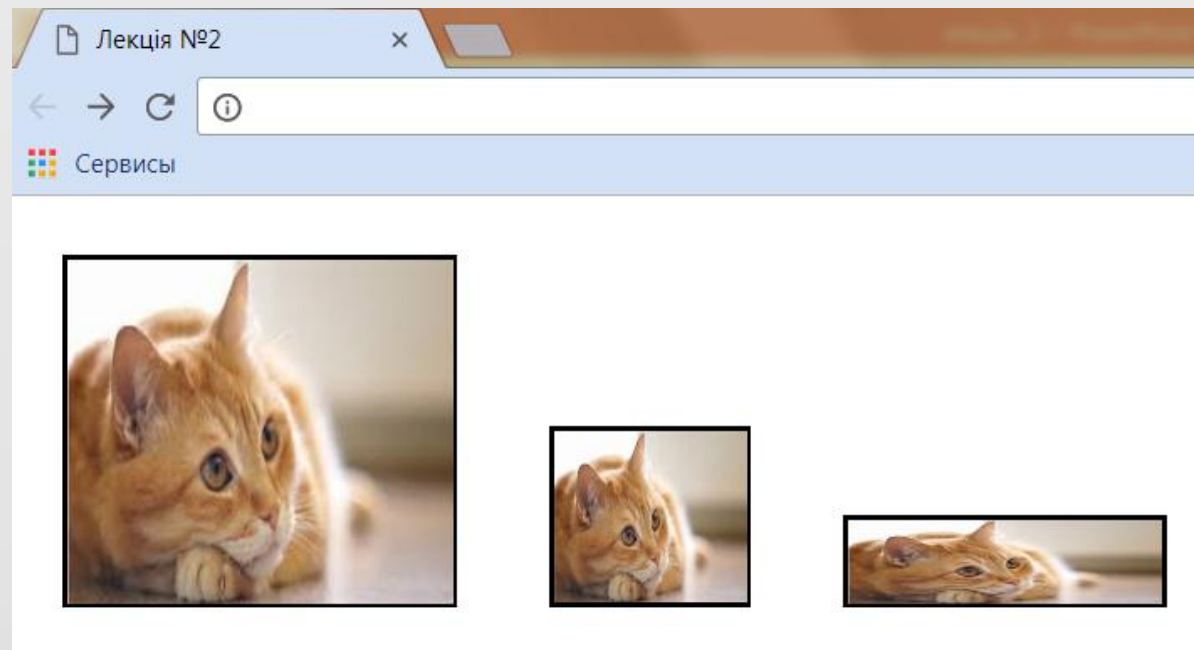


Рис. 2.9. Використання параметрів `hspace`, `vspace`, `border`, `width` та `height` тегу ``

Параметр **align** використовується для визначення горизонтального або вертикального вирівнювання рисунку відносно інших об'єктів на HTML-сторінці. Параметри вирівнювання залежать від значень параметру **align**.

При використанні цього параметру із значеннями **left** або **right** рисунок розміщується відповідно до лівої або правої сторони вікна браузера, а текст "обтікає" зображення з протилежної сторони. Відзначимо, що в цьому випадку текст може займати декілька рядків.

Приклад використання параметру **align**:

```
<body>  

```

Приклад використання параметру **align="right"**.

Рисунок розміщений біля правої сторони вікна браузера. Текст обтікає зображення з лівої сторони.
</body>

Відповідне вікно браузера показано на рис. 2.10.

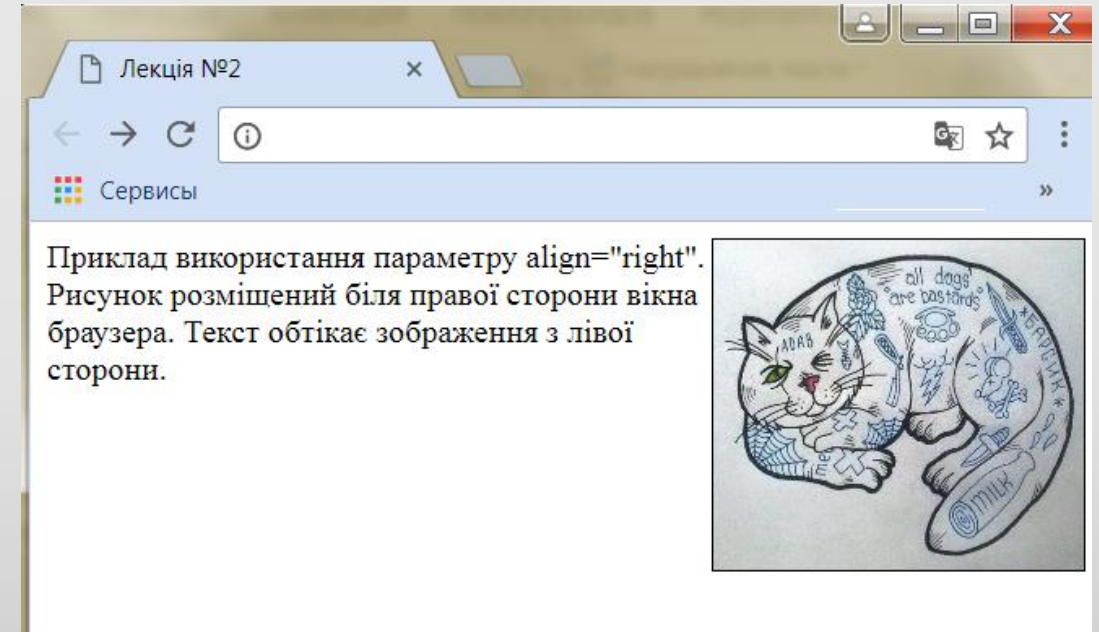


Рис. 2.10. Приклад вирівнювання рисунка по правому краю вікна браузера

Крім **left** та **right** параметр **aling** може приймати наступні значення:

- **top** - верхня межа зображення вирівнюється по найбільш високому елементу рядка;
- **texttop** - верхня межа зображення вирівнюється по найбільш високому текстовому елементу рядка;
- **middle** - середина зображення вирівнюється по базовій лінії рядка;
- **absmiddle** - середина зображення вирівнюється по середині рядка;
- **bottom** - нижня межа зображення вирівнюється по базовій лінії рядка;
- **absbottom** - нижня межа зображення вирівнюється по нижній межі рядка.

В цих випадках зображення можна розглядати як звичайний елемент рядка.

Відзначимо, що базова лінія тексту - це нижня лінія рядка тексту без врахування нижньої частини деяких символів (p, j, y).

Наведемо приклад використання параметру **align** тегу **** з значеннями **top**, **middle** та **bottom**:

<body>

Приклад використання параметру **align="top"**. **
**

Приклад використання параметру **align="middle"**. **
**

**** Приклад використання параметру **align="bottom"**.

</body>

Відповідне вікно браузера див. на рис. 2.11.

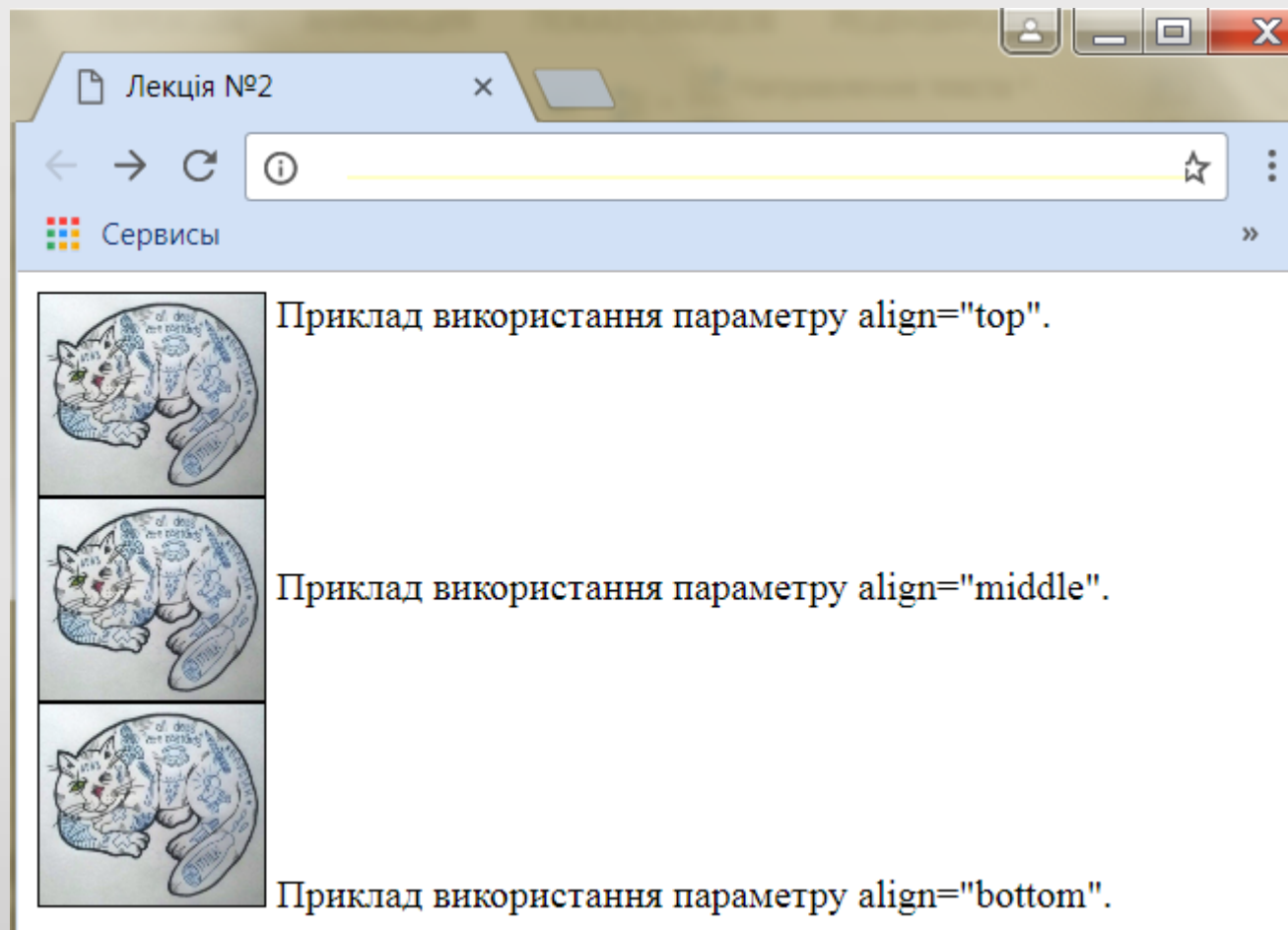


Рис. 2.11. Приклад використання параметру align теги з значеннями top, middle та bottom

Лекцію закінчено.
Дякую за увагу!