

## **Текст в HTML**

План:

1. Особливості тексту в HTML
2. Оформлення абзаців, заголовків
3. Вирівнювання тексту, накреслення
4. Верхній і нижній індекси, аббревіатури
5. Спецсимволи, посилання

### **Особливості тексту в HTML**

Перш ніж редагувати код веб-сторінки, важливо врахувати деякі особливості, пов'язані з роботою з текстом у HTML.

HTML не використовує розстановку переносів в тексті за замовчуванням, тому для перенесення тексту до наступного рядка необхідно використовувати тег `<br>`.

Також, HTML не робить різниці між одним і багатьма пропусками, і відображає їх як один пропуск. Це правило також стосується символів табуляції.

Однак, існує виняток з цього правила, коли мова йде про тег `<pre>`. Цей тег використовується для форматування тексту з урахуванням пробілів та символів табуляції. Всередині тегу `<pre>` будь-яка кількість пропусків та символів табуляції відображається точно так, як вона зазначена в коді.

HTML не підтримує автоматичну розстановку переносів у словах, як це можна зробити в текстових редакторах. Це не є суттєвим до тих пір, поки не використовується вирівнювання тексту по ширині.

В такому випадку блок тексту вирівнюється по лівому та правому краю, а короткі рядки можуть розтягуватися за рахунок додавання пробілів між словами.

Проте, іноді занадто багато пробілів між словами може погіршувати зовнішній вигляд сторінки та читабельність тексту.

HTML текст за замовчуванням займає ширину вікна браузера. Якщо ви напишете один довгий рядок в HTML коді, браузер автоматично вирівняє його по ширині вікна і, якщо можливо, додаст переноси в місцях пробілів або дефісів.

Однак, якщо в тексті немає пробілів або дефісів, браузер не зможе автоматично створити переноси і відобразить текст як один довгий рядок.

Якщо текст ширший за вікно браузера, з'явиться горизонтальна смуга прокрутки, що може погіршити зовнішній вигляд сторінки і ускладнити читання тексту.

Тому рекомендується використовувати пробіли та дефіси в тексті для поліпшення його читабельності та зовнішнього вигляду.

### **Оформлення абзаців, заголовків**

В HTML для оформлення абзаців використовують тег <p>. Цей тег означає "параграф" і використовується для виділення окремих блоків

тексту, що мають свій власний смисловий зміст і зазвичай розділяються від інших блоків тексту вертикальним проміжком.

Наприклад, щоб створити абзац з текстом "Це перший абзац. Це другий абзац.", потрібно використати наступний код:

```
<p>Це перший абзац.</p>
```

```
<p>Це другий абзац.</p>
```

При використанні тега <p> між абзацами створюються відступи, які можуть бути занадто великими. Але їх можна позбутися, якщо використати тег перенесення рядка <br> замість перенесення на новий рядок за допомогою тегу <p>.

Тег <br> не створює додаткових вертикальних відступів між рядками і може застосовуватися практично в будь-якому тексті для вирівнювання вмісту по ширині сторінки.

В HTML для оформлення заголовків використовуються теги <h1> до <h6>. Ці теги вказують на рівень важливості заголовку, де <h1> має найбільшу важливість, а <h6> – найменшу.

## Заголовок першого рівня

### Заголовок другого рівня

#### Заголовок третього рівня

#### Заголовок четвертого рівня

#### Заголовок п'ятого рівня

#### Заголовок шостого рівня

Заголовки використовуються для позначення розділів або підрозділів на веб-сторінці. Вони зазвичай відображаються більшим шрифтом, болдовано, можуть мати інший колір або стиль.

Зазвичай на веб-сторінках використовують заголовки з першого по третій рівні, оскільки вони забезпечують достатню ієрархію заголовків на сторінці. Для більш складних структур, які вимагають використання заголовків нижчого рівня, можна застосовувати їх з обмеженням.

Наприклад, на сторінці можна мати декілька заголовків четвертого рівня, які використовуються для виділення важливих блоків тексту або відокремлення малих розділів внутрішньої структури сторінки.

Однак варто пам'ятати, що надмірне використання заголовків нижчого рівня може призвести до змішування читача і зниження зрозумілості структури сторінки.

### **Вирівнювання тексту, накреслення**

Вирівнювання тексту в HTML визначає його зовнішній вигляд та може бути виконано за допомогою атрибута `style` або за допомогою CSS. Такі параметри, як `text-align`, можуть бути встановлені для вирівнювання тексту зліва, справа, по центру або по ширині.

Зазвичай, найбільш поширеним варіантом є вирівнювання тексту по лівому краю, де текст зліва зсувається до краю, а правий залишається нерівним.

Вирівнювання по правому краю і по центру, в основному, використовується в заголовках і підзаголовках. Проте, при використанні вирівнювання по ширині текст між словами може мати великі інтервали, що не завжди є естетичним.

Аналогічним для text-align, є атрибут align, який може приймати такі значення:

Назва	Функція
left	вирівнює елемент по лівому краю.
right	вирівнює елемент по правому краю.
center	вирівнює елемент по центру.
justify	вирівнює текст по ширині, рівномірно розподіляючи пробіли між словами в рядку.

Значення атрибуту align можна встановити для таких елементів HTML, як <p>, <img>, <table>, <div> та інші. Однак, використання атрибуту align зазвичай не рекомендується через те, що він застарілий і замінюється більш сучасними методами вирівнювання за допомогою CSS.

Насиченість шрифту визначається збільшенням товщини ліній і відповідним збільшенням контрастності. Зазвичай розрізняють чотири види насиченості: світле, нормальне, напівжирне і жирне.

Проте, у HTML можна використовувати тільки теги для нормального і жирного накреслення. Для жирного накреслення використовуються теги <b> і <strong>, які встановлюють текст у жирному шрифті: <b>Це текст у

жирному шрифті</b> <strong>Цей текст також виділений жирним шрифтом</strong>

Курсивний шрифт використовується для виділення тексту і може імітувати рукописний стиль. Для курсиву в HTML використовуються теги <i> і <em>: <i>Цей текст в курсиві</i> <em>Цей текст також виділений курсивом</em>

Теги <b> і <strong> обидва використовуються для візуального виділення тексту жирним шрифтом, однак між ними є різниця в їх семантиці.

Тег <b> використовується для стилізації тексту з метою зробити його більш видимим або відзначити важливу інформацію, але не надає йому особливого значення з точки зору структури документа.

Тег <strong> також змінює вигляд тексту на жирний, але при цьому передає його семантику, підкреслюючи те, що текст є важливим з точки зору контексту документа, так що це може вплинути на його інтерпретацію.

Теги <i> і <em> також відображають текст курсивом, але знову ж таки, є різниця в їх семантиці.

Тег <i> використовується для стилізації тексту з метою виділити його від іншого тексту, але не передає йому особливого значення з точки зору структури документа.

Тег <em> також відображає текст курсивом, але зазначає, що текст є важливим з точки зору контексту, підкреслює акцент на слово або словосполучення, яке може вплинути на інтерпретацію тексту документа.

Загалом, теги `<strong>` і `<em>` передають семантику тексту, тоді як теги `<b>` і `<i>` використовуються виключно для візуального відображення тексту з відповідним стилем.

### **Верхній і нижній індекси, аббревіатури**

Верхній і нижній індекси – це символи, які використовуються для відображення чисел, літер або символів, розміщених нижче або вище звичайного рівня тексту. В HTML є теги для створення верхніх і нижніх індексів – `<sup>` і `<sub>` відповідно.

Тег `<sup>` використовується для створення верхнього індексу, наприклад, для позначення показника степеня в математичних формулах. Цей тег відображає текст вище звичайного рівня тексту і зазвичай він має менший розмір шрифту.

Тег `<sub>` використовується для створення нижнього індексу, наприклад, для позначення хімічних формул. Цей тег відображає текст нижче звичайного рівня тексту і зазвичай має менший розмір шрифту.

Також варто зазначити, що можна використовувати і комбіновані верхні/нижні індекси, що можуть бути корисними, наприклад, у відображенні математичних формул або хімічних реакцій

В HTML аббревіатури можна позначати за допомогою тегу `<abbr>`. Цей тег використовується для створення аббревіатур, які при наведенні миші на них, можуть відображати повну форму слова або фрази в спливаючому вікні.

Тег <abbr> має два обов'язкових атрибути:

title: вказує повну форму або значення аббревіатури, яка буде відображатися в спливаючому вікні при наведенні на аббревіатуру мишою;

aria-label: вказує текст, який буде прочитаний синтезатором мови, якщо сторінка відкривається за допомогою програми для читання екрану.

Акронім – це стале скорочення, яке вимовляється як слово і є самостійним поняттям. Елемент ACRONYM в HTML вказує на те, що текст є акронімом. Він використовується для створення посилань на визначення акронімів або скорочень, які можуть бути незрозумілі для багатьох читачів.

Наприклад, для скорочень, які використовуються в спеціалізованих областях знань, таких як наука, техніка, медицина, тощо. Зараз рекомендовано використовувати замість елементу ACRONYM атрибут title у тезі. Наприклад, <abbr title="соціальне страхування">Соцстрах</abbr>.

### **Спецсимволи, посилання**

У HTML є спеціальний підмножина символів, яка не може бути введена на клавіатурі за допомогою звичайних клавіш. Ці символи використовуються для відображення різних елементів, таких як математичні формули, фрагменти програмного коду, музичні ноти, емодзі та ін.

Для вставки спецсимволів в HTML використовується спеціальний код символу (наприклад, , ім'я: &nbsp; код: &#160; – нерозривний пробіл, ім'я: &euro; код: &#8364;; – знак євро.)



Посилання (англ. `hyperlink`) в HTML – це елемент, який дозволяє створювати зв'язок між різними сторінками в Інтернеті. Посилання дають можливість користувачам переходити з однієї веб-сторінки на іншу, з одного розділу на інший, зображення, відео або іншого елементу до відповідної веб-сторінки.

Головною метою посилань є забезпечення зручності навігації по веб-сторінках і покращення користувацького досвіду. Посилання можуть використовуватися для посилання на інформацію, звіти, новини, фотографії, відео, аудіо, форми зворотного зв'язку тощо.

Синтаксис посилання починається з тегу `<a>`, після якого слідує атрибут `href`, в якому вказується адреса сторінки, на яку буде переходити користувач.

Текст, що відображається на веб-сторінці, і який буде посиланням, поміщається між відкриваючим і закриваючим тегом `<a>`.

Для створення посилання на документ, який знаходиться на тому ж сервері, що і веб-сторінка, на якій відображається посилання, у атрибуті `href` можна вказати шлях до цього документу відносно кореневої папки веб-сайту.

Для створення посилання на зовнішній ресурс (інші веб-сайти, файли, інтернет-ресурси тощо), у атрибуті `href` необхідно вказати повну адресу ресурсу, з яким потрібно створити посилання.

При потребі, до тексту посилання можна додати класи і ідентифікатори, що дає можливість налаштувати його вигляд і поведінку за допомогою CSS і JavaScript.

Будь-яке посилання на веб-сторінці може перебувати в одному з наступних п'яти станів:

Стан	Опис
:link	Стан посилання, яке ще не було відвідане. Посилання зазвичай відображається в блакитному кольорі за замовчуванням.
:visited	Стан посилання, яке було відвідане. Посилання зазвичай відображається в фіолетовому кольорі за замовчуванням.
:hover	Стан посилання, на яке наведено курсор миші. Цей стан зазвичай використовується для зміни кольору посилання при наведенні на нього курсора.
:active	Стан посилання, на яке було клікнуто. Цей стан зазвичай використовується для зміни стилю посилання при натисканні на нього мишкою.
:focus	Стан посилання, на яке було переведено фокус клавіатури. Цей стан зазвичай використовується для зміни стилю посилання при переході до нього за допомогою клавіші Tab або іншими засобами навігації з клавіатури.

Окрім, href посилання може мати і інші атрибути:

Атрибут	Опис
href	Вказує URL, на який вказує посилання
target	Вказує, де відкривати URL, вказаний в атрибуті href (наприклад, в тому ж вікні, в новому вікні, в iframe тощо)
rel	Вказує стосунок між поточною сторінкою та сторінкою, на яку вказує посилання (наприклад, rel="nofollow" вказує на те, що посилання не повинно підвищувати рейтинг сторінки в пошукових системах)

title	Вказує додаткову інформацію про посилання, яка відображається при наведенні курсору на посилання
download	Вказує, що посилання призначене для завантаження файлу замість переходу на сторінку
media	Вказує, яке середовище повинно відображати посилання (наприклад, на екранах з високою роздільною здатністю можуть відображатися інші медіафайли)
type	Вказує MIME-тип файлу, на який вказує посилання
blank	вказує на те, що посилання буде відкрито в новому вікні або вкладці браузера.
self	вказує на те, що посилання буде відкрито в тій же області, де воно було натиснуте.
parent	вказує на те, що посилання буде відкрито в батьківському вікні, якщо воно було вбудоване в фрейм (frame) або вкладку (tab).
top	вказує на те, що посилання буде відкрито в верхній області вікна браузера, замінюючи всі фрейми і вкладки.

Якоря в HTML – це спеціальні теги, які використовуються для створення посилань на певну частину сторінки. Їхня мета полягає в тому, щоб дозволити користувачеві легко перейти до конкретної секції веб-сторінки, яка може містити багато різного контенту.

Особливість якорів полягає в тому, що вони не переадресовують користувача на іншу сторінку, а лише переміщують його в межах поточної сторінки до вказаного розділу або елемента.

Це дуже зручно для навігації на сторінці з багатою інформацією, такої як довгий список товарів або статей.

Основні атрибути якорів в HTML:

Атрибу т	Опис
-------------	------

href	Вказує URL-адресу документу, на який повинен перейти користувач після кліку по якорю
name/ id	Встановлює унікальне ім'я або ідентифікатор якоря, яке використовується для посилання на нього з інших частин документу
target	Вказує, де має відкриватись посилання, яке веде на якір (в поточному вікні, новому вікні, в тому ж фреймі або в батьківському фреймі)
title	Встановлює підказку, яка відображається при наведенні курсору миші на якір
tabindex	Встановлює порядковий номер, який визначає послідовність фокусування на клавішній панелі, коли використовується клавіша Tab
rel	Встановлює відносини між поточним документом і документом, на який вказує посилання
class	Встановлює ім'я класу для стилізації якоря за допомогою CSS
style	Встановлює стилі для якоря за допомогою CSS

Якорів в HTML існує два типи: внутрішні якорі та зовнішні якорі.

Внутрішні якорі – це якорі, які вказують на елемент або розділ на тій же сторінці, на якій вони знаходяться.

Їх створення складається з двох частин: перша частина – це створення самого якоря з використанням атрибуту id, а друга частина – це створення посилання на цей якір з використанням спеціального синтаксису #id.

Зовнішні якорі – це якорі, які вказують на елемент або розділ на іншій сторінці. Вони створюються шляхом додавання додаткового інформаційного фрагменту до URL-адреси, що вказує на цільовий елемент на сторінці, до якої веде посилання.

Якщо на веб-сторінці є посилання на якір, але самого якоря на сторінці немає, то посилання буде неактивним, тобто при кліку на нього нічого не відбудеться. Тому рекомендується перевіряти, чи існує якір на сторінці перед створенням посилання на нього.