МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
“Ужгородський національний університет”
Математичний факультет
Кафедра системного аналізу і теорії оптимізації

ВСТУП ДО WEB-ПРОГРАМУВАННЯ. ОСНОВИ HTML

Методичні вказівки для самостійної роботи студентів математичного факультету з дисципліни «Вступ до web-програмування»

Ужгород – 2018

Брила А.Ю., Глебена М.І., Ломага М.М., Млавець Ю.Ю. Методичні вказівки для самостійної роботи студентів математичного факультету з дисципліни «Вступ до web-програмування». Вступ до web-програмування. Основи HTML.– Ужгород, 2018 . – 72 с.

Методичні вказівки призначені для студентів спеціальностей 113 Прикладна математика та 124 Системний аналіз та можуть бути використані як комплекс рекомендацій навчально-методичного характеру з питань організації самостійної роботи та для успішного засвоєння студентами сучасних web технологій і суміжних галузей знань, вивчення та практичного використання методів і засобів створення web-сайтів.

Рекомендовано до друку Вченою радою математичного факультету

ДВНЗ "Ужгородський національний університет"

від 14 червня 2018 року, протокол № 11.

Вступ

Спеціаліста будь-якого фаху в сучасних умовах неможливо уявити без уміння використання комп'ютерної техніки. Особливо це стосується спеціалістів, чия діяльність безпосередньо пов'язана з використанням ЕОМ. Створення сучасних програмних продуктів передбачає використання мережі інтернет, а отже, сучасний додаток повинен бути web- додатком.

В методичних вказівках розкриваються технології HTML в плані вирішення різних аспектів створення web-сторінок. Простий візуальний спосіб подачі інформації зі зрозумілими прикладами і невеликим фрагментом коду допоможе читачам освоїти HTML та за допомогою засобів цієї мови навчитись розробляти сучасні ефективні web-додатки.

Наведено можливості і засоби по оформленню тексту і зміни його виду. Висвітлено питання роботи з малюнками, посиланнями, списками, лініями і рамками. Розкрито можливості управління видом таблиці, використання шаблонів. Розглянуто елементи форм, їх параметри і приклади зміни оформлення за допомогою кольору, зображень і рамок. Наведено способи вирівнювання малюнків, тексту, використання відступів і полів і багато інших прийомів оформлення web-сторінок.

Методичні вказівки будуть корисними студентам спеціальностей 113 Прикладна математика та 124 Системний аналіз при вивченні дисципліни «Вступ до web-програмування».

Мова розмітки гіпертексту HTML

HTML (від англ. *HyperText Markup Language* − **мова розмітки гіпертексту**) − стандартна мова розмітки веб-сторінок. Документ HTML оброблюється браузером та відтворюється на екрані у звичному для людини вигляді.

HTML надає засоби для:

- створення структурованого документу шляхом позначення структурного складу тексту: заголовки, абзаци, списки, таблиці, цитати та інше;
- створення зв'язків між веб-сторінками через гіперпосилання;
- створення інтерактивних форм;
- включення зображень, звуку, відео та ін.

Розмітка HTML

Весь контент описується з використанням спеціальних правил – правил розмітки. Розмітка в HTML складається з чотирьох основних компонентів:

- елементів;
- базових типів даних (визначають тип вмісту атрибутів: символьні дані, числа та ін.);
- символьних мнемонік;
- декларації типу документа (оголошує версію HTML).

Мнемоніки – спеціальні позначення (псевдоніми), які використовуються для вставки деяких символів (наприклад символів, яких немає у даній таблиці кодування).

Символ	Призначення	Сутнісне позначення	Десяткове позначення	Шістнадцяткове позначення
"	машинні подвійні лапки	"	"	"
&	амперсанд	&	&	&
'	апостроф	'	'	'
<	знак менше	<	<	<
>	знак більше	>	>	>

Елементи HTML

Елементи являють собою базові компоненти розмітки HTML. Кожен елемент має назву і задається за допомогою тегів, що записуються у кутових дужках.

Призначення	Загальна форма	Приклад
Відкриваючий тег	< **назва елемента** >	
Закриваючий тег	</ **назва елемента** >	

Вміст (контент) елемента записують між відкриваючим і закриваючим тегами.

Загальна форма	< назва елемента > **Вміст** </ назва елемента >
Приклад	**Lorem ipsum**

Також елементи можуть мати атрибути (властивості), які задаються всередині відкриваючого тегу. Значення атрибута записують у подвійних лапках.

Загальна форма	< назва елемента **атрибут** = “ **значення** ” >
Приклад	< div **class="block1**">Lorem ipsum

Приклад.

Деякі елементи, можуть не мати атрибутів, інші ж мають обов'язкові атрибути, без яких вони не зможуть виконувати свої функції.

Існують також елементи, які не мають вмісту (контенту) і задаються за допомогою одиночних тегів

Призначення	Загальна форма	Приклад
За правилами HTML	< **назва елемента** >	** **
За правилами XHTML	< **назва елемента** />	**< br/ >**

Категорії елементів

За своїм призначенням та дією елементи HTML можна розділити на три великі категорії: структурні, презентаційні та елементи для інтерактивної взаємодії з користувачем.

Структурні. Елементи структурної розмітки застосовуються задля опису семантики тексту та задання структури документу. В свою чергу, в залежності від змісту документа, структурні елементи поділяються на: текст, списки, таблиці, гіперпосилання та додаткові об'єкти (зображення, аплети та ін). Вони не зазначають ніякого спеціального (візуального) відтворення тексту, проте більшість браузерів мають стандартні стилі форматування для кожного елемента. Для подальшого стилізування тексту рекомендується використовувати Каскадні таблиці стилів (CSS).

Презентаційні. Елементи презентаційної (візуальної) розмітки застосовуються задля опису зовнішнього вигляду документа та не зазначають при цьому функцій тексту, до якого вони застосовуються. Більшість з цих елементів є такими, що не рекомендується застосовувати у розмітці. В залежності від свого змісту, елементи презентаційної розмітки поділяють на: елементи додавання каскадних таблиць стилів, візуальні елементи тексту (шрифти, кольори, насиченість та нахил шрифту, лінії та інші) та фрейми.

Інтерактивної взаємодії. Елементи інтерактивної взаємодії дозволяють користувачам впливати на вміст або вигляд документу. Вони поділяються на два види: форми для введення данних користувачами та скрипти для активних документів.

Основні елементи

У процесі свого розвитку було запропоновано багато різних версій мови розмітки HTML і крім того, є XHTML (EXtensible HyperText Markup Language, розширена мова розмітки гіпертексту). Вона схожа на HTML, але розрізняється від нього за синтаксисом. Елемент **<!DOCTYPE>** призначений для вказівки типу поточного документа – DTD (document type definition, опис типу документа). Це необхідно, щоб браузер розумів, як слід інтерпретувати поточну веб-сторінку (згідно якого стандарту).

HTML 4	
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">	Строгий синтаксис HTML.
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">	Перехідний синтаксис HTML.
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">	У документі застосовуються фрейми.
HTML 5	
<!DOCTYPE html>	У цій версії тільки один тип.
XHTML	
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">	Вважається, що XHTML витіснить з часом HTML. Ніякого поділу на типи немає.

Надалі будемо розглядати одну із найбільш популярних на даний час версій HTML 5.

Загальна структура HTML-документа

Призначення	Приклад
Декларація типу документа	<!doctype HTML>
Початок документу	<html>
Заголовок	<head> <title> First HTML</title> </head>
Тіло (контент)	<body> Hello world! </body>
Кінець документу	</html>

Документ починається із декларації типу, яка містить опис версії HTML. При використанні версії HTML 5, номер версії можна не вказувати.

Елемент **<html>** є контейнером, який містить в собі весь вміст сторінки. Як правило, **<html>** йде в документі другим, після <**!DOCTYPE**>. Закриваючий тег **</html>** повинен завжди стояти в документі останнім. Все, що йде після нього – ігнорується.

Заголовок

Заголовок документу може містити загальні технічні відомості (текст заголовку, підключення зовнішніх стильових файлів, скриптів та ін.) та додаткову інформацію, яка не відображається та може використовуватися під час пошуку.

Елемент **<head>** призначений для опису заголовку і може містити загальні технічні відомості та додаткову інформацію, мета якої – допомогти браузеру в роботі з даними. Також всередині контейнера <**head**> знаходяться метатеги, які використовуються для зберігання інформації, призначеної для браузерів і пошукових систем. Наприклад, механізми пошукових систем звертаються до метатегів для отримання опису сайту, ключових слів і інших даних.

Вміст <**head**> не відображається безпосередньо на веб-сторінці, за винятком <**title**> . Всередині контейнера <**head**> допускається розміщувати такі елементи: <**base**>, <**basefont**>, <**bgsound**>, <**link**>, <**meta**>, <**script**>, <**style**>, <**title**>.

Елемент **<title>** визначає заголовок документа. Він не є частиною документа і не показується прямо на сторінці. Допускається використовувати тільки один тег <**title**> на документ і розміщувати його тільки в контейнері <**head**>.

Елемент **<meta>** визначає метатеги для зберігання інформації, призначеної для браузерів і пошукових систем. Наприклад, механізми пошукових систем звертаються до метатегів для отримання опису сайту, ключових слів і інших даних. Дозволяється використовувати більше ніж один метатег, всі вони розміщуються в контейнері <**head**>.

Елемент <**meta**/> не має закриваючого тега.

Основні атрибути:

Атрибут	Призначення
charset	Задає кодування документа
name	Встановлює ідентифікатор метатега для пари «ім'я = значення»

Часто використовуються наступні значення для атрибута **name**:

Значення	Зміст
"author"	Ім'я автора документа
"description"	Опис поточного документа
"keywords"	Список ключових слів, що зустрічаються на сторінці
"content"	встановлює значення атрибуту, заданого за допомогою name

Приклад.

```
<meta name="keywords" content="HTML, META, метатег, ключові слова">
<meta charset="utf-8">
<meta name="author" content="Іванов І.І.">
```

Приклад.

Призначення	Приклад
	<!doctype html> <html>
Початок заголовку	**<head>**
Текст заголовку	**<title> First HTML</title>**
Короткий опис	**<meta name ="description" content="Creating the first HTML document">**
Ключові слова	**<meta name ="keywords" content="studying,HTML,markup text">**
Підключення стильового файлу	**<link rel="stylesheet" href="…"/>**
Підключення файлів скрипта	**<script src="…"></script>**
	. . . . .
Кінець заголовку	**</head>**
	<body> . . . </body> </html>

Тіло документа

Елемент **<body>** призначений для зберігання змісту сторінки (контенту), що відображається у вікні браузера. Всю інформацію в документі слід розташовувати саме всередині контейнера <**body**>. До такої інформації відноситься текст, зображення, теги, скрипти і т.д.

Форматування тексту

Деякі особливості форматування тексту

Перш ніж редагувати код веб-сторінки, слід взяти до уваги деякі особливості, які властиві HTML при роботі з текстом.

- *Значення пробільних символів.* Будь-яка кількість пробілів що йдуть підряд, в браузері відображається як один. Скільки б ви не поставили пробілів між словами, це ніяк не вплине на кінцевий вигляд тексту. Це ж правило відноситься до символів табуляції і переходу на новий рядок.
- *Переноси у словах.* HTML не підтримує розстановку переносів у словах, як це роблять текстові редактори, всі слова пишуться без розбиття. Ця умова є несуттєвою, поки не використовується вирівнювання тексту по ширині. У цьому випадку блок тексту вирівнюється по лівому і правому краю. Короткі рядки при цьому розтягуються за рахунок автоматичного додавання пробілів між словами. Іноді порожні блоки між словами настільки великі, що псують зовнішній вигляд сторінки і погіршують читабельність тексту.
- *Переноси тексту.* Текст займає ширину вікна браузера. Якщо ви просто напишете один довгий рядок в коді HTML, то в браузері він буде відформатований так, щоб текст помістився по ширині вікна. Переноси тексту будуть додані автоматично в місцях пробілу або дефіса. Що відбудеться, якщо в тексті немає ні того, ні іншого символу? Браузер не зможе створити переноси і відобразить текст одним рядком. Якщо він ширше вікна браузера, то з'явиться горизонтальна смуга прокручування.

Блочні та рядкові елементи форматування тексту в HTML

Блочним називається елемент, який відображається на веб-сторінці у вигляді прямокутника (вміст знаходиться у прямокутній області). Такий елемент займає всю доступну ширину, висота визначається його вмістом, і він завжди починається з нового рядка. За допомогою блочних елементів формують структуру документа (абзаци, заголовки та ін.). В той же час елементи рядка призначені для форматування певної частини в середині рядка (абревіатури, вбудовані цитати та ін.).

Блочні елементи форматування тексту в HTML

Стандартні параграфи

Параграф є будівельним блоком для більшості документів. В HTML параграфи представляють елементом **p**.

Приклад	На екрані
** **Це дуже короткий параграф. Він містить тільки два речення ** ** ** **А це ще один дуже короткий параграф. ** **	Це дуже короткий параграф. Він містить тільки два речення А це ще один дуже короткий параграф.

Атрибут **align** задає вирівнювання тексту (**left, right, center, justify**). За замовчуванням – **left.**

Перехід на новий рядок

Тег <**br**> (від англ. break) створює перехід на новий рядок. У HTML тег <**br**> слід використовувати в текстовому блоці для позначення нового рядка, де речення (фраза, думка) не закінчене, наприклад у віршах.

На відміну від елемента <**p**>, використання <**br**> не додає порожній відступ перед рядком.

Приклад	На екрані
Реве та стогне Дніпр широкий, ** ** Сердитий вітер завива**, ** Додолу верби гне високі**, ** Горами хвилю підійма**. **	Реве та стогне Дніпр широкий, Сердитий вітер завива, Додолу верби гне високі, Горами хвилю підійма.

Розділення горизонтальною лінією

Теґ <**hr**> візуально виглядає як горизонтальна лінія, але її вигляд залежить від використовуваних параметрів та браузера. Тег <**hr**> відноситься до блокових елементів, лінія завжди починається з нового рядка, а всі елементи після неї відображаються в наступному рядку.

Приклад	На екрані
Реве та стогне Дніпр широкий, Сердитий вітер завива, **<hr>** Додолу верби гне високі, Горами хвилю підійма. **<hr>**	Реве та стогне Дніпр широкий, Сердитий вітер завива, --- Додолу верби гне високі, Горами хвилю підійма. ---

Текст зі збереженням вихідного форматування

Тег <**pre**> визначає блок попередньо форматованого тексту. Текст в елементі <**pre**> відображається шрифтом фіксованої ширини, і він зберігає як пропуски так і розриви рядків.

За замовчуванням, будь-яка кількість пропусків поспіль, на веб-сторінці відображається як один. Тег <**pre**> дозволяє обійти цю особливість і відображати текст в його оригінальному форматуванні.

Всередині контейнера <**pre**> допустимо застосовувати будь-які теги крім: <**big**>, <**img**>, <**object**>, <**small**>, <**sub**> і <**sup**>.

Приклад	На екрані

<pre>
Реве та стогне Дніпр широкий,
Сердитий вітер завива,
Додолу верби гне високі,
Горами хвилю підійма.
</pre>
	Реве та стогне Дніпр широкий,
	Сердитий вітер завива,
	Додолу верби гне високі,
	Горами хвилю підійма.

Заголовки розділів сторінки

Коли сторінка розбита на логічні розділи, кожний розділ повинен вводитися відповідним заголовком. HTML визначає шість рівнів заголовків: h1, h2, h3, h4, h5 і h6 (від найбільш важливого до найменш важливого). h1 буде основним

заголовком всієї сторінки, h2 використовується потім для розбивки сторінки на розділи, h3 на підрозділи і т.д. Важливо використовувати заголовки для опису документа в термінах розділів, підрозділів, під-підрозділів, оскільки це робить документ більш зрозумілим.

Теги <h1> - <h6> відносяться до блокових елементів, вони завжди починаються з нового рядка, а після них інші елементи відображаються у наступному рядку. Крім того, перед заголовком і після нього додається порожній простір.

Приклад	На екрані
** 1Продукти харчування ** ** 1.1.Овочі ** **Капуста, помідори, огірки** ** 1.2.Фрукти ** ** 1.2.1.Кісточкові ** **Вишні, абрикоси** ** 1.2.2.Безкісточкові ** **Полуниця, малина**	**1Продукти харчування** **1.1.Овочі** Капуста, помідори, огірки **1.2.Фрукти** **1.2.1.Кісточкові** Вишні, абрикоси **1.2.2.Безкісточкові** Полуниця, малина

Цитування

Багаторядкове цитування

Тег <**blockquote**> визначає довгий (багаторядковий) текст, який процитований з іншого джерела. Текст, позначений тегом <**blockquote**>, відображається як блок з відступами зліва та справа та з відступами зверху та знизу.

Атрибутом **cite** задається посилання на джерело, з якого було процитовано текст.

Приклад	На екрані
<blockquote cite="http://pro-status.com.ua"> Іноді доводитися оминути цілий світ, щоб дізнатися, що скарб був закопаний у порога власного будинку. **</blockquote>**	Іноді доводитися оминути цілий світ, щоб дізнатися, що скарб був закопаний у порога власного будинку.

Не потрібно використовувати атрибут **cite** у тому випадку, коли цитата береться із книги, журналу, або іншої форми немережевого вмісту.

Короткі цитати

Тег **<q>** визначає коротку цитату. Браузери, зазвичай, показують цитату в лапках.

Приклад	На екрані
<**q**>Терпи, терпи — терпець тебе шліфує.**</q>** - Василь Стус	“Терпи, терпи — терпець тебе шліфує.” - Василь Стус

Цитування авторів

Тег <**cite**> визначає назву твору (наприклад, книги, пісні, кіно, ТБ-шоу, картини, скульптури і т.д.). Це може бути твір, на який посилаються в цитаті, або просто робота, яка згадується побіжно. Браузери зазвичай виділяють текст всередині <**cite**> курсивом.

Приклад	На екрані
<q>Вірш у виконанні **<cite>**Ліни Костенко**</cite>**</q>	“Вірш у виконанні *Ліни Костенко*”

Абревіатури

Тег <**abbr**> визначає абревіатуру або акронім. Пошукові системи індексують повнотекстовий варіант скорочення, що може використовуватися для підвищення рейтингу документа.

Глобальний атрибут **title** може бути використаний з тегом <**abbr**>, щоб побачити повну версію скорочення при наведенні курсора миші на тег <**abbr**>.

Приклад	На екрані
Після включення **<abbr title=** "Електронної обчислювальної машини">ЕОМ**</abbr>** здійснюється перевірка пристроїв.</p>	Після включення ЕОМ здійснюється перевірка пристроїв. Електронної обчислювальної машини

Інвертування тексту

Тег <**bdo**> використовується для заміни поточного напрямку тексту і переважно призначений для використання з мовами, де читання відбувається справа наліво. Наприклад, арабська мова.

Атрибут **dir** визначає напрямок відображення тексту - зліва направо або справа наліво (**ltr** – звичайний текст , **rtl** – інвертований (справа-наліво).

Приклад	На екрані
<bdo dir="rtl"> This text will go right-to-left. **</bdo>**	.tfel-ot-thgir og lliw txet sihT

Оформлення означень

Зазвичай, в документі, коли згадується новий термін, він виділяється курсивом і дається його визначення. При використанні цього терміна в подальшому, він вважається вже відомим читачеві. Елемент <**dfn**> застосовується для визначення таких термінів при їх першій появі в тексті. Визначити термін можна декількома способами:

- за допомогою атрибута **title** у елемента <**dfn**>;

- за допомогою атрибута **title** у вкладеного елемента <**abbr**>;
- текстом всередині <**dfn**>.

Приклад	На екрані
 <dfn>HTML**</dfn>** - це мова розмітки гіпертексту. **<dfn title**="HyperText Markup Language">HTML**</dfn>** - це мова розмітки гіпертексту **<dfn><abbr title**="HyperText Markup Language">HTML**</abbr></dfn>** - це мова розмітки гіпертексту 	*HTML* - це мова розмітки гіпертексту. *HTML* - це мова розмітки гіпертексту *HTML* - це мова розмітки гіпертексту HyperText Markup Language

Теги для виведення комп'ютерного тексту

Тег <**code**> визначає текст, який являє собою програмний код. Сюди відносяться імена змінних, ключові слова, функції і 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 переноси тексту.

Приклад	**<code>**
	function n(){var a=p;if(!a||"none"==a.style.display)return!1;
	if(document.defaultView&&document.defaultView.getComputedStyle)
	{var c=document.defaultView.getComputedStyle(a);
	</code>

На екрані	`function n(){var a=p;if(!a||"none"==a.style.display)return!1;`
	`if(document.defaultView&&document.defaultView.getComputedStyle`
	`c=document.defaultView.getComputedStyle(a);`

Оформлення тексту - комбінацій клавіш та команд

Тег <**kbd**> позначає текст, що може бути введений з клавіатури (наприклад, консольна команда) або комбінації клавіш, що можуть бути натиснені. Браузери, зазвичай, позначають текст у контейнері <**kbd**> моноширинним шрифтом.

Приклад	Введіть в командний рядок слово **<kbd>back</kbd>**, або натисніть клавішу **<kbd>Backspace</kbd>** на клавіатурі.
На екрані	Введіть в командний рядок слово back, або натисніть клавішу Backspace на клавіатурі.

Оформлення тексту – результату роботи програми

Тег <**samp**> позначає текст як результат виведення комп'ютерної програми або скрипта. Браузери зазвичай відображають текст взятий у контейнер <**samp**> за допомогою моноширинного шрифту.

Приклад	На екрані
На екрані з"явиться повідомлення **<samp>**Операцію завершено**.</samp>**	На екраны з"явиться повідомлення Операцію завершено.

Оформлення тексту – назви змінної

Тег <**var**> може бути використаним у випадку, коли необхідно вивести позначення якоїсь змінної величини.

Приклад	На екрані
Змінну **<var>x</var>** потрібно помножинити на змінну **<var>y</var>**	Змінну *x* потрібно помножинити на змінну *y*

Теги форматування тексту

Виділення (акцентування) тексту

HTML містить два елементи для вказівки, що текст усередині необхідно виділити. Це можуть бути повідомлення про помилки, попередження або примітки. Для візуальних браузерів це звичайно означає застосування іншого кольору, шрифту або використання більше жирного шрифту або курсиву. Для тексту, який необхідно виділити, використовується елемент **em** у такий спосіб:

Приклад	На екрані
Please note: the kettle is to be unplugged at night	*Please note:* the kettle is to be unplugged at night.

Якщо треба виділити все речення, але в ньому є місце, яке повинно бути виділено ще сильніше, то використовується елемент **strong**, щоб вказати на ще сильніше виділення, ніж звичайне:

Приклад	На екрані
Please note: the **must** kettle to be unplugged at night	*Please note:* the kettle **must** to be unplugged at night.

<sub> Нижній індекс. Текст при цьому розташовується нижче базової лінії інших символів рядка і зменшеного розміру – H_2O.

<sup> Верхній індекс – $x^2+bx+c=0$.

<**del**> (від англ. deleted – видалений) 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.

<ins> (від англ. inserted − вставлений) визначає текст, котрий був доданий в нову версію документу. Браузери зазвичай перекреслюють видалений текст і підкреслюють вставлений.

Приклад	На екрані
Please note: the kettle **~~is~~** **<ins>** must**</ins>** to be unplugged at night.	*Please note:* the kettle ~~is~~ **must** to be unplugged at night.

Елементи подання (застаріли)

Специфікація HTML включає кілька елементів, які звичайно описуються як "елементи подання", тому що вони тільки визначають, як повинен виглядати розміщений у них текст, а не те, що він означає. Деякі з них були позначені як виключені в специфікації. Це означає, що вони для досягнення того ж результату були замінені більше новими методами.

Давайте опишемо тут їх коротко, але звернемо увагу, що це у більшій мірі становить історичний інтерес − ці елементи ніколи не повинні використовуватися у будь-якій сучасній Web-сторінці. Результат всіх цих елементів повинен досягатися за допомогою CSS.

font face="..." size="..."

Текст у цих елементах повинен представлятися браузером за допомогою шрифту відмінного від використовуваного за замовчуванням — замість цього шрифти повинні задаватися за допомогою CSS.

b

Текст у цьому елементі повинен бути жирним — це майже завжди означає, що текст був виділений, тому необхідно використовувати елементи **em** або **strong** як показано раніше.

s і strike

Текст у цьому елементі буде перекреслений лінією — якщо це тільки ефект подання, то цього можна домогтися за допомогою CSS. Інший варіант, якщо текст дійсно був позначений для видалення або як небажаний, то він повинен бути позначений елементом **del**.

u

Текст у цьому елементі буде підкреслений – це майже завжди візуальний ефект, і тому повинен реалізовуватися за допомогою CSS.

tt

Текст у цьому елементі представляється шрифтом "телетайпа" або моноширинним шрифтом − це можна реалізувати за допомогою CSS або більш підходящого семантичного елемента, такого як **pre** − як показано вище.

<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 використовується за замовчуванням, прийнятий 3. Таким чином, додавання тега <**big**> збільшує текст на одну умовну одиницю.

<small> зменшує розмір шрифту на одиницю. Протилежний до <**big**>. Розмір тексту всередині буде змінюватися − це можна реалізувати CSS.

Списки HTML

У цій частині ми розглянемо різні типи списків, доступних в HTML, коли і як їх потрібно використовувати і як застосовувати деякі базові стилі. Списки використовуються для об'єднання в групу зв'язаних об'єктів інформації, щоб полегшити їхнє асоціювання один з одним і читання. У сучасній розробці Web списки є основними елементами, які часто використовуються для навігації, а також звичайного вмісту. Списки добре підходять зі структурної точки зору, тому що вони допомагають створити добре структурований, доступніший і простий в обслуговуванні документ. Вони корисні також із чисто практичних міркувань − надають додаткові елементи для з'єднання зі стилями CSS для цілої множини стилів оформлення.

В HTML є три типи списків:

- **Неупорядкований список** − використовується для об'єднання в групу множини зв'язаних об'єктів без певного порядку.
- **Упорядкований список** − використовується для об'єднання в групу множини зв'язаних об'єктів у певному порядку.
- **Список визначень** − використовується для висновку пар ім'я/значення, таких як терміни і їхні визначення, або час і події.

Кожний з них має певне призначення — вони не є взаємозамінними!

Неупорядковані списки

Неупорядковані списки або маркіровані списки, використовуються, коли множина об'єктів може бути розміщена в будь-якому порядку. Прикладом є список покупок:

- **молоко**
- **хліб**
- **вершкове масло**
- **кавові зерна**

Однак всі ці об'єкти є частиною одного списку, можна розмістити об'єкти в будь-якому порядку, а список буде як і раніше мати сенс:

- хліб
- кавові зерна
- молоко
- вершкове масло

Можна використовувати CSS для зміни маркера на один з декількох використовуваних за замовчуванням стилів, використовувати своє власне зображення або навіть вивести список без маркерів.

Розмітка неупорядкованого списку

Неупорядковані списки використовують одну пару тегів `<ul></ul>` – тег списку, що охоплюють множину пар тегів `<li></li>` – теги для елементів списку:

Код	На екрані
   хліб **кава в зернах** **молоко** **масло** 	• хліб • кава в зернах • молоко • масло

Параметр **type.**

Цей параметр дозволяє задати тип маркера для кожного елемента списку (круг (**disc**), коло (**circle**), квадрат (**square**))

Код	На екрані
Список продуктів <ul **type="disk"**> - хліб - кава в зернах - молоко - масло 	**Список продуктів** • хліб • кава в зернах • молоко • масло
Список продуктів <ul **type="circle"**> - хліб - кава в зернах - молоко - масло 	**Список продуктів** ○ хліб ○ кава в зернах ○ молоко ○ масло
Список продуктів <ul **type="square"**> - хліб - кава в зернах - молоко - масло 	**Список продуктів** ▪ хліб ▪ кава в зернах ▪ молоко ▪ масло

Упорядковані списки

Упорядковані списки або нумеровані списки використовуються для виведення списку об'єктів, які необхідно розмістити в певному порядку. Прикладом можуть бути кулінарні інструкції з готування, які повинні бути виконані в певному порядку рецепта:

1. Зібрати інгредієнти
2. Змішати інгредієнти
3. Помістити інгредієнти у форму для випічки

4. Випікати в духовці протягом години
5. Вийняти з духовки
6. Дати постояти десять хвилин
7. Подати

Упорядковані списки можуть виводитися за допомогою однієї з декількох цифрових або алфавітних систем – тобто з буквами або числами. За замовчуванням у більшості браузерів використовуються десяткові числа, але є і інші варіанти.

- Літери (потрібно використати атрибут `type= "A | a "`)
  - Літери ASCII нижнього регістру (a, b, c...) (`type= "a"`)
  - Літери ASCII верхнього регістру (A, B, C...) (`type= "A"`)
- Числа (потрібно використати атрибут `type= "1 | I | i"`)
  - Десяткові числа (1, 2, 3...) (`type= "1"`)
  - Римські числа у нижньому регістрі (i, ii, iii...) (`type= "i"`)
  - Римські числа у верхньому регістрі (I, II, III...) (`type= "I"`)

Розмітка упорядкованого списку

Упорядковані списки використовують одну пару тегів **
**, що охоплюють множину пар тегів **
**:

```
<ol>
  <li> Зібрати інгредієнти </li>
  <li> Змішати інгредієнти </li>
  <li> Помістити інгредієнти у форму для випічки </li>
  <li> Випікати в духовці протягом  години </li>
  <li> Вийняти з духовки </li>
  <li> Дати постояти десять хвилин </li>
  <li> Подати </li>
</ol>
```

Початок упорядкованих списків із числа, відмінного від 1

Можна створити впорядкований список, нумерація якого починається із числа відмінного від 1 (або і, або I і т.д.). Це робиться за допомогою атрибута **start**, що одержує числове значення, навіть якщо використовується CSS для зміни нумерації списку на алфавітну або римську за допомогою властивості **list- style-type**. Це буде корисно, якщо є єдиний список об'єктів, але ви хочете розбити список деякими примітками або деякою іншою підходящою інформацією. Наприклад, можна було б зробити це з попереднім прикладом:

```
<ol>
  <li>Зібрати інгредієнти </li>
  <li>Змішати інгредієнти </li>
  <li> Помістити інгредієнти у форму для випічки </li>
</ol>
```

Перш ніж помістити інгредієнти у форму для випічки, нагрійте духовку до 180 градусів по Цельсію /350 градусів по Фаренгейтові, щоб бути готовим до наступного кроку

```
<ol start="4">
  <li> Випікати в духовці протягом  години </li>
  <li> Вийняти з духовки </li>
  <li> Дати постояти десять хвилин </li>
  <li> Подати </li>
</ol>
```

Це приведе до наступного результату:

4. Зібрати інгредієнти
5. Змішати інгредієнти
6. Помістити інгредієнти у форму для випічки

Перш ніж помістити інгредієнти у форму для випічки, нагрійте духовку до 180 градусів по Цельсію /350 градусів по Фаренгейтові, щоб бути готовим до наступного кроку

4. Випікати в духовці протягом години
5. Вийняти з духовки
6. Дати постояти десять хвилин
7. Подати

Нумерація у зворотному порядку. Атрибут reversed

Код	На екрані
Список продуктів <ol type="1" **reversed**> - хліб - кава в зернах - молоко - масло 	**Список продуктів** 4. хліб 3. кава в зернах 2. молоко 1. масло

Списки означень

Списки означень з'єднують певні об'єкти і їхні означення у списку. Наприклад, якщо ви хочете додати означення до об'єктів у списку покупок, можна зробити це так:

молоко
 білий рідкий молочний продукт
хліб
 продукт харчування, одержуваний випічкою з борошна

вершкове масло
 жовтий твердий молочний продукт
кавові зерна
 насіння плодів деяких кавових дерев

Кожне означення й термін є групою означення(або групою ім'я-значення). Можна мати будь-яку необхідну кількість груп означень, але в кожній групі повинні бути як мінімум один термін і як мінімум одне означення. Не можна мати термін без означення або означення без терміна. Можна асоціювати більше одного терміна з одним означенням або навпаки. Наприклад, термін "кава" може мати кілька значень і можна показати їх одне за одним:

кава
 напій, приготовлений зі смажених, мелених кавових зерен
 чашка кави
 зустріч, під час якої п'ють каву
 колір від середнього до темно коричневого

Альтернативно можна мати більше одного терміна з одним означенням. Це використовується, щоб показати варіації терміна, які всі мають одне значення:

содова
шипучка
газована
кола
 солодкий, насичений вуглекислим газом напій

Списки визначень відрізняються від інших видів списків, тому що вони використовують обумовлені терміни й описи визначень замість об'єктів списку. Тому списки визначень використовують одну пару елементів <**dl**></**dl**>, що охоплює групи тегів <**dt**></**dt**> і <**dd**></**dd**>. Необхідно мати у своєму розпорядженні пари як мінімум одну групу тегів <**dt**></**dt**> з однією групою <**dd**></**dd**>; теги **<dt></dt>** повинні завжди бути першими по порядку. Простий список визначень одного терміна з одним означенням буде виглядати так:

```
<dl>
  <dt>Термін </dt>
  <dd>Визначення терміна </dd>
  <dt>Термін </dt>
  <dd>Визначення терміна </dd>
  <dt>Термін </dt>
  <dd>Визначення терміна </dd>
</dl>
```

Який виводиться у такий спосіб:

Термін
 Визначення терміна

Термін
Визначення терміна
Термін
Визначення терміна

У цьому прикладі ми з'єднуємо більше одного терміна з означенням і навпаки:

```
<dl>
  <dt>Термін </dt>
  <dd>Визначення терміна </dd>
  <dt>Термін </dt>
  <dt>Термін </dt>
  <dd>Визначення яке застосовується до обох попередніх термінів </dd>
  <dt>Термін, що може мати обидва наступні визначення </dt>
  <dd>Одне визначення терміна </dd>
  <dd>Інше визначення терміна </dd>
</dl>
```

Що буде представлено у такий спосіб:

Термін
Визначення терміна
Термін
Термін
Визначення яке застосовується до обох попередніх термінів
Термін, що може мати обидва наступні означення
Одне визначення терміна
Інше визначення терміна

Звичайно не прийнято зв'язувати кілька термінів з одним означенням, але якщо виникне така необхідність, корисно знати, що це можливо.

Вибір типу списку

При ухваленні рішення про використання певного типу списку, звичайно можна вирішити це, задаючи два простих питання:

1. Визначаються терміни (або з'єднуються інші пари ім'я/значення)?
   - Якщо так, використовуйте список визначень.
   - Якщо ні, не використовуйте список визначень — перейдіть до наступного питання.
2. Чи важливий порядок елементів списку?
   - Якщо так, використовуйте впорядкований список.
   - Якщо ні, використовуйте неупорядкований список.

Розбіжності між HTML- списками і текстом

Можна задатися питанням, у чому розбіжності між списком HTML і якимось текстом з маркерами або числами, написаними вручну. Існує кілька переваг використання списку HTML:

- Якщо буде потрібно змінити порядок елементів списку в упорядкованому списку, то ви просто переміщаєте їх у коді HTML. Якщо числа будуть написані вручну, то прийдеться все переглянути й змінити число кожного елемента, щоб виправити порядок – що принаймні досить нудно!
- Використання списку HTML дозволяє правильно оформити стиль список. Якщо використовується просто великий текст, то виявиться значно складніше оформити стиль окремих елементів яким-небудь більш-менш корисним чином.
- Використання списку HTML створює для вмісту підходящу семантичну структуру, а не просто подібний до списку візуальний ефект. Це має важливі переваги, тому що дозволяє агентам користувачів повідомити користувачів з вадами зору, що вони читають список, а не просто плутану суміш тексту й чисел.

З іншої сторони: текст і списки не є тим самим. Використання тексту замість списку вимагає більше роботи і може створювати проблеми для читачів документа. Тому, якщо в документі потрібен список, то необхідно використовувати правильний список HTML.

Вкладені списки

Елемент списку може містити інший цілий список — він називається "вкладеним" списком. Це може бути корисно для таких речей як структура змісту:

1. Глава один
   1. Розділ один
   2. Розділ два
   3. Розділ три
2. Глава два
3. Глава три

Важливо пам'ятати — вкладений список повинен відноситися до одного конкретного пункту списку. Щоб відбити це в коді, вкладений список поміщають усередину цього пункту списку. Код для наведеного вище списку виглядає так:

```
<ol>
<li>Глава один
<ol>
<li>Розділ один </li>
<li>Розділ два </li>
<li>Розділ три </li>
</ol>
</li>
<li>Глава два </li>
<li>Глава три  </li>
</ol>
```

Відзначимо, що вкладений список починається після елемента <**li**> і тексту, що містить список елемента ("Глава один"); а закінчується перед елементом </**li**>, що містить список елемента. Вкладені списки часто формують основу для навігаційного меню Web-сайту, тому що вони є зручним способом означення структури Web-сайту.

Теоретично можна вкласти будь-яку кількість списків, хоча на практиці це може внести плутанину при занадто глибокому вкладенні списків. Для дуже великих списків краще розбити вміст на кілька списків із заголовками, або навіть розбити на окремі сторінки.

Таблиці в HTML

<table></table>	Створює таблицю ***Атрибути і параметри*** — ***Значення*** *width="..."* — Задає ширину таблиці в пікселах або відсотках *height="..."* — Задає висоту рядків в таблиці в пікселах або відсотках *valign="top"* — Вертикальне вирівнювання тексту – притиснути вміст догори *valign="bottom"* — Вертикальне вирівнювання тексту – притиснути вниз *valign="middle"* — Вертикальне вирівнювання тексту – розмістити по центру *align="left"* — Горизонтальне вирівнювання тексту – притиснути вліво *align="right"* — Горизонтальне вирівнювання тексту – притиснути праворуч *align="center"* — Горизонтальне вирівнювання тексту – розмістити по центру *cellpadding="..."* — Визначає відстань між границею комірки та її вмістом у пікселах *cellspacing="..."* — Визначає відстань між комірками у пікселах *bgcolor="..."* — Встановлює колір фону (у форматі RGB) *background="url"* — Встановлює фонове граічне зображення (адреса) *bordercolor="..."* — Встановлює колір рамки (у форматі RGB) Вищенаведені атрибути і параметри застосовуються також і для тегів <tr> і <td>
<tr></tr>	Визначає рядок у таблиці ***Атрибути і параметри*** — ***Значення*** *colspan="..."* — Встановлює число комірок, які повинні бути об'єднані по горизонталі *rowspan="..."* — Встановлює число комірок, які повинні бути об'єднані по вертикалі
<td></td>	Визначає окрему комірку в таблиці ***Атрибути і параметри*** — ***Значення*** *colspan="..."* — Встановлює число комірок, які повинні бути об'єднані по горизонталі *rowspan="..."* — Встановлює число комірок, які повинні бути об'єднані по вертикалі
<th></th>	Визначає заголовок таблиці (нормальна комірка з відцентрованим жирним текстом)
<caption></caption>	Визначає підпис таблиці
<thead></thead> <tbody> </tbody> <tfoot></tfoot>	Дозволяють задати структуру таблиці

Основний контейнер

Тег <**table**> вказує браузеру, що контент необхідно організувати у табличному вигляді

```
<table >
. . . . Вміст таблиці . . . .
</table>
```

Підпис таблиці

Тег <**caption**> дозволяє створити заголовок для даних таблиці. Більшість браузерів будуть вирівнювати заголовок по центру й робити його такої ж ширини, як і таблиця (якщо тільки для іншого вирівнювання тексту не використовується CSS).

Зовнішній вигляд (загальна схема)	На екрані (загальна схема)
<table> **<caption>Підпис таблиці</caption>** . . . . Вміст таблиці . . . . </table>	Підпис таблиці . . . . Вміст таблиці . . . .

Додавання рядків

Кожен рядок задається за допомогою пари тегів: <**tr**> і </**tr**> . Між цими тегами розміщують весь вміст рядка (тобто описують клітинки таблиці).

```
<tr> . . . Вміст рядка. . . </tr>
```

Зовнішній вигляд (загальна схема)	На екрані (загальна схема)
<table> <caption>Підпис таблиці</caption> **<tr>** **. . . . Рядок 1 . . . .** **</tr>** **<tr>** **. . . . Рядок 2 . . . .** **</tr>** **. . . .** </table>	Підпис таблиці . . . . Рядок 1 . . . . Рядок 2 . . . .

Вставка клітинок у рядки

Для додавання клітинки у рядок необхідно між відповідними тегами: <**tr**> і </**tr**> потрібного рядка вставити теги <**td**> і </**td**>.

```
<tr>
    <td> Вміст клітинки 1 </td>
    <td> Вміст клітинки 2 </td>
    . . .
</tr>
```

Зовнішній вигляд	На екрані
<table> <caption>Підпис таблиці</caption> **<tr>** **<td> Клітинка 1 </td>** **<td> Клітинка 2 </td>** **<td> Клітинка 3 </td>** **</tr>** **<tr>** **<td> Клітинка 4 </td>** **<td> Клітинка 5 </td>** **<td> Клітинка 6 </td>** **</tr>** **. . . .** </table>	Підпис таблиці Клітинка 1 Клітинка 2 Клітинка 3 Клітинка 4 Клітинка 5 Клітинка 6 . . . .

Об'єднання клітинок

По горизонталі

Атрибут **colspan**="кількість стовпців" тега <**td**> дозволяє створити клітинку, яка буде об'єднувати у собі вказану кількість клітинок по горизонталі (або ж стовпців, починаючи з її місця розташування)

Зовнішній вигляд (загальна схема)	На екрані (загальна схема)
<table border="2px" width="300px"> <caption>Підпис таблиці</caption> <tr> **<td colspan="2"> Клітинка 1 </td>** <td> Клітинка 2 </td> </tr> <tr> <td> Клітинка 3 </td> <td> Клітинка 4 </td> <td> Клітинка 5 </td> </tr> **. . . .** </table>	Підпис таблиці Клітинка 1 Клітинка 2 Клітинка 3 Клітинка 4 Клітинка 5 . . . .

По вертикалі

Атрибут **rowspan**="кількість рядків" тега <td> дозволяє створити клітинку, яка буде об'єднувати у собі вказану кількість клітинок по вертикалі (або ж рядків, починаючи з її місця розташування)

Зовнішній вигляд (загальна схема)	На екрані (загальна схема)
<table border="2px" width="300px"> <caption>Підпис таблиці</caption> <tr> **<td rowspan="2"> Клітинка 1 </td>** <td> Клітинка 2 </td> <td> Клітинка 3 </td> </tr> <tr> <td> Клітинка 4 </td> <td> Клітинка 5 </td> </tr> <tr> <td> Клітинка 6 </td> <td> Клітинка 7 </td> <td> Клітинка 8 </td> </tr> **. . . .** </table>	Підпис таблиці Клітинка 1 / Клітинка 2 / Клітинка 3 Клітинка 4 / Клітинка 5 Клітинка 6 / Клітинка 7 / Клітинка 8 . . . .

Об'єднання по вертикалі і горизонталі

Для об'єднання по вертикалі і горизонталі необхідно використати обидва атрибути : **`colspan i rowspan.`**

Зовнішній вигляд (загальна схема)	На екрані (загальна схема)
<table border="2px"> <caption>Підпис таблиці</caption> <tr> **<td colspan="3" rowspan="2"> Клітинка 1 </td>** <td> Клітинка 2 </td> </tr> <tr> <td> Клітинка 3 </td> </tr> <tr> <td> Клітинка 4 </td> <td> Клітинка 5 </td> <td> Клітинка 6 </td> <td> Клітинка 7 </td> </tr> **. . . .** </table>	Підпис таблиці Клітинка 1 / Клітинка 2 / Клітинка 3 Клітинка 4 / Клітинка 5 / Клітинка 6 / Клітинка 7 . . . .

Заголовки для рядків і стовпців

Заголовки для стовпців

Теги <**th**> розміщені у рядку визначає вміст між тегами як заголовок таблиці для відповідного стовпця.

Зовнішній вигляд (загальна схема)	На екрані (загальна схема)
<table border="2px"> <caption>Підпис таблиці</caption> <tr> **<th> Заголовок 1 </th>** **<th> Заголовок 2 </th>** **<th> Заголовок 3 </th>** </tr> <tr> <td> Клітинка 1 </td> <td> Клітинка 2 </td> <td> Клітинка 3 </td> </tr> <tr> <td> Клітинка 4 </td> <td> Клітинка 5 </td> <td> Клітинка 6 </td> </tr> <tr> <td> Клітинка 7 </td> <td> Клітинка 8 </td> <td> Клітинка 9 </td> </tr> . . . . </table>	Підпис таблиці **Заголовок 1** **Заголовок 2** **Заголовок 3** Клітинка 1 Клітинка 2 Клітинка 3 Клітинка 4 Клітинка 5 Клітинка 6 Клітинка 7 Клітинка 8 Клітинка 9 . . . .

Заголовки для рядків

Теги <**th**> розміщені першими у кожному із рядків задають заголовки для рядків.

Зовнішній вигляд (загальна схема)	На екрані (загальна схема)
<table border="2px"> <caption>Підпис таблиці</caption> <tr> **<th> Заголовок 1 </th>** <td> Клітинка 1 </td> <td> Клітинка 2 </td> <td> Клітинка 3 </td> </tr> <tr> **<th> Заголовок 2 </th>** <td> Клітинка 4 </td> <td> Клітинка 5 </td> <td> Клітинка 6 </td> </tr>	Підпис таблиці **Заголовок 1** Клітинка 1 Клітинка 2 Клітинка 3 **Заголовок 2** Клітинка 4 Клітинка 5 Клітинка 6 **Заголовок 3** Клітинка 7 Клітинка 8 Клітинка 9 . . . .

<tr> **<th> Заголовок 3 </th>** <td> Клітинка 7 </td> <td> Клітинка 8 </td> <td> Клітинка 9 </td> </tr> **. . . .** </table>	

Зауважимо, що одночасно можна використовувати як заголовки стовпців, так і заголовки рядків.

Зовнішній вигляд	На екрані
<table border="2px"> <caption>Підпис таблиці</caption> <tr> **<th> Заголовоки </th>** **<th> Стовпець 1 </th>** **<th> Стовпець 2 </th>** **<th> Стовпець 3 </th>** </tr> <tr> **<th> Рядок 1 </th>** <td> Клітинка 1 </td> <td> Клітинка 2 </td> <td> Клітинка 3 </td> </tr> <tr> **<th> Рядок 2 </th>** <td> Клітинка 4 </td> <td> Клітинка 5 </td> <td> Клітинка 6 </td> </tr> <tr> **<th> Рядок 3 </th>** <td> Клітинка 7 </td> <td> Клітинка 8 </td> <td> Клітинка 9 </td> </tr> **. . . .** </table>	Підпис таблиці **Заголовоки** | **Стовпець 1** | **Стовпець 2** | **Стовпець 3** **Рядок 1** | Клітинка 1 | Клітинка 2 | Клітинка 3 **Рядок 2** | Клітинка 4 | Клітинка 5 | Клітинка 6 **Рядок 3** | Клітинка 7 | Клітинка 8 | Клітинка 9 . . . .

Структурні елементи таблиці

Тег <**thead**> використовується для зберігання групи заголовків HTML-таблиці (верхній колонтитул), котрі будуть відображені зверху таблиці. Браузери можуть використовувати цей елемент для того, щоб прокручувати тіло таблиці незалежно від верхнього колонтитула. Крім того, при друці великої таблиці, яка займає кілька сторінок, цей елемент може бути надрукований у верхній і нижній частині кожної сторінки. Допустимо використовувати не більше одного елемента <**thead**> в межах однієї таблиці, і він повинен йти в вихідному коді відразу після елемента <**table**>. Якщо в таблиці вказана її назва (тег <**caption**>), тег <**thead**> мусить іти відразу після <**caption**>.

Тег <**tbody**> визначає тіло HTML-таблиці. Допускається використання декількох тегів всередині однієї таблиці.

Тег <**tfoot**> використовується для групування підсумкових рядків в HTML-таблиці, котрі будуть відображені внизу таблиці. Допускається використання не більше одного в межах однієї таблиці

Всередині тегів допускається розміщення тегів **tr**, **td** та **th**.

```
<table >
    <thead>
    . . . . Верхній колонтитул таблиці . . . .
    </thead>
    <tbody>
    . . . . Тіло таблиці . . . .
    </tbody>
    <tfoot>
    . . . . Нижній колонтитул таблиці . . . .
    </tfoot>
</table>
```

Зовнішній вигляд	На екрані

```
<table border="2px" >
 <caption>
 Оформленнязамовлення
 </caption>
<thead>
 <tr>
 <th colspan="4"> Список продуктів </th>
 </tr>
 <tr>
 <th colspan="2"> З городу </th>
 <th colspan="2"> Молочна продукція </th>
 </tr>
 <tr>
 <th> Овочі </th>
 <th> Фрукти </th>

 <th> Консервовані </th>
 <th> Охолоджені </th>
 </tr>

</thead>

<tbody>
 <tr>
 <td> Капуста </td>
 <td> Вишні </td>

 <td> Згущене молоко </td>
 <td> Сметана </td>
 </tr>
 <tr>
 <td> Помідори </td>
 <td> Груші </td>
 <td> Карамель </td>
 <td> Молоко </td>
 </tr>
</tbody>

<tfoot>
 <tr>
 <td colspan="3"> Дата пакування </td>
 <td>12.09.2017</td>
 </tr>
 <tr>
 <td colspan="3"> Час доставки </td>
 <td>14.00</td>
 </tr>
<tfoot>

</table>
```

Оформлення замовлення

Список продуктів			
З городу		Молочна продукція	
Овочі	Фрукти	Консервовані	Охолоджені
Капуста	Вишні	Згущене молоко	Сметана
Помідори	Груші	Карамель	Молоко
Дата пакування			12.09.2017
Час доставки			14.00

Приклад. Для нижнього і верхнього колонтитулів встановлено кольори

Зовнішній вигляд	На екрані

```
<table border="2px" >
 <caption>
 Оформленнязамовлення
 </caption>
 <thead style="background:red">
 <tr>
 <th colspan="4"> Список продуктів </th>
 </tr>
 <tr>
 <th colspan="2"> З городу </th>
 <th colspan="2"> Молочна продукція </th>
 </tr>
 <tr>
 <th> Овочі </th>
 <th> Фрукти </th>

 <th> Консервовані </th>
 <th> Охолоджені </th>
 </tr>
</thead>
<tbody>
 <tr>
 <td> Капуста </td>
 <td> Вишні </td>

 <td> Згущене молоко </td>
 <td> Сметана </td>
 </tr>
 <tr>
 <td> Помідори </td>
 <td> Груші </td>
 <td> Карамель </td>
 <td> Молоко </td>
 </tr>
</tbody>
<tfoot style="background:green">
 <tr>
 <td colspan="3"> Дата пакування </td>
 <td>12.09.2017</td>
 </tr>
 <tr>
 <td colspan="3"> Час доставки </td>
 <td>14.00</td>
 </tr>
<tfoot>
</table>
```

Оформленнязамовлення

Список продуктів			
З городу		Молочна продукція	
Овочі	Фрукти	Консервовані	Охолоджені
Капуста	Вишні	Згущене молоко	Сметана
Помідори	Груші	Карамель	Молоко
Дата пакування			12.09.2017
Час доставки			14.00

HTML-посилання

HTML-посилання дозволяють створювати зв'язки між документами та ресурсами. Посилання поділяються на два види:

- **посилання на допоміжні ресурси** (тег <**link**>) , за допомогою яких до документу можна підключити коди програм, таблиці стилів і т.д.;
- **гіперпосилання** (теги <**a**>, <**area**>) , посилання на інші документи або ж матеріали, до яких можна перейти (за допомогою браузера) або ж які можна завантажити.

Гіперпосилання

Гіперпосилання задаються за допомогою тегів <**a**></**a**>.

Найпростіший загальний вигляд	**** об'єкт-мітка **** об'єкт мітка (або ж покажчик) – це текст або зображення, на які треба клікнути для доступу до ресурсу
Приклад	Новини можна знайти **** тут ****
На екрані	Новини можна знайти тут

Атрибути тега **<a>.**

Атрибут	Опис
download	Повідомляє браузеру, що даний ресурс потрібно завантажити, а не відкривати за допомогою браузера (наприклад, PDF-файл). Даний атрибут може мати значення, яке дозволяє одразу вказати ім'я під яким буде збережено документ. <img src="/images/logo.png" alt="logo"> <img src="/images/logo.png" alt="logo"> Загрузить отчет за Февраль 2014
href	За допомогою цього атрибуту задаємо шлях до ресурсу (іншого HTML-документу, PDF-файлу, графічного зображення …)
hreflang	Містить код мови ("ua", "ru", "en",…), на якій створено зовнішній ресурс Anysite
media	Атрибут media визначає, які медіа-пристрої оптимізовані для документу. Цей атрибут використовується для вказівки того, що цільовий URL призначений для спеціальних пристроїв (наприклад, iPhone), мови або друкуючих пристроїв. Цей атрибут може приймати кілька значень розділених за допомогою операторів. Позначення Призначення *And* Вказує на оператор AND *Not* Вказує на оператор NOT

Вказує на оператор OR

,

```
<a href="att_a_media.asp?output=print"
media="print and (resolution:300dpi)">
Open media attribute page for print.</a>
```

Значення атрибута

Значення	Опис значення
all	Без задання. Підходить для всіх пристроїв.
aural	Синтезатори мови.
braille	Пристрої зворотного зв'язку Брайля.
handheld	Кишенькові пристрої (маленький екран, обмежена пропускна здатність).
projection	Проектори.
print	Для друку.
screen	Для екранів комп'ютерів.
tty	Телетайпи та аналогічні методи ЗМІ з використанням сітки символів із фіксированим кроком.
tv	Для пристроїв ТВ-телебачення.
width	Визначає ширину цільової області відображення. Можуть бути використан префікси «min-» та «max-». Приклад: media="screen and (min-width:500px)".
height	Визначає висоту цільової області дисплея. Можуть бути використан префікси «min-» та «max-». Приклад: media="screen and (max-height:700px)".
device-width	Визначає ширину цільового відображення/паперу. Можуть бути використан префікси «min-» та «max-». Приклад: media="screen and (device-width:500px)".
device-height	Вказує висоту цільового відображення / паперу. Можуть бути використан префікси «min-» та «max-». Приклад: media="screen and (device-height:500px)".
orientation	Визначає орієнтацію цільового відображення/паперу. Можливі значення: "portrait" або "landscape" Приклад: media="all and (orientation: landscape)".
aspect-ratio	Визначає співвідношення ширина/висота цільової області дисплея. Можуть бути використан префікси «min-» та «max-». Приклад: media="screen and (aspect-ratio:16/9)".
device-aspect-ratio	Задає співвідношення ширини_пристрою/висоти_пристрою цільового дисплея/паперу. Можуть бути використан префікси «min-» та «max-». Приклад: media="screen and (aspect-ratio:16/9)".
color	Визначає кількість біт на колір цільового дисплея. Можуть бути використан префікси «min-» та «max-». Приклад: media="screen and (color:3)".

<table>
<tr><td></td><td>
<table>
<tr><td>color-index</td><td>Визначає кількість кольорів цільового дисплея який він може обробляти. Можуть бути використан префікси «min-» та «max-». Приклад: "screen and (min-color-index:256)".</td></tr>
<tr><td>monochrome</td><td>Задає кількусть бітів на піксель в буфері кадру монохромних зображень. Можуть бути використан префікси «min-» та «max-». Приклад: media="screen and (monochrome:2)"</td></tr>
<tr><td>resolution</td><td>Визначає щільність пікселів (точок на дюйм або точок на сантиметр) цільового відображення/паперу. Можуть бути використан префікси «min-» та «max-». Приклад: media="print and (resolution:300dpi)"</td></tr>
<tr><td>scan</td><td>Визначає метод сканування на екрані телевізора. Можливі значення: "progressive" і "interlace". Приклад: media="tv and (scan:interlace)".</td></tr>
<tr><td>grid</td><td>Вказує, якщо пристрій виведення це сітка або растрове зображення. Можливі значення: "1" для сітки, і "0" в іншому випадку. Приклад: media="handheld and (grid:1)".</td></tr>
</table>
</td></tr>
<tr><td>ping</td><td>Дозволяє відстежувати, за якими посиланнями переходить користувач
<code>Blasters</code>
У цьому прикладі при переході за адресою <code>href="/products/blaster"</code> буде надіслано інформацію (невеликий HTTP POST запит) за адресою <code>ping="/track/products/blaster"</code></td></tr>
<tr><td>rel</td><td>Атрибут <code>rel</code> визначає відношення між поточним документом і документом-ресурсом (на який посилаємося).
<code>Space Flights</code>
<h1>Значення атрибута</h1>
<table>
<tr><th>Значення</th><th>Опис значення</th></tr>
<tr><td>alternate</td><td>Надає посилання на альтернативний вигляд документа (тобто друк сторінки, переклад або дзеркала).</td></tr>
<tr><td>author</td><td>Надає посилання на автора документа.</td></tr>
<tr><td>bookmark</td><td>Постійний URL використовується для установки закладок.</td></tr>
<tr><td>external</td><td>Вказує, що посилання на документ не є частиною одного і того ж сайту, що і поточний документ.</td></tr>
<tr><td>help</td><td>Надає посилання на документ довідки.</td></tr>
<tr><td>license</td><td>Надає посилання на інформацію про авторські права для документа.</td></tr>
<tr><td>next</td><td>Надає посилання на наступний документ в серії.</td></tr>
<tr><td>nofollow</td><td>Вказує на те, що гіперпосилання не передає свою вагу (зміст) тій сторінці, на яку посилається. ("Nofollow" використовується Google, щоб вказати, що пошукова система Google не повинна слідувати за цим посиланням, в основному використовується для платних посилань).</td></tr>
<tr><td>noreferrer</td><td>Вимагає, що браузер не повинен посилати заголовок HTTP Referer, якщо користувач слідує із гіперпосилання іншої сторінки.</td></tr>
</table>
</td></tr>
</table>

	noopener	Вимагає, щоб будь-який контекст перегляду, створений після гіперпосилання не повинен мати контекст перегляду браузера.
	prev	Попередній документ у виборі.
	search	Посилання на інструмент пошуку для документа.
	tag	Тег (ключове слово) для поточного документа.
target	Вказує на те, в якому вікні повинен відкритися документ, на який посилаємося. Значення атрибуту: **_self** — документ відкривається у поточному вікні; **_blank** — документ відкривається у новому вікні браузера; **_parent** — сторінка завантажується у фрейм-предка; **_top** — сторінка відкриваєтья у всьому вікні браузера; *framename* – сторінка відкривається у фреймі з вказаним ім'ям	
type	Вказує MIME-тип документа, на який веде посилання. CSS.In.UA	

Абсолютна і відносна адреси

Адреси задають за допомогою URL.

Адреси поділяють на:

абсолютна адреса

Загальний вигляд	метод доступу://им'я серверу:порт/шлях
Приклад	https://www.ukr.net/main.html

відносна адреса

Метод доступу (або ж протокол) здійснює обмін даних між робочими станціями. Найбільш поширеними є

Метод доступу	Призначення
file	забезпечує читання файлу з локального диску file:/gallery/pictures/summer.html
http	забезпечує роботу з HTML-документами http://site.ua/
	https – спеціальна реалізація протоколу HTTP, яка виокристовує шифрування (SSL або ж TLS) https://privarbank.com/
ftp	забезпечує передачу файлів великого розміру (здійснює доступ до FTP-сервера) ftp://pgu/directory/library
mailto	запускає сеанс поштового зв'язку з вказаним адресатом и хостом: mailto: nika@gmail.com

Ім'я сервера задає повне DNS ім'я комп'ютера (або IP-адресу). Якщо ім'я не вказане, то посилання вважається локальним (тобто ресурс знаходиться на цьому ж комп'ютері).

Приклад.

my_site.ua

Номер порта TCP вказує на порт, який необхідно використати. Якщо порт не зазначено, то використовується 80.

Стандартні порти:
21 – FTP
23 – Telnet
70 – Gopher
80 — HTTP

Шлях до файлу містить шлях до папки, у якій знаходиться ресурс.

Абсолютний і відносний шляхи

Якщо у посиланні зберігається тільки ім'я файлу, то вважаться, що цей файл знаходиться у тій же самій папці, у якій знаходиться даний (відкритий у браузері) документ.

Приклад: `http://my_site.ua/`**`index.html`**

Якщо не вказано імені файлу, то вважається, що буде завантажено сторінку, яка призначена сторінкою за замовчуванням у налаштуваннях сервера (як правило, сторінка `index.html`).

Приклад. `http:// my_site.ua`

Наявність символу «/» в кінці (Приклад: `http:// my_site.ua`**`/`**) вказує на те, що звертання йде до папки. Якщо ж цього символу немає, то напряму до файлу. В іншому випадку, потрібно задати шлях файлу.

Шляхи бувають двох видів:

- **абсолютний** – містить повний шлях із зазначенням структури папок на сервері
  Приклад. `http://my_site.ua`**`/docs/examples/Tables.doc`**
- **відносний** – не містить назви кореневої папки і грунтується на місці розташування даного (відкритого у браузері) документу. Відносні шляхи використовуються у тому випадку, коли необхідно задати шлях до документів в межах одного сайту. Відносний шлях може містити перелік імен папок і назву файлу.
  Приклад.
  `<a href=`**`"/docs/examples/Tables.doc"`**`>Теорія до масивів</a>`
  При цьому можуть бути використані три спеціальних позначення:
  / – вказує на кореневу папку (папку, у якій знаходяться документи сайту);
  `./` – вказує на поточну папку (у якій знаходиться відкритий документ);
  `../` – вказує на папку, у якій знаходиться поточна (потрібно піднятися на один рівень вверх) (Приклад. `<a href=`**`"../Tables.doc"`**`>Теорія до масивів</a>`).

Якорі

Якорі, або ж внутрішні посилання дають можливість швидко переходити до потрібного розділу відкритого документу. Для цього у потрібних розділах (до яких захочемо швидко перейти) необхідно додати атрибути **`id`**.

```
<p id="p1">...</p> <!—додаємо відповідний id до елементу (розділу)-->
```

```
<p id="p2">...</p>
<p id="p3">...</p>
<p id="p4">...</p>
```

Внутрішні посилання також задаються за допомогою тега <**a**> але при цьому замість шляху до файлу вказується якір (перед якорем обов'язково необхідно ставити знак #).

```
<h1>Времена года</h1>
<h2>Оглавление</h2>
<a href="#p1">Лето</a> <!-- додаємо якір #id елемента -->
<a href="#p2">Осень</a>
<a href="#p3">Зима</a>
<a href="#p4">Весна</a>
<p id="p1">...</p> <!-- додаємо відповідний id до елементу (розділу)-->
<p id="p2">...</p>
<p id="p3">...</p>
<p id="p4">...</p>
```

Якщо потрібно зробити посилання з однієї сторінки на певну частину іншої сторінки, то необхідно додати цей якір до шляху до потрібної сторінки

```
<a href="https://html5book.ru/css-shrifty/#about-color" class="site" target="_blank">color</a>
```

Зображення-посилання

Для того, щоб перетворити зображення (задається за допомогою тега <**img**>) у посилання, неохідно це зображення вставити в середину тега <**a**>

```
<a href="http://www.fast-torrent.ru/film/gran-za-granyu-tv.html" target="_blank"><img src="//html5book.ru/wp-content/uploads/2014/07/Fringe.jpg" alt="Fringe"></a>
```

Посилання на телефонний номер, скайп, адресу електронної пошти

Телефонний номер

```
<a href="tel:+74951234567">+7 (495) 123-45-67</a>
```

Електронна пошта

```
<a href="mailto:example@mail.ru">example@mail.ru</a>
```

Скайп(при кліку телефонувати)

```
<a href="skype:имя-пользователя?call">Skype</a>
```

Скайп (при кліку відкрити чат)

```
<a href="skype:имя-пользователя?chat">Skype</a>
```

Скайп (при кліку додати у список контактів)

```
 <a href="skype:имя-пользователя?add">Skype</a>
```

Скайп (при кліку надсилати файл)

```
<a href="skype:имя-пользователя?sendfile">Skype</a>
```

Вставка графічних зображень у HTML-документ

Вставка графічних зображень. Тег **<img>**

Вставка графічних зображень здійснюється за допомогою тегу <img>. Найбільш використовуваними є графічні зображення форматів GIF, JPEG та PNG. Розглянемо найпростіший приклад використання цього тегу.

Загальний вигляд	**<img src**="шлях до файлу" **alt=**"альтернативний текст">
Приклад	<html> <head> <title>Вставка графічних зображень</title> </head> <body> Ми за мир!!! **<img src=**"images/Piagon.jpeg" **alt=**"Peace"**>** </body> </html>
На екрані	**Ми за мир!!!**

Основні атрибути тегу <**img**>

Атрибут	Значення
src	Використовується для задання абсолютного або ж відносного шляху до зображення (URL) Приклади (використання абсолютних та відносних шляхів). <img **src=**"Piagon.jpeg"**> <img **src=**"images/Piagon.jpeg"**> <img **src=**"../Piagon.jpeg"**> <img **src=**"http://www.mycite.ua/images/Piagon.jpeg"**>
alt	Містить альтернативний текст і виводиться у випадку, коли виведення графічного зображення є неможливим (неможливо здійснити доступ до графічного файлу або ж користувач вимкнув опцію відображення графічних файлів). Приклад. <img src="Piagon.jpeg" **alt=**"Peace" **>**
height, width	Задають відповідно висоту та ширину графічного зображення (px або %). Якщо задано тільки один атрибут, то значення іншого визначається автоматично із збереженням пропорції. Приклад. <img src="Piagon.jpeg" **width="200px" height="300px"**>
title	Дозволяє відобразити підказку при наведенні покажчика миші на графічне зображення <img src="Piagon.jpeg" alt="Peace" **title="МИР!!!"**>
longdesc	Задає шлях до ресурсу, який містить розширений опис графічного зображення <img src="Piagon.jpeg" **longdesc="http://www.myCite.net"**>

Розширені атрибути тегу <**img**>

crossorigin	Встановлює можливість використання технології CORS - міждоменного доступу до ресурсів. Можливі значення: • *anonymous* – забороняє CORS-запитам доступ до персональних даних; • *use-credentials* – дозволяє CORS-запити, це означає, що запит буде надавати облікові дані.
ismap	Він визначає, що зображення є частиною серверної карти (частиною графічного зображення, яке зберігається на сервері) зображень (зображення з інтерактивними областями). Атрибут Ismap працює тільки якщо тег <img> є нащадком (знаходиться всередині) тегу <a>. При натисненні на таке графічне зобарежння координати натиснення надсилаються на сервер. Приклад. <img src="Piagon.jpeg" alt="Peace" **ismap**>
sizes та srcset	sizes – задає розмір графічнго зображення в залежності від параметрів відображення. srcset – Задає список можливих шляхів для завантаження у залежності від розширення екрану. Може містити декілька значень розділених комою. <img src="flower.jpg" srcset=" img/flower-mobile.jpg 320w, img/flower-wide-mobile.jpg 480w, img/flower-tablet.jpg 768w, img/flower-desktop.jpg 1024w, img/flower-hires.jpg 1280w" sizes=" (max-width: 20em) 30vw, (max-width: 30em) 60vw, (max-width: 40em) 90vw" alt="Роза в моём саду">
usemap	Пов'язує між собою картинку і карту-зображення, що задається за допомогою контейнера <map>. Значення атрибуту повинно починатися з символу # і містити значення атрибуту name або ж id тегу <map>.

Створення карт-зображень

Створення карт-зображень здійснюється шляхом поєднання тегу <**img**> та тегу <**map**>. Тег <**map**> дозволяє задавати так звані активні області, при кліку на які користувач може перейти до відповідного зв'язаного з цією областю ресурсу. Активні області задаються за допомогою тегу **<area>**.

Загальна схема виглядає так:

```
<img src="шлях_до_графічного_зображення" usemap="#ім'я_карти">
<map name="ім'я_карти">
 <area … параметри_активної_області_1 …>
 <area … параметри_активної_області_2 …>
 . . .
</map>
```

Тег **<area>**дозволяє описати одну активну область (у формі круга, прямокутника або замкнутого багатокутника).

Атрибути тега **<area>**

Атрибут	Короткий опис
alt	Задає альтернативний текст для активної області
coords	Задає межі активної області (послідовність чисел розділених комами): **для крука** – кооридати центра і радіус; **для прямокутника** – координати верхнього лівого і правого нижнього кутів; **для багатокутника** – координати вершин багатокутника у порядку слідування.
download	Якщо атрибут href містить посилання на ресурс, який може бути завантажено, то наявність цього атрибута вказує на те, що ресурс повинен бути завантаженим, а не відкритим за допомогою браузера.
href	Задає URL-адресу для посилання.
hreflang	Задає мову ресурсу, на яку посилається документ.
media	Атрибут media визначає, які медіа-пристрої оптимізовані для документу. Цей атрибут використовується для вказівки того, що цільовий URL призначений для спеціальних пристроїв (наприклад, iPhone), мови або друкуючих пристроїв.
rel	Атрибут rel визначає відношення між поточним документом і документом-ресурсом (на який посилаємося).
shape	Задає форму активної області на карті. Можливі значення: rect — активна область прямокутник; circle — активна область круг; poly — активна область багатокутник; default — активна область займає всю область зображення.
target	Вказує на те, в якому вікні повинен відкритися документ, на який посилаємося. Значення атрибуту: **_self** — документ відкривається у поточному вікні; **_blank** — документ відкривається у новому вікні браузера; **_parent** — сторінка завантажується у фрейм-предка; **_top** — сторінка відкриваєтья у всьому вікні браузера;
type	Вказує на MIME-тип ресурсу, на який задається посилання.

Приклад.

```
<html>
<head>
 <title>Головна сторінка</title>
</head>
<body>
 <h1>Величні будівлі світу</h1>
 <img src="Foto7.png" alt="World Miracles" usemap="#WorldMiracles" width="50%">
<map name="WorldMiracles">
 <area shape="circle" coords="740,128,140" href="https://en.wikipedia.org/wiki/Colosseum" alt="Колизей"
target="_blank">

 . . . . . .

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

Координати центру

Радіус

```
<html>
<head>
 <title>Головна сторінка</title>
</head>
<body>
 <h1>Величні будівлі світу</h1>
 <img src="Foto7.png" alt="World Miracles" usemap="#WorldMiracles" width="50%">
<map name="WorldMiracles">
 <area shape="rect" coords="400,268,868,304" href="https://en.wikipedia.org/wiki/Parthenon" alt="Parthenon"
target="_blank">
 . . . . . .
</map>
</body>
</html>
```

Координати верхн.лівого кута

Координати нижн.правого кута

```
<html>
<head>
 <title>Головна сторінка</title>
</head>
<body>
 <h1>Величні будівлі світу</h1>
 <img src="Foto7.png" alt="World Miracles" usemap="#WorldMiracles" width="50%">
<map name="WorldMiracles">
 <area shape="poly" coords="416,240,486,26,526,240" href="https://en.wikipedia.org/wiki/Eiffel_Tower"
alt="Eiffel_Tower" target="_blank">
 . . . . . .
</map>
</body>
</html>
```

Послідовність координат вершин

Повний текст документу

```
<html>
<head>
 <title>Головна сторінка</title>
</head>
<body>
 <h1>Величні будівлі світу</h1>
 <img src="Foto7.png" alt="World Miracles" usemap="#WorldMiracles" width="50%">
<map name="WorldMiracles">
 <area shape="circle" coords="740,128,140" href="https://en.wikipedia.org/wiki/Colosseum" alt="Колизей" target="_blank">
 <area shape="poly" coords="416,240,486,26,526,240" href="https://en.wikipedia.org/wiki/Eiffel_Tower" alt="Eiffel_Tower" target="_blank">
 <area shape="poly" coords="12,80,86,80,326,140,390,220,394,384,256,512,138,512,220,354,260,234,12,180"
 href="https://en.wikipedia.org/wiki/Great_Wall_of_China" alt="Great_Wall_of_China" target="_blank">
 <area shape="rect" coords="400,268,868,304" href="https://en.wikipedia.org/wiki/Parthenon" alt="Parthenon" target="_blank">
 </map>
</body>
</html>
```

Вставка аудіо-контенту

Додавання аудіо-контенту в HTML5 здійснюється за допомогою тегу <**audio**>.

Розглянемо найпростіший приклад використання

Загальний вигляд	`<audio src=" шлях до файлу " controls></audio>`
Приклад	`<audio src="TheBest.mp3" controls></audio>`

Атрибути тегу <**audio**>

Атрибут	Значення
`src`	Містить шлях (URL-адресу) аудіо файлу.
`controls`	Наявність цього тегу вказує на необхідність відображення елементів управління (відображення плеєра) відтворенням аудіо (гучність, пауза і ін.)
`autoplay`	Автоматичний старт відтворення одразу після завантаження сторінки
`loop`	Циклічне відтворення аудіо
`muted`	Відключення звуку
`preload`	Попереднє завантаження контенту. Возможные значения: `auto` – завантаження повність перед стартом; `metadata` – завантаження частини файлу; `none` – відсутність автоматичного завантаження.

Вставка відео-контенту

Вставка відео-контенту здійснюється за допомогою тегу <**video**>

Розглянемо найпростіший приклад використання

Загальний вигляд	`< video src=" шлях до файлу " controls></ video>`
Приклад	`< video src="TheBest.mp4" controls></ video >`

Атрибути тегу <**video**>

Атрибут	Значення
`autoplay`	Автоматичине відтворення одразу після завантаження стоірки
`controls`	Відображення елементів управління відтворенням (гучність, пауза та ін).
`src`	Содерж Містить шлях (URL-адресу) аудіо файлу
`width`	Ширина вікна для відображення відео-контенту (`px` або `%`)
`height`	Висота вікна відображення відео-контенту (`px` або `%`)
`loop`	Циклічне відтворення
`muted`	Відключення звуку під час відтворення.
`poster`	Шлях до графічного зображення-постера (URL). Якщо не задано, то буде відображено перший кард.
`preload`	Попереднє завантаження контенту. Возможные значения: `auto` – завантаження повність перед стартом; `metadata` – завантаження частини файлу; `none` – відсутність автоматичного завантаження.

Відео-контент (або ж векторні зображення у форматі swf) також може буде додано за допомогою тегу <**embed**>. Даний тег є контейнером для плагіну, який і дозволяє відобразити контент.

Загальний вигляд	`<embed src=" шлях до файлу " >`
Приклад	`<embed src="movie.mov" type="video/quicktime" width="640" height="480">` `<embed src="helloworld.swf">`

Атрибути тегу <embed>

Атрибут	Значення
src	URL-адреса ресурсу медиафайла.
type	MIME-тип ресурсу.
height	Висона вбудованого контенту (px або %).
width	Ширина вбудованого контенту (px або %).

Додавання декількох альтернативних джерел аудіо- та відео-контенту

Для забезпечення можливості відтворення аудіо у різних браузерах при розміщенні аудіо-контенту виникає необхідність зазначати декілька альтернативних файлів з різними форматами. Для цього використовують тег <**source**>.

Загальний вигляд	`<audio >`
	`<source src=" шлях 1 " type="audio/ формат 1">`
	`<source src=" шлях 2 " type="audio/ формат 2">`
	`. . . . .`
	`</audio>`

Приклад	`<audio controls>`
	`<source src="name.ogg" type="audio/ogg">`
	`<source src="name.mp3" type="audio/mpeg">`
	`</audio>`

На практиці також додатково може бути вказано альтернативний ресурс для завантаження у випадку, коли жоден з вказаних ресурсів не дозволить відтворити аудіо-контент.

Загальний вигляд	`<audio >`
	`<source src=" шлях 1 " type="audio/ формат 1">`
	`<source src=" шлях 2 " type="audio/ формат 2">`
	`. . . . .`
	`<a href=" альтернативний шлях "> Текст посилання </a>`
	`</audio>`

Приклад	`<audio controls>`
	`<source src="name.ogg" type="audio/ogg">`
	`<source src="name.mp3" type="audio/mpeg">`
	`<a href="MyAudio/TheBest.mp3">Завантажити TheBest.mp3</a>`
	`</audio>`

Аналогічно даний тег використовується і при відображенні відео-контенту

Загальний вигляд	`< video >`
	`<source src=" шлях 1 " type="video/ формат 1">`
	`<source src=" шлях 2 " type="video/ формат 2">`
	`. . . . .`
	`</video>`

Приклад	`<video width="320" height="240" controls>`
	`<source src="forrest_gump.mp4" type="video/mp4">`
	`<source src="forrest_gump.ogg" type="video/ogg">`
	`</video>`

Також як альтернативний варіант для попередніх версій браузерів може бути використаний тег <**object**>

Приклад	<video controls width="400" height="300"> <source src="video.mp4" type="video/mp4"> <source src="video.webm" type="video/webm"> <source src="video.ogv" type="video/ogg"> **<object data="video.swf" type="application/x-shockwave-flash">** **<param name="movie" value="video.swf">** **</object>** </video>

Атрибути тегу <**source**>

Атрибут	Значення
src	Шлях до файлу (URL)
type	Визначає MIME-тип файлу.
media	Визначає тип пристрою, для якого оптимізовано даний ресурс

Субтитри і заголовки

Додавання субтитрів та іншої додаткової інформації для аудіо- і відеоконтенту здійснюється за допомогою тегу <**track**>.

Розглянемо приклад використання тегу <**track**>

Загальний вигляд	<video > . . . . . **<track src=" шлях до ресурсу " kind="тип " srclang=" мова " label=" мітка">** </video>
Приклад	<video width="320" height="240" controls> <source src="forrest_gump.mp4" type="video/mp4"> **<track src="subtitles_en.vtt" kind="subtitles" srclang="en" label="English">** **<track src="subtitles_uk.vtt" kind="subtitles" srclang="ua" label="Українська">** </video>

Атрибути тегу <**track**>

Атрибут	Значення
src	Шлях до файлу з текстовою доріжкою (URL)
default	Текстова доріжка за замовчуванням. Тільки одна доріжка може мати атрибут default
kind	Вказує тип текстової доріжки. Можливі значення: captions — Дублювання діалогів, звукових ефектів, музичного супроводу в вигляді тексту для тих випадків, коли звук недоступний або для глухих користувачів. Виводиться поверх відео, при цьому позначається, що підходить для погано чуючих люде.

	`chapters` — Назви глав використовуються для швидкої навігації по відео чи аудіо. Відображаються у вигляді списку. `descriptions` — Звуковий опис того, що відбувається в відео для тих випадків, коли зображення недоступно або для сліпих людей. `metadata` — Призначені для використання скриптами і не відображаються в браузері. `subtitles` — Призначені для дублювання звукової доріжки фільму у вигляді тексту мовою оригіналу для глухих людей. Також можуть містити переклад на інші мови для тих, хто не знайомий з мовою оригіналу. Текст субтитрів виводиться поверх відео.
`label`	Назва доріжки для відображення. Якщо цей атрибут не вказаний, браузер стане використовувати значення, яке застосовується у нього за замовчуванням.
`srclang`	Мова доріжки

Форми в HTML

HTML-форми використовуються для збору інформації від користувача (введення логіна і пароля, даних користувача при реєстрації, опитування і т.д.).

Основою форми є тег-контейнер <**form**>, який може містити опис різного роду елементів вводу: текстові поля, списки, перемикачі та ін.

Розглянемо найпростіший приклад форми.

Загальний вигляд	`<form >`
	`. . . опис елементів вводу даних . . .`
	`</form>`

Приклад	`<form action="mailto:brila.andrij@ukr.net" method="get">` `<label for="login"> Login </label>` `<input type="text" id="login">  ` `<label for="password"> Password </label>` `<input type="password" id="password">  ` `<input type="submit" value="Увійти">` `</form>`
На екрані	Login Ivan Password ••••• Увійти

Атрибути тегу <**form**>

Атрибут	Значення
`accept-charset`	Містить перелік імен таблиць кодування (розділених комами) даних, які надсилаються на сервер. `<form accept-charset="ISO-8859-1">`.
`action`	Вказує URL адресу обробника форми на сервері. Якщо необхідно дані опрацювати на стороні клієнта, то у якості значення необхідно вказати значення `#`. Якщо необхідно дані надіслати на пошту, то в якості значення потрібно вказати `mailto:адреса електронної пошти` `<form action="mailto:адрес вашей электронной почты" enctype="text/plain"></form>`
`autocomplete`	Включає або відключає автозаповнення текстових полів форми. Можливі значення `on` – увікнено; `off` – вимкнено.
`enctype`	Вказує принцип кодування даних форми, що відправляються на сервер у разі використання методу передачі «`post`». Містить **MIME**-тип даних За замовчуванням – `application/x-www-form-urlencoded` (означає, що дані є списком URL-кодованих значень (при цьому пробіл кодується як `+`, `!` – в `%21`, `і т.д.`; `multipart/form-data` – надсилання файлів;

	text/plain – надсилається звичайний текст.
method	Визначає спосіб передачі даних. Можливі значення: get – використовується при передачі невеликих об'ємів даних. Усі дані будуть передаватися через адресний рядок браузера (після адреси обробника після знаку питання вказують послідовність значень елементів розділених знаком &). адреса обробника ? **ім'я1=значення1** & **ім'я2=значення2** . . . www.anysite.ua/form.php?var1=1&var2=2. post - використовується при передачі великих об'ємів даних та конфіденційної інформації. <form action="action.php" enctype="multipart/form-data" **method="post"**> . . . </form>
name	Задає ім'я форми, яке можна використати при програмних маніпуляціях
novalidate	Наявність цього атрибуту відключає перевірку значень при надсиланні даних форми.
target	Вказує вікно, у яке буде надіслано інформацію: _blank, _self, _parent,_top

Мітки для елементів форми

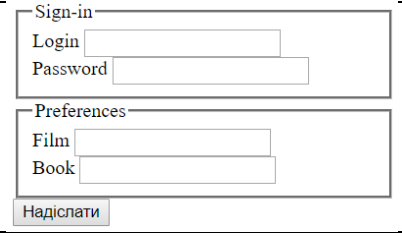
Мітки для елементів вводу задаються за допомогою тегів <**label**>...</**label**>. При кліку на мітку курсор автоматично встановлюється у пов'язаний із ним елемент вводу.

Загальний вигляд	**<label for=**"id елемента вводу"> … текст мітки … </label> **< елемент вводу із заданим id>** або ж без використання атрибуту for, тоді відповідний елемент повинен знаходитися всередині тегу <label> **<label>** … текст мітки … **< відповідний елемент вводу>** **</label>**
Приклад	<label for="**film**"> Film </label> <input type="text" **id="film"**> <label> Book **<input type="text">** </label>
На екрані	Film Book

Групування елементів форми

Іноді елементи форми містять декілька груп даних, які об'єднані за деякою ознакою. В цьому випадку дані можна групувати у окремі логічні блоки за допомогою тегу <**fieldset**>**...**</**fieldset**>.

Розглянемо приклад використання

Загальний вигляд	<form> <fieldset> . . . елементи групи 1 . . . </fieldset> <fieldset> . . . елементи групи 2 . . . </fieldset> </form>
Приклад	<form action="mailto:brila.andrij@ukr.net" method="post" enctype="text/plain"> <fieldset> <legend>Sign-in</legend> <label for="login"> Login </label> <input type="text" id="login"> <label for="password"> Password </label> <input type="password" id="password"> </fieldset> <fieldset > <legend>Preferences</legend> <label for="film"> Film </label> <input type="text" id="film"> <label for="Book"> Book </label> <input type="text" id="book"> </fieldset> <input type="submit" value="Надіслати"> </form>
На екрані	

Атрибути тегу <fieldset>

Атрибут	Значення
disabled	Блокує поля форми в групі. <fieldset disabled>.
form	Пов'язує групу з формою.
name	Задає ім'я для fieldset. Аналог атрибуту id.
title	Додає пояснювальний текст

Поля форми

Переважна більшість полів форми створюється з використанням тегу <**input**> і відповідним значенням його атрибуту **type**.

Атрибут	Значення
accept	Встановлює фільтр на типи файлів, які можна відправити через поле завантаження файлів.

	Можливі значення атибуту: accept=".gif", accept=".pdf", accept=".doc" audio/* – дозволяє завантажувати аудіо-файли; video/* – дозволяє завантажувати відео-файли; image/* – дозволяє завантажувати графічні зображення; media_type – вказує на тип файлу. <input type="file" accept=".gif">
alt	Задає альтернативний текс для графічних зображень <input type="image">.
autocomplete	Дозволяє автозаповнення: on — увімненено off — вимкнено.
autofocus	Дозволяє задати елемент, який автоматично одержує фокус.
checked	Задає значення за замовчуванням (вибрано чи ні) для елементів вибору (type="checkbox" та type="radio").
disabled	Деактивує елемент (не можна редагувати і копіювати).
form	Визначає форму (форми) до яких належить елемент вводу.
formaction	Містить url обробника (може бути використатаним тільки для type="submit" і type="image"). Атрибут перевизначає значення атрибуту action, який визначено у формі.
formenctype	Визначає спосіб кодування даних для надсилання. Перевизначає значення атрибуту enctype форми (для type="submit" и type="image"). Можливі значення: application/x-www-form-urlencoded — значення за замовчуванням (url - кодування). multipart/form-data – символи не кодуються; text/plain — звичайний текст (пробіл заміняємо на +, спецсимволи не кодуються.
formmethod	Задає метод, який використовується при надсиланні даних (використовується для type="submit" и type="image" і перевизначає атрибут, заданий у формі). Можливі значення : get, post.
formnovalidate	Відключає перевірку даних при надсиланні (перевизначає атрибут novalidate форми).
formtarget	Визначає де буде відображено відповідь після надсилання форми (перевизначає атрибут target форми). Можливі значення : _blank, _self, _parent, _top, framename.
height width	Задає висоту та ширину поля у пікселях. <input type="image" src="img_submit.gif" height="50">.
list	Містить id елемента-списку варіантів (<**datalist**>)
max	Містить максимально можливе значення елемента вводу (для елементів типу number, range, date, datetime, datetime-local, month, time, week).
maxlength	Задає максимальну кількість символів у полі вводу.
min	Задає мінімальне значення для елемента вводу
multiple	Дозволяє здійснити введення декількох значень, які розділені комою.
name	Задає ім’я атрибута (аналог id).

pattern	Дозволяє задати регулярний вираз. Приклад: pattern="[a-z]{3}-[0-9]{3}" квадратні дужки – діапазон можливих значень; фігурні дужки – кількість обов'язкових елементів..
placeholder	Текст (довідкова інформація), який відображається у полі елемента вводу до введення інформації.
readonly	Забороняє змінювати значення елемента вводу.
required	Визначає, що елемент вводу є обов'язковим для заповнення.
size	Задає ширину поля у символах (за замовчуванням 20).
src	Вказує URL-адресу зображення для використання її в якості кнопки відправки (для «type =" image "»).
step	Вказує величину зміни числового значення (икористовується для елементів числового вводу
type	button – задає кнопку (натиснувши, на яку можна ініціалізувати деяку команду); checkbox – дозволяє створити елемент множинного (можна вибрати декілька елементів) вибору color – дозволяє створити елемент для задання кольору; date – дозволяє ввести дату у форматі день.місяць.рік; datetime-local – дозволяє здійснювати введення дати і часу; email –озволяє здійснювати введення адреси електронної пошти; file – дозволяє здійснювати введення шляху до файлу, який необхідно завантажити; hidden – дозволяє створити прихований елемент; image – дозволяє створити кнопку, на якій можна відобразити зображення; month – дозволяє здійснювати введення року і місяця; number – дозволяє здійснювати введення числа (при цьому можна вказати мінімальне, максимальне значення та крок зміни – min, max, step): password – дозволяє створити поле вводу для введення паролю (символи при вводі заміняються на символи-маркери). radio – створює елемент одиночного вибору (при наявності декількох пунктів у всіх повинно бути однакове значення атрибуту name <label> Автомобіль <input type="radio" **name="transport"** value="Auto"> </label> <label> Автобус <input type="radio" **name="transport"** value="Bus"> </label> <label> Поїзд <input type="radio" **name="transport"** value="Train"> </label>); range – створює елемент вводу – повзунок (при заданих максимальному і мінімальному значеннях); reset – створює кнопку, за допомогою якої можна очистити поля вводу;

	`search` – позначає поле пошуку, по замовчуванню поле вводу має прямокутну форму. Пошук: `submit` – створює кнопку, при натисненні на яку здійснюється надсилання даних; `text` – створює елемент вводу для одного рядка тексту; `time` – створює елемент для введення часу у 24-часовому форматі. `url` – створює елемент для введення URL-адреси. `week` – дозволяє ввести номер тижня
`value`	Задає текст, який відображається у елементі вводу (кнопка чи текстовий елемент вводу).
`width`	Значення атрибута містить кількість пікселів. Дає змогу задати ширину полів форми.

Текстове поле багаторядкового вводу

Розглянемо приклад використання

Загальний вигляд	`<textarea>...</textarea>`

Приклад	`<textarea>`
	`...`
	`вводьте текст тут`
	`...`
	`</textarea>`

На екрані	... вводьте текст тут ...

Атрибути тегу <**textarea**>

Атрибут	Значення
`autofocus`	Автоматично встановлюється при відкритті форми
`cols` `rows`	Встановлює ширину та висоту елементу вводу (кількість символів для відображення)
`disabled`	Робить елемент вводу неактивним
`form`	Містить `id` форми (або їх список), до якої він належить.
`maxlength`	Задає максимальну кількість символів, яка може бути введена за допомогою елемента вводу
`name`	Задає ім’я елемента вводу
`placeholder`	Містить довідкову інформацію, яка відображається до того, як користувач почне вводити текст.
`readonly`	Забороняє змінювати текст
`required`	Робить елемент вводу обов’язковим для заповнення
`wrap`	Дозволяє задати, чи потрібно зберігати переноси рядків при надсиланні тексту. Можливі значення: `hard` – зберігає переноси (обов’язковим є задання атрибуту `cols`); `soft` – не зберігає переноси.

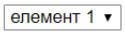
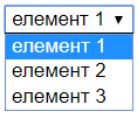
Елемент вводу список

Для введення одного або декількох значень із наперед заданого списку можна використати тег <**select**>...</**select**>, в середині якого необхідно перерахувати можливі значення (включені за допомогою тегів <**option**>...</**option**>).

Розглянемо приклад використання даного елемента

Загальний вигляд	`<select>`
	`<option>...елемент 1 ...</option>`
	`<option>...елемент 1 ...</option>`
	`. . .`
	`</select>`

Приклад	`<select>`
	`<option>елемент 1 </option>`
	`<option>елемент 2 </option>`
	`<option>елемент 3 </option>`
	`</select>`

На екрані	До активізації елемента  Після активізації елемента 

Атрибути тегу <**select**>

Атрибут	Значення
`autofocus`	Автоматична активізація елемента при завантаження сторінки
`disabled`	Деактивує елемент вводу
`form`	Задає id форми, до якої належить елемент
`multiple`	Надає можливість множинного вибору
`name`	Задає ім'я елемента вводу
`required`	Вказує, що елемент є обов'язковим для введення
`size`	Задає кількість елементів, які одразу відображаються на в елементі вводу

Атрибути тегу <**option**>

Атрибут	Значення
`disabled`	Дозволяє зробити елемент вводу неактивним
`label`	Дозволяє задати скорочену версію елементу вводу
`selected`	Дозволяє відображати елемент списку як вибраний автоматично при завантаженні сторінки
`value`	Містить значення, яке буде надіслено на сервер при надсиланні форми

Групування елементів списку вибору

Групування значень у елементі вибору можна здійснити за допомогою тегу <**optgroup**>

Розглянемо приклад

Загальний вигляд	<select> <optgroup label="… мітка групи 1 … "> <option>...елемент ...</option> <option>...елемент ...</option> </optgroup> <optgroup label="… мітка групи 2… "> <option>...елемент ...</option> <option>...елемент ...</option> </optgroup> . . . . </select>
Приклад	<select> <optgroup label="Овочі"> <option>Кавун</option> <option>Буряк</option> </optgroup> <optgroup label="Фрукти"> <option>Груша</option> <option>Черешня</option> </optgroup> </select>
На екрані	Кавун Овочі Кавун Буряк Фрукти Груша Черешня

Атрибути <optgroup>

Атрибут	Значення
disabled	Деактивує групу елементів вибору
label	Задає назву заголовку групи

Кнопки

Кнопки можуть бути створені за допомогою тегу <**input**> (<input type="submit"></input>, <input type="image">, <input type="reset">, <input type="button">) або за допомогою тегу <**button**>...</**button**>.

Розглянемо приклад

Загальний вигляд	<button> ... контент, що відображається на кнопці ... </button>
Приклад	<button> <img src="Auto.png"> </button>
На екрані	

Атрибути тегу <**button**>

Атрибут	Значення
autofocus	Встановлю автоматичний фокус при завантаженні сторінки
disabled	Деактивує кнопку
form	Містить id форми, до якої відноситься елемент
formaction	Задає url-адресу обробника (перевизначає атрибут action форми, тільки при заданні type="submit").
formenctype	Задає тип кодування даних, що надсилаються (перевизначає атрибут enctype форми, тільки при заданні type="submit").
formmethod	Задає метод, який використовується при надсиланні даних (використовується для type="submit" и type="image" і перевизначає атрибут, заданий у формі, тільки при заданні type="submit"). Можливі значення : get, post.
formnovalidate	Відключає перевірку даних при надсиланні (перевизначає атрибут novalidate форми, тільки при заданні type="submit").
formtarget	Визначає де буде відображено відповідь після надсилання форми (перевизначає атрибут target форми). Можливі значення : _blank, _self, _parent, _top, framename.
name	Задає ім'я кнопки.
type	Задає тип кнопки Можливі значення: button – кнопка; reset – очищає елементи форми; submit — кнопка для надсилання даних.
value	Значення за замовчуванням, яке надсилається при натисненні на кнопку

Семантичні елементи HTML5

Семантичні елементи призначені для структурування і групування контенту сторінки. До виходу HTML5 структурування здійснювали за допомогою тегів <**div**>, надаючи кожному із них відповідні значення атрибутів **id** та **class**. Тег <**div**> – блочний елемент, який використовувався виключно для групування елементів. Він не має спеціальних правил стосовно візуального представлення браузерами.

Приклад.

```
<html>
 <head>
 . . . . .
 </head>
 <body>
 <div id="header">
 . . . . . Верхній колонтитул (заголовок сторінки)
 </div>

 <div id="main">
 . . . . . Основна частина сторінки
      </div>

 <div id="footer">
 . . . . . Нижній колонтитул
 </div>
 </body>
</html>
```

З появою HTML5 для структурування і групування використовують ряд нових тегів.

Структурування контенту сторінки

Для формування структури сторінки можуть бути використані такі теги як: **<header>, <main>, <footer>.**

Приклад.

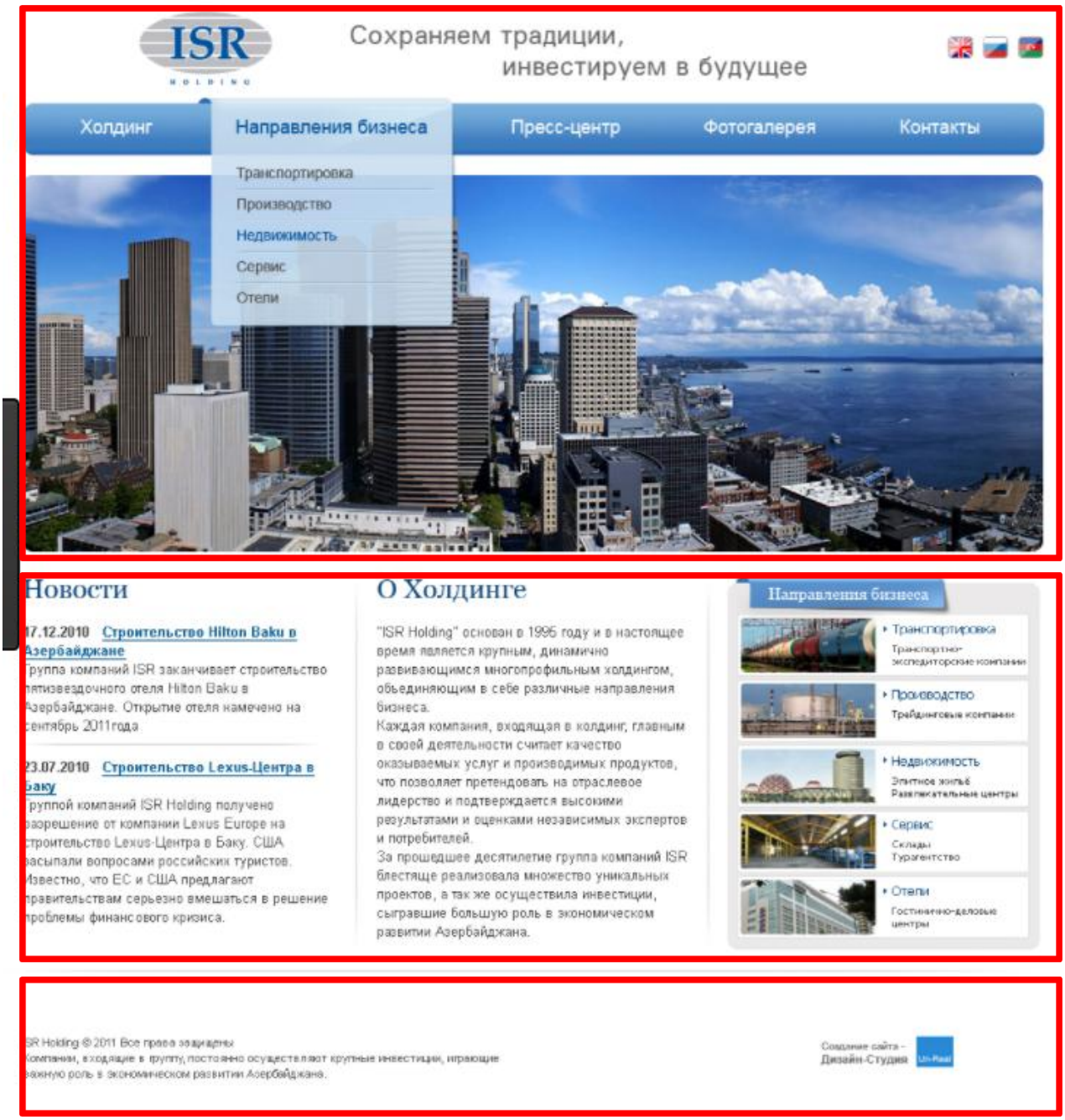
```
<html>
 <head>
 . . . . .
 </head>
 <body>
 <header>
 . . . . . Верхній колонтитул (заголовок сторінки)
 </header>

 <main>
 . . . . . Основна частина сторінки
      </main>

 <footer>
 . . . . . Нижній колонтитул
 </footer>
```

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

Як і при використанні тегів <**div**> браузери не мають якихось наперед визначених правил відображення і їх графічне оформлення повинно здійснюватися розробником. Зауважимо також, що ці теги можуть бути використані не тільки для структурування усієї сторінки, а й структурування окремих її складових.



Розглянемо ці теги більш детально.

Тег <**header**>

Тег <**header**> використовуються як контейнер для групування складових заголовку елемента, до якого він був включений. Для сторінки тег <**header**> як правило містить:

- набір навігаційних посилань;

- теги заголовків (<h1> - <h6>);
- логотип компанії або графічне зображення;
- інформацію про авторство;
- та ін.

Приклад. Розглянемо приклад використання тега <**header**>, який містить навігаційні посилання

```
<html>
 <head>
 <title></title>
 </head>
 <body>
 <header>
 Вітаємо на сайті найкращого магазину
 <nav>
 <a href='#'>Головна</a>
 <a href='#'>Каталог</a>
 <a href='#'>Контакти</a>
 </nav>
 </header>
 . . . .
 </body>
</html>
```

Тег <**header**> може зустрічатися декілька разів в одному документі, але не може знаходитися у тегах: <**footer**>, <**address**>, іншому <**header**>.

Тег **<main>**

Тег <**main**> використовується як контейнер для групування складових основної частин (основного змісту) сторінки (тега <**body**>).

Зміст всередині елемента <**main**> повинен бути унікальним для документа (зустрічатися тільки один раз). Він **не повинен** містити будь-який зміст, який повторюється на вашому сайті, наприклад:

- бічні панелі
- навігаційні посилання
- інформація про авторські права
- логотип сайту
- пошукові форми

Тег <**main**> може зустрічатися тільки один раз і не може бути включених у теги : <**article**>, <**aside**>, <**footer**>, <**header**>, <**nav**>.

Тег **<footer>**

Тег <**footer**> використовується як контейнер для групування складових, які входять до нижнього колотитула елемента, у який він був поміщений. Як правило даний тег містить таку інформацію:

- інформацію про авторство;

- інформацію про авторські права;
- контактну інформацію;
- карту сайту;
- пов'язані з ним документи.

Тег <**footer**> може зустрічатися декілька разів у документі (для сторінки, для статті, для цитування).

Елементи групування контенту

У HTML5 також додано ряд елементів для групування контенту. Такого роду теги дозволяють об'єднувати логічно пов'язані між собою частини інформації, і спрощує подальше форматування сторінки. Розглянемо деякі з них.

Тег **<section>**

Тег <**section**> задає розділ документа (логічну частину документа, яка є окремою логічною одиницею). Як правило використовується для групування тематичного вмісту :

- змісту;
- розділи статей і публікацій;
- перелік планових операцій;
- та ін.

Приклад. Наведемо приклад використання даного тегу для розділення окремих розділів статті.

```
<article>
 <h1>...заголовок першого розділу …</h1>
 <section>
 ... інформація першого розділу ...
 </section>
 <h1>... заголовок другого розділу . . .</h1>
 <section>
 ... інформація другого розділу . . .
 </section>
</article>
```

Тег **<aside>**

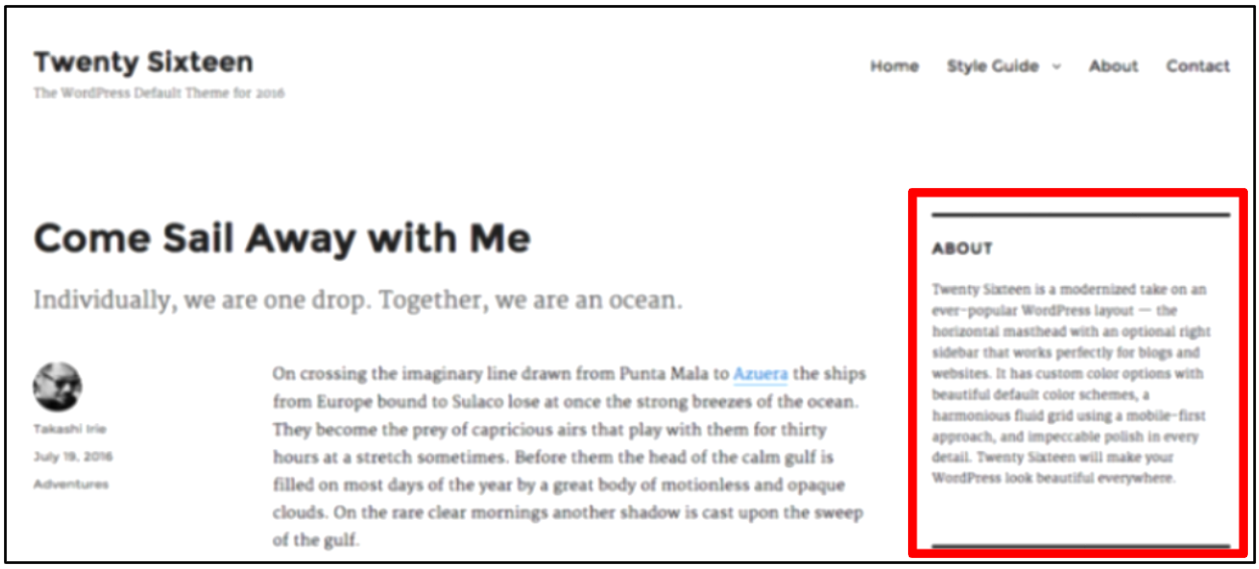
HTML тег <**aside**> використовується для групування додаткової інформації стосовно іншої частини інформації. Як правило використовується для розміщення:

- рубрик;
- посилань на архів;
- міток;
- рекламних блоків;
- архівних записів;
- цитат;
- та ін.

Приклад.

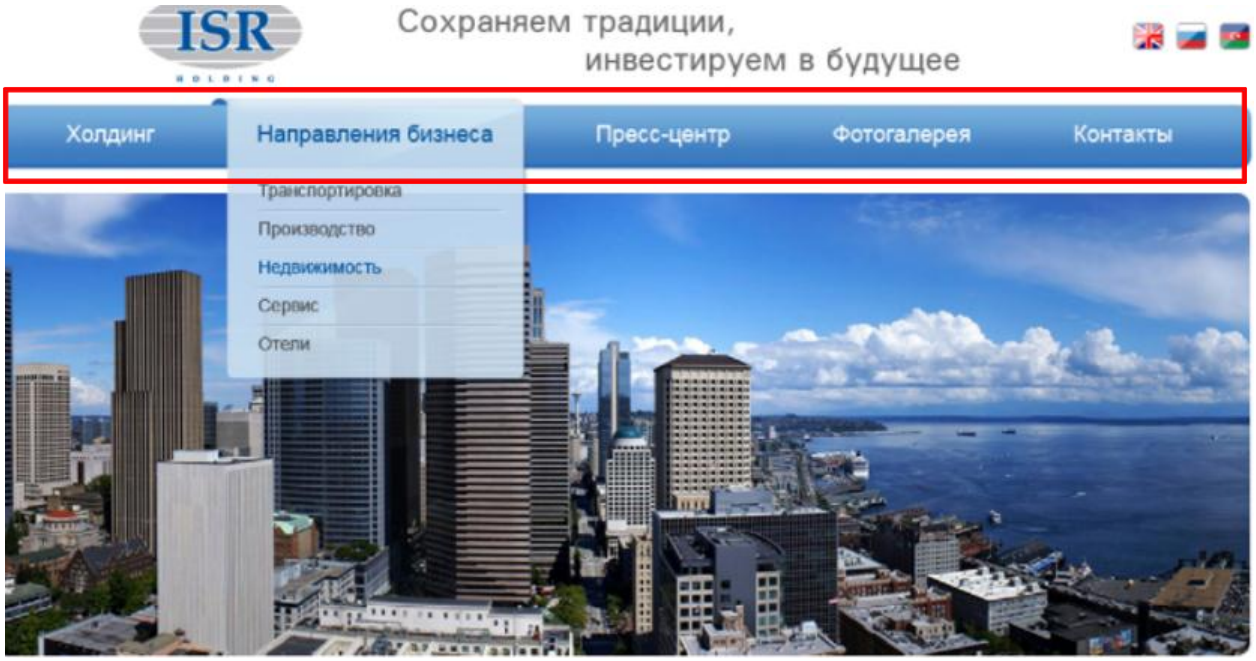
```
<aside>
 <h2>...</h2>
 <p>...</p>
</aside>
```

```
<aside>
 <h2>...</h2>
 <p>...</p>
 <blockquote>
 <p>...<cite>...</cite>...</p>
 <p>...</p>
 </blockquote>
</aside>
```



Тег **<nav>**

HTML тег <**nav**>, зазвичай, визначає, групу посилань, що є навігацією по сайту. Такі посилання зазвичай ведуть на інший розділ, головну сторінку, контакти і інше. Контент, що містить тег <**nav**>, призначений тільки для навігаційних посилань, тому не всі посилання можуть бути поміщені в цей контейнер. Якщо на сторінці кілька блоків посилань, то в <**nav**> зазвичай поміщають пріоритетні посилання. Забороняється вкладати тег <**nav**> всередину <**address**>.



Приклад.

```
<nav>
 <ul>
 <li><a>...</a></li>
 <li><a>...</a></li>
 <li><a>...</a></li>
 </ul>
</nav>
```

Приклад. Посилання у вигляді абзаців

```
<nav>
 <p><a href="…">...</a></p>
 <p><a href="…">...</a></p>
</nav>
```

Приклад. Блок навігаційних посилань з заголовками

```
<nav>
 <h2>...</h2>
 <ul>
 <li><a>...</a></li>
 <li><a>...</a></li>
 <li><a>...</a></li>
 </ul>
</nav>
```

Тег **<article>**

Тег <**article**> використовується для розміщення важливих розділів контенту всередині веб-сторінки. Наприклад, у блозі кожен окремий пост являє собою значимий фрагмент контенту і саме цей контент має бути виділений за допомогою елемента <**article**>.

Зазвичай HTML тег <**article**> містить:

- повідомлення на форумі;
- повідомлення в блозі;
- коментар;
- статтю;

та ін.

Даний тег може зустрічатися декілька разів і, як правило, містить заголовок.

Приклад.

```
<article>
 <header>
 <h2>... заголовок статті ...</h2>
 </header>
 <p>... основна інформація . . .</p>
 <footer>
 Категорія <a href="">Музыка</a>.
 <a href="">0 коментарів</a>
 </footer>
</article>
```

Тег <**address**>

Тег **<address>** визначає контактну інформацію автора/власника документа чи статті і може включати в себе будь-які HTML елементи, на кшталт посилань, тексту, блоків і т. п. Вважається, що пошукові системи будуть аналізувати вміст цього елементу для збору інформації про авторів сайтів.

Якщо елемент **<address>** знаходиться всередині елемента <**body**>, він визначає контактну інформацію для даного документа.

Якщо елемент **<address>** знаходиться всередині елемента <**article**>, він визначає контактну інформацію для цієї статті.

Текст в елементі **<address>** зазвичай відображається курсивом. Більшість браузерів додасть розрив рядка до і після цього елемента. Якщо ця особливість не потрібна, використовуйте стилі для зміни шрифту, за це відповідає CSS властивість **display**.

Тег **<address>** відноситься до маловідомих тегів, які рідко використовуються середньостатистичним веб-розробником. Його придумали в 1995 році разом зі специфікацією HTML 2.0. Можна припустити, що все просто і очевидно – address переводиться з англійського як адреса тому і слід його використовувати для позначення адрес на веб сторінках. Але, якщо не гадати, заглянути в специфікацію, стає очевидним, що таке припущення помилкове.

Рекомендації W3C до елемента address:

- Елемент **address** може бути використаний авторами документа або його частини для надання контактної інформації, щоб бажаючі могли зв'язатися з автором матеріалу.

З специфікації можна зробити висновок, яку саме інформацію може містити цей тег:

- ПІБ автора матеріалу
- його контактні дані: поштові адреси, номери телефонів, адреси сайтів і електронної пошти, ICQ і так далі
- дата створення (останньої модифікації) матеріалу

- Порада: тег **<address>** не повинен використовуватися для опису поштової адреси, якщо вона не є частиною контактної інформації.
- Порада: Елемент **<address>**, як правило, використовується разом з іншою інформацією в елементі <**footer**>.

Тег <**figure**>

HTML тег <figure> використовується для групування будь-яких елементів, наприклад, зображень і підписів до них, ілюстрацій, фрагментів коду, схем, графіків, діаграм та іншого. Є блочним елементом. Підпис до елемента <**figure**> додають з використанням тега <**figcaption**>.

Приклад.

```
<figure>

 <img src="picture.jpg" alt="Осінь">

 <figcaption> Осінь у місті </figcaption>

</figure>
```

Тег <**time**>

HTML тег <**time**> визначає час та дату в форматі легкому для читання. Це полегшує клієнтським додаткам додавати події в свій календар. Дата і час можуть бути вказані безпосередньо всередині контейнера <**time**>, або задаватися через атрибут **datetime**. При цьому, якщо дата повинна бути зчитана іншими додатками, то дату задають у форматі YYYY-MM-DD, а час задається у форматі HH:MM з додаванням префікса T (**time**)/

Приклад.

```
<time datetime="2017-10-25"> 25 Жовтня 2017</time>

<time datetime="2017-10-25T12:00"> 25 Жовтня 2017</time>
```

Тег <**mark**>

HTML тег <**mark**> позначає текст як виділений. У браузерах, зазвичай, фоновий колір тексту всередині тага <**mark**> виділяється жовтим кольором. Цей тег не є семантичним, а лише виділяє текст, щоб користувач звернув на нього увагу.

Приклад.

```
<mark>текст</mark>
```

Список використаної літератури

1. Daniel Woolston. Pro Ajax and the .NET 2.0 Platform. – М.: Инфра-М, 2017. – 143 с.
2. Колисниченко Д.Н. Самоучитель PHP 5. – СПб.: Наука и Техника, 2015. – 640 с.
3. Кузнецов М. В., Симдянов И. В., Голышев С. В. PHP 5. Практика разработки Web-сайтов. – СПб.: БХВ-Петербург, 2015.
4. Менаске Д., Алмейда В. Производительность Web-служб. Анализ, оценка и планирование.. – СПб.: ООО «ДиаСофтЮП», 2013. – 480 с.
5. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL и JavaScript. – СПб.:Питер, 2016. – 496 с.
6. Розенфильд Л., Морвиль П. Информационная архитектура в Интернете. – М.: ООО “И.Д. Вильямс”, 2016. – 544 с.
7. Хабибулин И.Ш. Разработка Web-служб средствами Java. – СПб.: БХВ-Петербург, 2013. – 400 с.
8. Хольцнер С. HTML5 за 10 минут. – 5-е изд. – М. : Издательский дом «Вильямс», 2011. – 240 с.
9. Шмитт К. Симпсон К. HTML5. Рецепты программирования. – СПб.: Питер, 2012. – 288 с.
10. Эрик А. Мейер CSS-каскадные таблицы стилей: подробное руководство. – М.: "Символ", 2006. – С. 576.

Зміст

Вступ 3

Мова розмітки гіпертексту HTML 4

Основні елементи 6

Форматування тексту 9

Списки HTML 16

Таблиці в HTML 24

HTML-посилання 33

Вставка графічних зображень у HTML-документ 39

Вставка аудіо-контенту 45

Вставка відео-контенту 45

Форми в HTML 49

Кнопки 57

Семантичні елементи HTML5 58

Список використаної літератури 68

Для нотаток

Для нотаток

Відповідальний за випуск: завідувач кафедрою системного аналізу і теорії оптимізації
к. ф.-м. н., доц. Кузка О.І.

Автори: к. ф.-м. н., доц. Брила А.Ю.,
к. ф.-м. н., доц. Глебена М.І.,
к. ф.-м. н., доц. Млавець Ю.Ю.,
ст.викл., Ломага М.М.

Рецензенти: к.ф.-м.н., доц. Погоріляк О.О.,
к. ф.-м.н., доц. Тегза А.М.

ВСТУП ДО WEB-ПРОГРАМУВАННЯ. ОСНОВИ HTML

Методичні вказівки для самостійної роботи студентів математичного факультету з дисципліни «Вступ до web-програмування»