

# ЛАБОРАТОРНА РОБОТА №1

## Вивчення HTML тегів (а не створення сайту)

### Мета роботи:

- навчитись створювати HTML документи
- вивчити та навчитись використовувати теги для створення web-сторінок
- зрозуміти, які теги відносяться до застарілих, а які до технології «html5»(<header> ... <footer> ...

### Заувага:

В цій лабораторній роботі вивчаємо і використовуємо лише html. Завданням роботи не є створити сайт, а вивчити html теги та їх атрибути, знати які з них є актуальними сьогодні, а які не використовуються. Використання інших web технологій (css, js) в цій лабораторній роботі є **забороненим**.

### Теоретичні відомості:

Для вивчення використовуємо сучасний сайт  
<https://www.w3schools.com/html/default.asp>


Загальна структура html - коду для web сторінки має вигляд

```

<html>
  <head>
    <title>Заголовок документа – Ваше прізвище та ім'я </title>
  </head>

  <body>
    Тіло документа
  </body>
</html>

```



можна розмістити (в Тіло документа)

1. Гіперпосилання
2. Блоки (тег div) та спеціалізовані блоки (<header>...<footer>)
3. Рисунки
4. Текст
5. Спеціальні символи
6. Таблиці
7. Списки
8. Форми
9. Фрейми

Для цього в тіло документа потрібно написати відповідні теги та їх атрибути.

## ЗАВДАННЯ.

1. Створити папку з назвою web проекту (проект на довільну тему). Наприклад: фотоблог.  
Створити в папці
  - Папку з назвою image (для малюнків)
  - html сторінку. Для цього відкрити довільний текстовий редактор (наприклад NotePad++, Atom,, Sublime .....)  
[http://www.w3schools.com/html/html\\_editors.asp](http://www.w3schools.com/html/html_editors.asp)
2. Помістити у файл html загальну структуру web сторінки, дати заголовок <title>...</title> – тему сайту (це важливо для SEO оптимізації), написати ваше прізвище в <body> .... </body> та зберегти файл, як index.html
3. Відкрити файл в браузері. Перевірити чи правильно відображається заголовок і прізвище.
4. В тілі документу розмістити блоки та вкладені блоки не менше ніж 15. Використати теги <div> та теги html5 для розмітки сторінки. Врахувати, що якщо в блоці нічого не має, то він невидимий.
5. В блоках ( (тег div) (можна в спеціалізованих блоках (<header>...<footer>)) розмістити
  1. Гіперпосилання (на сторінки сайтів)
  2. Вкладені блоки (як вкладені цикли)
  3. Рисунки
  4. Текст з заголовком.- відформатовані тегамі для форматування тексту
  5. Спеціальні символи
  6. Таблицю.
  7. Списки (всі види)
  8. Форми (не менше 7 видів форм)
  9. ...
6. Зберегти файл та відкрити в браузері.

Для захисту лабораторної необхідно знати html теги та їх атрибути.

### Наприклад питання

1. **Напишіть декілька тегів для обробки тексту, які ви використали на html сторінці.**
2. **Який html код потрібно написати для того щоб на сайті з'явився рисунок шириною 100px.**
3. **Які види форм ви знаєте? Напишіть зразок html коду для них.**
4. **Напишіть html теги структури web сторінки**
5. ....

# Перелік тегів

## Приклади використання тегів

Реалізувати в тілі документа приклади на тему:

### використання тегів H1>...<H6

```
<h1>Це заголовок першого рівня</h1>
  <h2>Це заголовок другого рівня</h2>
  <h3>Це заголовок третього рівня</h3>
  <h4 align="right">Це заголовок четвертого рівня</h4>
  <h5 align="center">Це заголовок п'ятого рівня</h5>
  <h6 align="jastify">Це заголовок шостого рівня</h6>
    Це просто текст
```

### використання тегу <HR>

Лінія без параметрів

```
<hr>
```

Лінія по центру

```
<hr align="center" color="red" size=2 width=50%>
```

Лінія зліва

```
<hr align="left" color="blue" size=3 width="150">
```

### використання тегу <PRE>

```
<pre>
```

```
Як набрали,
    так і   відображається.
```

```
</pre>
```

### використання тегу <abbr>

```
<abbr title="HyperText Markup Language">
  HTML
</abbr>
```

### використання тегу <address>

Пишіть за адресою:

```
<address>
```

```
вул.Драгоманова, 50
```

```
</address>
```

### Інші приклади використання тегів

```
<cite>
```

```
Це текст в тегах cite
```

```
</cite><br>
```

<dfn>  
 Це текст в тегах dfn  
 </dfn><br>

<em>  
 Це текст в тегах em  
 </em><br>

<i>  
 Це текст в тегах i  
 </i>  
 </body>

Просто текст<br>

<strong>  
 Це текст в тегах strong  
 </strong><br>

<b>  
 Це текст в тегах b  
 </b>

Просто текст<br>

<ins>  
 Це текст в тегах ins  
 </ins><br>

<u>  
 Це текст в тегах u  
 </u>

$y=x^2$  - рівняння параболи.<br><br>

$H_2O$  - формула води.

<font face="Verdana" size=5 color="red">

Це застарілий тег, який зараз не використовується, а використовуєть технологія CSS.  
 Це текст червоний, розміру 5.  
 </font><br>

Реалізувати в тілі документа приклади:

3 &gt; 2 <br>

4 &lt; 5 <br>

32 &deg; <br>

Заувага! Деякі з тегів для зміни дизайну тексту в сучасному веб-дизайні не використовуються, оскільки управління дизайном здійснюється за допомогою CSS.

## Теги таблиць

<table> відкриває тег таблиці

<tr> тег - початок рядка

<th> - *table header (заголовок таблиці) текст усередині цих тегів буде виділений напівжирним шрифтом і відцентрований* </th> закриваючий тег осередку

<td> відкриває тег комірки

</td> закриваючий тег комірки

</tr> тег - кінець рядка

</table> закриваючий тег таблиці

### атрибути

width - ширина таблиці вказується в "%" або в пікселях "800"

border - товщина рамки

cellspacing - відстань між комірками

cellpadding - відстань між рамкою комірки і текстом

align - вирівнювання по горизонталі (за замовчуванням зліва = "left" центр = "center" праворуч = "right")

valign - вирівнювання по вертикалі (верх = "top" середина = "middle" низ = "bottom")

colspan = x - кількість комірок, що об'єднуються по горизонталі

rowspan = x - кількість комірок, що об'єднуються по вертикалі

### Приклади таблиць

```
<table border="1" cellpadding="0" cellspacing="0" width="100%">
<tr>
<th width="25%">навігація</th>
<th width="75%">заголовок</th>
</tr>
<tr>
<td width="25%" valign="top">меню</td>
<td width="75%" valign="top">контент</td>
</tr>
</table>
```

```
<table border="1" cellpadding="0" cellspacing="0" width="100%">
<tr>
<th colspan=2>логотип</th>
</tr>
<tr>
<th width="25%">навігація</th>
<th width="75%">заголовок</th>
</tr>
<tr>
<td width="25%" valign="top">
```

```

<ul>
<li><a href="index.html" title="Гіперпосилання">Гіперпосилання</a></li>
<li><a href="index.html" title="Гіперпосилання">Гіперпосилання</a></li>
<li><a href="index.html" title="Гіперпосилання">Гіперпосилання</a></li>
</ul>
</td>
<td width="75%" valign="top">контент</td>
</tr>
<tr>
<td colspan=3> низ</td>
</tr>
</table>

```

```

<table border="1" cellpadding="0" cellspacing="0" width="100%">
<tr>
<th colspan=3>логотип</th>
</tr>
<tr>
<th width="20%">навігація</th>
<th width="60%">заголовок 1</th>
<th width="20%">заголовок 2</th>
</tr>
<tr>
<td width="20%" valign="top">
<ul>
<li><a href="index.html" title="Ссылка">Гіперпосилання</a></li>
<li><a href="index.html" title="Ссылка">Гіперпосилання</a></li>
<li><a href="index.html" title="Ссылка">Гіперпосилання</a></li>
</ul>
</td>
<td width="60%" valign="top">контент</td>
<td width="20%" valign="top">суб-контент</td>
</tr>
<tr>
<td colspan=3>низ</td>
</tr>
</table>

```

```

<table border="1" cellpadding="0" cellspacing="0" width="100%" id="table-1">
<tr>
<td width="100%">логотип</td>
</tr>
<tr>
<td width="100%">
<table border="1" cellpadding="0" cellspacing="0" width="100%" id="table-2">
<tr>
<td width="20%" valign="top">>

```

```

<ul>
<li><a href="index.html" title="Ссылка">Гіперпосилання</a></li>
<li><a href="index.html" title="Ссылка">Гіперпосилання</a></li>
<li><a href="index.html" title="Ссылка">Гіперпосилання</a></li>
</ul>

</td>
<td width="50%" valign="top">контент-верх</td>
<td width="30%" valign="top">суб-контент</td>
</tr>
<tr>
<td width="20%">текст</td>
<td width="50%">контент-низ</td>
<td width="30%">суб-контент-низ</td>
</tr>
</table>
</td>
</tr>
<tr>
<td width="100%">підвал</td>
</tr>
</table>

```

## Теги для списків

### Списки бувають

**нумеровані** (теги <ol> <li> </li> </ol> )

**марковані** (теги <ul> <li> </li> </ul> )

**визначень** (теги <dl> <dt> </dt> <dd> </dd> </dl> )

**вкладені** (використання тегів нумерованих і маркованих списків вкладених одні в одні)

### Нумеровані списки

Такий список є переліком "пронумерованим" яким-небудь чином. Це можуть бути арабські цифри, цифри римського алфавіту або букви.

Щоб вказати браузеру, що список буде нумерованим, використовуються теги <ol></ol>.

Увесь список розташовується між цими тегами і кожен його пункт задається тегами <li></li>.

Приклад коду для нумерованого списку:

```

<ol>
  <li>книги</li>
  <li>зошити</li>
  <li>олівці</li>
</ol>

```

Результат

1. книги
2. зошити

### 3. олівці

тег <ol> має **атрибути**

**type** – атрибут за допомогою якого можна змінити нумерацію. По замовчуванню нумерація ведеться арабськими числами.

- - type="A" - нумерація заголовними латинськими буквами (A, B, C).
- - type="a" - нумерація прописними латинськими буквами (a, b, c).
- - type="I" - нумерація великими римськими цифрами (I, II, III).
- - type="i" - нумерація маленькими римськими цифрами (i, ii, iii).

**start** - Якщо нам треба почати нумерацію не з одиниці, то слід використовувати параметр start, як значення якого вказується число, з якого починатиметься список (якщо нумерація задається за допомогою букв, то це число вказуватиме на позицію в алфавіті).

тег <li> має **атрибут value**, який дозволяє змінити номер даного елемента списку. При чому змінюється номер всіх наступних номерів.

**Приклад** з використанням всіх атрибутів нумерованих списків

```
<ol type="I" start="3">
  <li>білий</li>
  <li>рожевий</li>
  <li>червоний</li>
  <br>.....
  <li value="7">фіолетовий</li>
</ol>
```

Результат

III. білий  
IV. рожевий  
V. червоний

.....  
VII. фіолетовий

### Марковані списки

У маркованому списку замість букв і цифр використовуються маркери. Щоб вказати браузеру, що список буде маркованим, використовуються теги <ul></ul>. Увесь список розташовується між цими тегами і кожен його пункт задається тегами <li></li>.

За замовчуванням маркер відображується у вигляді чорного кружечка. Це можна змінити за допомогою **атрибути** тега <ul> **type** :

- - type="disc" - зафарбований кружечок.
- - type="circle" - порожній кружечок.
- - type="square" - зафарбований квадратик.

Приклад коду маркованих списків:

```
<ul type="circle">
  <li>комп'ютери</li>
```



```

<li>телефони</li>
<li>флешки</li>
</ul>

```

### Списки визначень (рідко використовуються)

Такі списки використовуються в словниках. Кожен пункт складається з двох частин: термін і його визначення. Щоб вказати браузеру, що буде список визначень, використовуються теги `<dl></dl>`. Кожен термін розміщуємо в теги `<dt></dt>`, а їх визначення - в теги `<dd></dd>`.

Приклад

```

<dl>
  <dt>Термін 1</dt>
  <dd>Визначення терміна 1</dd>
  <dt>Термін 2</dt>
  <dd>Визначення терміна 2</dd>
</dl>

```

### Вкладені списки

Бувають ситуації, коли в елемент списку необхідно включити свій список. Такі списки називаються багаторівневими або вкладеними. Для цього використовується комбінація вже відомих нам тегів списків.

## Теги для ФОРМ

*Види форм*

1. текстове поле
2. текстове поле для введення пароля
3. прапорці
4. перемикачі
5. кнопки
6. поле для файлів
7. багаторядкове текстове поле
8. списки, що розкриваються
9. надписи

Отже, в html форма задається тегами `<form></form>`. Усі інші елементи форми розташовуються між цими тегами.

У тезі `<form>` є кілька атрибутів. Атрибут `action` або `method`. Атрибут `method` визначає, як дані посилаються сценарієм, який буде їх обробляти. Двома найбільш поширеними методами є “GET” та “POST”. Метод “GET” посилає дані в URL браузера. Якщо немає спеціальної причини для використання “GET”, ймовірно, краще не використовувати його, якщо потрібно послати секретну інформацію, так як хто завгодно зможе побачити інформацію, що пересилається через URL. Метод “POST” посилає дані за допомогою сценарію, який підтримує форму, або в повідомленні e-mail, яке надсилається адміністратору сайту, або в базу даних для зберігання і подальшого використання, а не в URL, як метод “GET”. Метод “POST” є більш захищеним і зазвичай кращим варіантом.

Якщо ви стурбовані безпекою даних у формі, наприклад, якщо ви посилаете номер кредитної карти на сайт магазину, то потрібно використовувати протокол `https` за допомогою рівня захищених сокетів (SSL). По суті це означає, що дані будуть надсилатися через протокол `https`, а не через протокол `http`. Подивіться на URL, коли наступного разу будете робити покупку на сайті магазину, або при використанні онлайн-банку – ви, ймовірно, побачите в панелі адреси `https://`, а не `http://`. Різниця полягає в тому, що з'єднання `https` трохи повільніше, ніж `http`, але дані шифруються, тому при передачі дані будуть недоступні зловмисникові. При необхідності з'ясуйте у провайдера Web-хостингу, як вони можуть забезпечити послуги `https` та SSL.

## 1. Текстове поле

Просте однорядкове текстове поле, в яке можна вводити і редагувати текст. Задається тегом `<input>`

```
<form name="form1">

  <input type="text" name="text1" size="20"
    maxlength="50" value="Введіть текст">

</form>
```

### Атрибути:

**name** - ім'я елемента

**type** - тип елемента (в даному випадку - `text`)

**size** - розмір текстового поля в символах, які одночасно будуть видні, при введенні більшої кількості символів, вони прокручуватимуться

**maxlength** - максимальна кількість символів, яку можна ввести в поле, якщо опустити цей параметр, то число символів буде необмеженим

**value** - текст, який відображатиметься (його можна стерти), за відсутності цього параметра поле буде порожнім.

Можливі ще два атрибути:

**disabled** - блокує поле від будь-яких змін

**readonly** - робить поле доступним тільки для читання.

Приклад:

```
<form name="form1">

<input type="text" name="text1" size="20"
      maxlength="50" value="неактивное поле" disabled>

<input type="text" name="text1" size="20"
      maxlength="50" value="только для чтения" readonly>

</form>
```

## 2. Текстове поле для введення пароля

Це таке ж текстове поле, як і попередній елемент. Різниця тільки в тому, що текст, що вводиться, не відображується, замість нього з'являються спеціальні символи, наприклад зірочки. Найчастіше використовується при введенні паролів. Усі параметри такі ж, як у простого текстового поля, окрім параметра **type="password"**.

Приклад:

```
<form name="form1">

Введіть пароль:<br>
<input type="password" name="text1" size="20"
      maxlength="50">

</form>
```

## 3. Прапорці (checkbox)

Приклад:

Якими мовами ви володієте:

☐ англійський ☐ німецький ☐ іспанський ☐ французький

Він задається все тим же тегом `<input>`, причому один тег задає один прапорець. Потрібно чотири прапорці і чотири рази писати `input`.

Приклад:

```
<form name="form1">

Якими мовами ви володієте:<br>

<input type="checkbox" name="lan 1" value="english"
      checked>англійський
<input type="checkbox" name="lan 2" value="german">
                                     німецький
```

```

☐

```

Розглянемо його атрибути:

**type** - тип елементу (в даному випадку - checkbox)

**name** - ім'я елементу, вказує програмі обробникові форми, який пункт вибрав користувач

**value** - значення елементу, вказує програмі обробникові форми значення пункту, який вибрав користувач. У нашому прикладі вибраний пункт англійський, отже, програма-обробник отримає: lan1="english"

**checked** - ним зазвичай позначають найбільш вірогідні для вибору пункти, користувач клацанням миші може вибрати інші пункти.

#### 4. Перемикачі (radio)

Відмінність від прапорців - можна вибрати тільки один пункт. У зв'язку з цим значення параметра name мають бути однакові для усіх елементів групи. Атрибут type="radio", усі інші такі ж, як у прапорців.

Приклад:

```

<form name="form1">

Вкажіть вашу стать:<br>

☐

```

#### 5. Кнопки

Існує чотири види кнопок :

**submit** - кнопка відправки вмісту форми web -серверу. Її атрибути:

type="submit" - тип кнопки,

name - ім'я кнопки

value - напис на кнопці.

**image** - графічна кнопка відправки вмісту форми web -серверу. Для її використання необхідно підготувати картинку кнопки, а потім використовувати її у вигляді кнопки. Її параметри:

`type="image"` - тип графічної кнопки,

`name` - ім'я кнопки,

`src` - адреса картинки для кнопки.

**reset** - кнопка, що дозволяє відновити усі значення за умовчанням у формі. Її параметри:

`type="reset"` - тип кнопки очищення,

`name` - ім'я кнопки,

`value` - напис на кнопці.

**button** - довільна кнопка, її дії призначаються вами, тобто сама вона робити нічого не уміє. Її параметри:

`type="button"` - тип довільної кнопки,

`name` - ім'я кнопки,

`value` - напис на кнопці.

`onclick` - вказує, що робити при натисканні по кнопці. Взагалі, у цього типу кнопок є і інші події (наприклад, `dblclick`), але тут ми їх не розглядатимемо.

Якщо на формі декілька кнопок, то вони повинні мати різні назви.

Приклад коду :

```
<form name="form1">

  <input type="submit" name="submit" value="Відправити">

  <input type="image" name="but_img" src="but.gif">

  <input type="reset" name="reset" value="Очистити">

  <input type="button" name="button" value="Відправити">

</form>
```

## 6. Поле для файлів

Поле для введення імені файлу, супроводжуване кнопкою Browse (Огляд), при клацанні по якій відкривається вікно перегляду дерева тек комп'ютера, де можна вибрати потрібний файл. Вибраний файл приєднується до вмісту форми при відправці на сервер.

Приклад:

```
<form name="form1">

  <input type="file" name="load" size="50">

</form>
```

## 7. Багаторядкове текстове поле

Для об'ємних текстів, наприклад для поштових повідомлень, зручно використовувати саме цей елемент. Він створюється тегами `<textarea>` `</textarea>` і має наступні параметри:

**name** - ім'я поля

**cols** - ширина поля в символах

**rows** - кількість рядків тексту, видимих на екрані

**wrap** - спосіб перенесення слів :

off - не відбувається,

virtual – відображується, але на сервер не ділимий рядок,

physical - перенесення і на екрані і при і на сервер.

**disabled** - неактивне поле

**readonly** - дозволено тільки читання.

Приклад

```
<form name="form1">
  <textarea cols="20" rows="3" wrap="off"></textarea><br>
  <textarea cols="35" rows="5" wrap="virtual"></textarea><br>
  <textarea cols="50" rows="7" wrap="physical"></textarea>

</form>
```

## 8. Списки, що розкриваються

Списки бувають з можливістю вибору одного елементу і з множинним вибором. Задіються і ті, і інші за допомогою тегів `<select>` `</select>`, усередині яких розташовуються елементи значень, заданих тегом **<option>**. Розглянемо атрибути цих тегів :

`<select>`:

**name** - ім'я списку. Кожен вибраний елемент списку при передачі на сервер матиме вигляд: name.value, де значення (value) береться з тега option.

**size** - визначає кількість видимих елементів в списку: 1 - простий список, що розкривається, більше 1 - список із смугою прокрутки.

**multiple** - дозволяє вибір декількох елементів списку.

`<option>`:

**selected** – ним позначають найбільш вірогідний для вибору елемент списку, якщо список з множинним вибором, то можна помітити декілька пунктів.

**value** - значення, яке буде відправлено серверу, якщо пункт вибраний.

Приклад:

```
<form name="form1">
```

Яку мову Ви хочете вивчати:

```
<select name="language" size="1">
<option selected value="html">html
<option value="php">php
<option value="java">java
</select><br><br>
```

Які дні тижня Вас влаштовують:<br>

(вибирайте з натиснутою клавішею ctrl)<br>

```
<select name="day" size="7" multiple>
<option selected value="mon">понеділок
<option value="tue">вівторок
<option value="wen">среда
<option selected value="thu">четвер
<option value="fri">п'ятниця
<option value="sat">субота
<option value="san">неділя
</select>
```

```
</form>
```

Існують ще теги `<optgroup>` `</optgroup>`, що дозволяють групувати елементи списку за якими-небудь ознаками. Наприклад, ми хочемо задати каталог сайтів у вигляді списку, тоді зручніше розбити його на групи по інтересах.

## 9. Надписи

Усі елементи форми можна зв'язати з їх за допомогою елементу **<label>** і його параметра `for`, значенням якого є ім'я елементу, з яким зв'язуємо . Наприклад:

```
<form name="form1">
```

```
<label for="load">Вибирайте файл: </label>
```

```
<input type="file" name="load" size="30">
```

```
</form>
```

## HTML5 Input Types

HTML5 added several new input types:

color

date  
 datetime-local  
 email  
 month  
 number  
 range  
 search  
 tel  
 time  
 url  
 week

[https://www.w3schools.com/html/html\\_form\\_input\\_types.asp](https://www.w3schools.com/html/html_form_input_types.asp)

## Теги для форматування тексту

ТЕГ	ОПИС ТЕГУ	Атрибути тегу
<b>H1&gt;...&lt;H6</b> закриваючий тег обов'язковий	<p>HTML пропонує шість заголовків різного рівня, які показують відносну важливість секції, розташованої після заголовка. Так, тег &lt;H1&gt; представляє собою найбільш важливий заголовок першого рівня, а тег &lt;H6&gt; служить для позначення заголовка шостого рівня і є найменш значним. За замовчуванням, заголовок першого рівня відображається найбільшим шрифтом жирного накреслення, заголовки подальшого рівня за розміром менше. Теги &lt;H1&gt; ... &lt;H6&gt; відносяться до блочних елементів, вони завжди починаються з нового рядка, а після них інші елементи перенесені на наступний рядок. Крім того, перед заголовком і після нього додається порожній простір. Видом заголовком легко можна керувати за допомогою стилів.</p> <p>Заголовки дуже «люблять» пошукові системи, вони підвищують цінність тексту на веб-сторінці, який розташовується всередині тегів &lt;H1&gt; ... &lt;H6&gt;. З цієї причини краще завжди використовувати ці теги, незважаючи на те, що за допомогою стилів будь-який текст можна встановити великого розміру і, тим самим, зробити його заголовком.</p>	<b>ALIGN</b> Для зображень та тексту можна вказувати їх положення щодо тексту або інших зображень на веб-сторінці. Спосіб вирівнювання задається параметром align тега. align='bottom   left   middle   right   top'



<p>&lt;HR&gt; закриваючий тег не обов'язковий</p>	<p>Малює горизонтальну лінію, яка за своїм виглядом залежить від використовуваних параметрів, а також браузера. Тег &lt;HR&gt; відноситься до блокових елементів, лінія завжди починається з нового рядка, а після неї всі елементи відображаються на наступному рядку.</p>	<p>ALIGN WIDTH . Допускається використовувати значення в пікселях або відсотках width = "100%" означає, що малюнок буде розтягнуто на всю ширину веб-сторінки. COLOR Визначає колір тексту на веб-сторінці. SIZE Параметр size задає розмір шрифту в умовних одиницях. NOSHADE Використання параметра noshade скасовує тривимірні ефекти, які використовуються при малюванні лінії за замовчуванням.</p>
<p>&lt;P&gt;</p>	<p>Визначає текстовий абзац. Тег &lt;P&gt; є блоковим елементом, перед ним завжди додається порожній рядок, параграфи тексту йдуть один за одним розділяються між собою вертикальним проміжком. Величиною проміжку можна керувати за допомогою стилів. Якщо закриваючого тега немає, вважається, що кінець параграфа збігається з початком наступного блокового елемента.</p>	<p>ALIGN</p>
<p>&lt;BR&gt; закриваючий тег не обов'язковий</p>	<p>Тег &lt;BR&gt; встановлює новий рядок в тому місці, де цей тег знаходиться. На відміну від тега параграфа &lt;P&gt;, використання тега &lt;BR&gt; не додає порожній відступ перед рядком.</p>	<p>CLEAR &lt;br&gt;clear="all   left   right   none" використовується при обтіканні текстом малюнка</p>
<p>&lt;PRE&gt; закриваючий тег обов'язковий</p>	<p>Як набрали текст, так він і відображається.</p>	<p>немає</p>
<p>&lt;DIV&gt; закриваючий тег обов'язковий</p>	<p>Елемент &lt;DIV&gt; є блоковим елементом і призначений для виділення фрагмента документа з метою зміни виду вмісту.</p>	<p>ALIGN TITLE Додає пояснюючий текст на веб-сторінку у вигляді спливаючої підказки, яка відображається, коли курсор миші затримується у вікні веб-сторінки.</p>

<SPAN> закриваючий тег обов'язковий	Тег <SPAN> призначений для визначення вбудованих елементів документа. На відміну від блокових елементів, таких як <TABLE>, <P> або <DIV>, за допомогою тега <SPAN> можна виділити частину інформації всередині інших тегів і встановити для неї свій стиль.	немає
--	---	-------

## Теги для роботи з текстом

ТЕГ	ОПИС ТЕГУ	Атрибути тегу
<ABBR>	Тег <ABBR> вказує, що послідовність символів є аббревіатурою. За допомогою параметра title дається розшифровка скорочення, що дозволяє розуміти аббревіатуру тим людям, які з нею не знайомі.	TITLE
<ACRONYM>	<b>Not supported in HTML5. Use &lt;abbr&gt; instead.</b>	TITLE
<ADDRESS> закриваючий тег обов'язковий	Тег <ADDRESS> призначений для зберігання інформації про автора і може включати в себе будь-які елементи HTML на кшталт посилань, тексту, виділень і т.д. Планується, що пошукові системи будуть аналізувати вміст цього тегу для збору інформації про авторів сайтів. За замовчуванням текст всередині контейнера <ADDRESS> відображається курсивним накресленням.	немає
<B>	Встановлює жирне накреслення шрифту. Допустимо використовувати цей тег спільно з іншими тегами, які визначають накреслення тексту.	немає
<BASEFONT>: закриваючий тег не обов'язковий	<b>Not supported in HTML5</b> Тег <BASEFONT> призначений для завдання шрифту, розміру і кольору тексту за замовчуванням. Зазначені значення будуть використовуватися у всьому документі за винятком тега <FONT>, в якому можна перевизначити параметри оформлення тексту. Допускається використання тега в контейнері <HEAD> або <BODY>, причому кілька разів. Це дозволяє змінювати вигляд шрифту для частини документа.	COLOR      Визначає колір тексту на веб-сторінці. SIZE      Параметр size задає розмір шрифту в умовних одиницях. FACE      Встановлює сімейство шрифту, яке буде використовуватися для оформлення тексту вмісту. Список шрифтів може включати одне або кілька назв, розділених комою.
<BIG> закриваючий тег обов'язковий	<b>Not supported in HTML5</b> Тег <BIG> збільшує розмір шрифту на одиницю в порівнянні зі звичайним текстом. У HTML розмір шрифту вимірюється в умовних одиницях від 1 до 7, середній розмір тексту, що	немає

	використовується за умовчанням, прийнятий 3. Таким чином, додавання тега <BIG> збільшує текст на одну умовну одиницю. Допускається застосування вкладених тегів <BIG>, при цьому розмір шрифту буде більше з кожним рівнем.	
<BLOCKQUOTE> закриваючий тег обов'язковий	Тег <BLOCKQUOTE> призначений для виділення довгих цитат всередині документа. Текст, позначений цим тегом, традиційно відображається як вирівняний блок з відступами зліва і справа (приблизно по 40 пікселів), а також з відбиттям зверху і знизу.	немає
<CENTER>	<b>Not supported in HTML5</b>	
<CITE> закриваючий тег обов'язковий	Тег <CITE> позначає текст як цитату або виноску на інший матеріал. Таке виділення зручно для зміни стилю тексту через CSS, а також застосовується для розділення коду HTML на структурні елементи. Браузери зазвичай встановлюють текст всередині контейнера <CITE> курсивом. <cite>Текст</cite>	немає
<CODE> закриваючий тег обов'язковий	Тег <CODE> призначений для відображення однієї або декількох рядків тексту, який представляє собою програмний код. Сюди відносяться імена змінних, ключові слова, тексти функцій й т.д. Браузери зазвичай відображають вміст контейнера <CODE> як моноширинний текст зменшеного розміру. На відміну від тега <PRE> додаткові пробіли всередині контейнера <CODE> не враховуються, так само, як і перенесення тексту. Тому використовуйте тег   або <P> для створення переносів.	немає
<DEL> закриваючий тег обов'язковий	Тег <DEL> використовується для виділення тексту, який був вилучений у новій версії документа. Подібне форматування дозволяє відстежити, які зміни в тексті документа були зроблені. Браузери зазвичай позначають текст у контейнері <DEL> як перекреслений.	немає
<DFN> закриваючий тег обов'язковий	Як правило, в документі, коли згадується новий термін, він виділяється курсивом і дається його визначення. При використанні цього терміна в подальшому, він вважається вже відомим читачеві. Тег <DFN> застосовується для виділення таких термінів при їх першій появі в тексті. Браузери відображають вміст контейнера <DFN> за допомогою курсиву.	немає

<EM> закриваючий тег обов'язковий	Тег <EM> призначений для акцентування тексту. Браузери відображають такий текст курсивним накресленням.	немає
<FONT> закриваючий тег обов'язковий	<b>Not supported in HTML5</b> Тег <FONT> представляє собою контейнер для зміни характеристик шрифту, таких як розмір, колір і гарнітура. Хоча цей тег до цих пір підтримується всіма браузерами, він вважається застарілим і від його використання рекомендується відмовитися на користь стилів.	COLOR      Визначає колір тексту на веб-сторінці. FACE Встановлює сімейство шрифту, яке буде використовуватися для оформлення тексту вмісту. Список шрифтів може включати одне або кілька назв, розділених комою. SIZE      Параметр size задає розмір шрифту в умовних одиницях.
<INS> закриваючий тег обов'язковий	Тег <INS> призначений для виділення тексту, який був доданий до нової версії документа. Подібне форматування дозволяє відстежити, які зміни в тексті документа були зроблені. Браузери зазвичай позначають текст у контейнері <INS> як підкреслений.	немає
<KBD> закриваючий тег обов'язковий	Тег <KBD> використовується для позначення тексту, що набирається на клавіатурі, або для назви клавіш. Браузери зазвичай позначають текст у контейнері <KBD> моноширинним шрифтом.	немає
<Q> закриваючий тег обов'язковий	Тег <Q> використовується для виділення у тексті цитат. Вміст контейнера автоматично відображається в браузері в лапках.	немає
<SMALL> закриваючий тег обов'язковий	Тег <SMALL> зменшує розмір шрифту на одиницю в порівнянні зі звичайним текстом. У HTML розмір шрифту вимірюється в умовних одиницях від 1 до 7, середній розмір тексту, що використовується за умовчанням, прийнятий 3. Таким чином, додавання тега <SMALL> зменшує текст на одну умовну одиницю.	немає
<STRONG>	Тег <STRONG> призначений для акцентування тексту. Браузери відображають такий текст жирним шрифтом.	немає
<SUB> закриваючий тег обов'язковий	Відображає шрифт у вигляді нижнього індексу. Текст при цьому розташовується нижче базової лінії інших символів рядка і зменшеного розміру.	немає
<SUP> закриваючий тег обов'язковий	Відображає шрифт у вигляді верхнього індексу. Шрифт при цьому відображається вище базової лінії тексту і зменшеного	немає

	розміру.	
<TT> закриваючий тег обов'язковий	<b>Not supported in HTML5. Use CSS instead.</b> Контейнер <TT> відображає текст моноширинним текстом. Цей тег відноситься до групи тегів фізичного форматування.	немає
<VAR> закриваючий тег обов'язковий	Тег <VAR> використовується для виділення змінних комп'ютерних програм. Браузери зазвичай позначають текст у контейнері <VAR> курсивним накресленням.	немає
<U> закриваючий тег обов'язковий	Додає підкреслення до тексту. Цей тег засуджується специфікацією HTML, натомість рекомендується використовувати стилі.	немає
<S> закриваючий тег обов'язковий	Тег <S> відображає текст як перекреслений. Цей тег аналогічний тегу <STRIKE>, але на відміну від нього має скорочену форму запису подібно тегам <B>, <I> і <U>. У будь-якому випадку, теги <S> і <STRIKE> засуджуються специфікацією HTML, замість них рекомендується використовувати стилі.	немає