

Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна

О. Г. Сузікова

ВЕБ ТЕХНОЛОГІЇ В ІНФОРМАЦІЙНОМУ ОБМІНІ

Методичні рекомендації
до практичних занять для здобувачів вищої освіти першого (бакалаврського)
рівня з дисципліни «Обробка, зберігання та передача даних в сучасних
інформаційних технологіях» за спеціальністю «Прикладна математика»

У 3 частинах

Частина 1: HTML

Електронний ресурс

Харків – 2025

Рецензенти:

Ю. В. Ромашов – доктор технічних наук, доцент, професор кафедри комп'ютерно-інтегрованих технологій, автоматизації та робототехніки Харківського національного університету радіоелектроніки;

С. М. Севидов – кандидат фіз.-мат. наук, доцент кафедри математичного моделювання та аналізу даних ННІ комп'ютерних наук та штучного інтелекту Харківського національного університету імені В.Н. Каразіна.

*Затверджено до розміщення в мережі Інтернет рішенням Науково-методичної ради
Харківського національного університету імені В. Н. Каразіна
(протокол № 11 від 25 червня 2025 року)*

Сузікова О. Г.

С 89

Веб технології в інформаційному обміні : методичні рекомендації до практичних занять для здобувачів вищої освіти першого (бакалаврського) рівня з дисципліни «Обробка, зберігання та передача даних в сучасних інформаційних технологіях» за спеціальністю «Прикладна математика». У 3 ч. Частина 1: HTML [Електронний ресурс] / О. Г. Сузікова. – Харків : ХНУ імені В. Н. Каразіна, 2025. – (PDF 37 с.)

Методичні рекомендації містять початкові відомості про HTML. До кожного розділу наведені питання для самоконтролю та практичні завдання.

Методичні рекомендації призначені для здобувачів вищої освіти першого (бакалаврського) рівня, які вивчають дисципліну «Обробка, зберігання та передача даних в сучасних інформаційних технологіях».

УДК 004.439 (072)

© Харківський національний університет
імені В. Н. Каразіна, 2025

© Сузікова О. Г., 2025

ЗМІСТ

Вступ	5
Контекст	6
1. Знайомство з мовою HTML	7
1.1. Загальні відомості про HTML	7
1.2. Теги: атрибути та їх значення	8
1.3. Питання для самоконтролю	9
1.4. Практичне завдання	9
2. Загальна структура HTML-документа	10
2.1. Основні розділи HTML-документа та їх призначення	10
2.2. Питання для самоконтролю	15
2.3. Практичне завдання 1	15
3. Типографіка	16
3.1. Типографіка — заголовки, відступи та шрифти	16
3.2. Питання для самоконтролю	18
3.3. Практичне завдання	19
4. Кольорове оздоблення	19
4.1. Кольорове оздоблення та системи представлення кольорів	19
4.2. Питання для самоконтролю	22
4.3. Практичне завдання	22
5. Представлення інформації списками	23
5.1. Списки	23
5.2. Питання для самоконтролю	25
5.3. Практичне завдання	25
6. Посилання	26
6.1. Зовнішні та внутрішні посилання	26
6.2. Питання для самоконтролю	29
6.3. Практичне завдання	30
7. Використання мультимедіа, частина 1	31

7.1.Додавання зображень	31
7.1.Питання для самоконтролю	32
7.2.Практичне завдання	32
8. Використання мультимедіа, частина 2	33
8.1.Анімація та звук	33
8.2.Питання для самоконтролю	34
8.3.Практичне завдання	35
Список літератури	36

Вступ

HTML (Hyper Text Markup Language) є фундаментальною технологією у сфері веб-розробки та інформаційних технологій загалом. Це основа будь-якого веб-сайту, оскільки HTML визначає структуру веб-сторінки, організовуючи контент у заголовки, абзаци, списки, посилання, таблиці та інші елементи. Без знання HTML важко розуміти принципи роботи веб-додатків, навіть якщо ви не є програмістом, а працюєте в суміжних сферах, як-от управління проектами чи аналітика.

У сучасному IT-середовищі HTML використовується в поєднанні з CSS (каскадними таблицями стилів) та JavaScript, що дозволяє створювати динамічні та інтерактивні веб-застосунки. Навіть базове розуміння HTML допомагає фахівцям різних напрямків – від тестувальників до менеджерів – ефективніше комунікувати з розробниками та розбиратися в основах роботи веб-технологій.

Таким чином, HTML є не лише мовою розмітки, а й невід’ємною частиною цифрової грамотності. Уміння читати та редагувати HTML-код допомагає адаптувати контент під різні платформи, розуміти принципи SEO та впливати на якість користувацького досвіду. Це знання розширює можливості фахівців, навіть якщо їхня основна діяльність не пов’язана безпосередньо з веб-розробкою.

Методичні рекомендації містять початкові відомості про HTML. До кожного розділу наведені питання для самоконтролю та практичні завдання. Рекомендуємо обов’язково виконувати ці завдання, щоб швидше освоїтися у практичному застосуванні HTML.

Контекст

HTML (Hyper Text Markup Language) — це стандартна мова розмітки документів, яка використовується для створення та структурування веб-сторінок. HTML дозволяє визначати структуру та вміст веб-сторінки за допомогою різноманітних тегів і атрибутів, що задають текст, зображення, посилання, форми та інші елементи.

Більшість контенту на сайтах має спільну структуру та складається з параграфів і заголовків, незалежно від того, чи читаєте ви головну сторінку, чи опис товару, чи наукову статтю, чи електронну книгу тощо.

Впорядкований контент робить читання більш легким та приємним.

Мова HTML була розроблена британським вченим Тімом Бернерс-Лі приблизно в 1986-1991 роках в стінах ЦЕРНу в Женеві в Швейцарії. HTML створювалась як мова для обміну науковою і технічною документацією, придатний для використання людьми, які не є фахівцями в області верстки. HTML успішно впоралась з проблемою складності шляхом визначення невеликого набору структурних і семантичних елементів – дескрипторів. Інша назва дескриптора – їх також часто називають «тегами».

За допомогою HTML можна легко створити відносно простий, але красиво оформлений документ. Крім спрощення структури документа, в HTML додано підтримку гіпертексту. Мультимедійні функції додані пізніше. Спочатку мова HTML була задумана і створена як засіб структурування та форматування документів без їх прив'язки до засобів відтворення (відображення).

В ідеалі, текст з розміткою HTML повинен був без стилістичних та структурних спотворень відтворюватися на обладнанні з різною технічною оснащеністю (кольоровий екран сучасного комп'ютера, монохромний екран

органайзера, обмежений за розмірами екран мобільного телефону або пристрою і програми голосового відтворення текстів). Однак сучасне застосування HTML дуже далеко від його початкової задачі.

Наприклад, тег `<table>` призначений для створення в документах таблиць, але іноді використовується і для оформлення розміщення елементів на сторінці. З плином часу основна ідея платформної незалежності мови HTML була принесена в жертву сучасним потребам в мультимедійному і графічному оформленні.

Текстові документи, що містять розмітку на мові HTML (такі документи зазвичай мають розширення `.html` або `.htm`), обробляються спеціальними додатками, які відображають документ в його форматованому вигляді.

Такі додатки, звані «браузерами» або «інтернет-оглядачами», зазвичай надають користувачеві зручний інтерфейс для запиту веб-сторінок, їх перегляду (і виведення на інші зовнішні пристрої) і, при необхідності, відправки введених користувачем даних на сервер.

Найбільш популярними на сьогоднішній день браузерами є Google Chrome, Mozilla Firefox, Opera, Edge і Safari.

1. Знайомство з мовою HTML

1.1. Загальні відомості про HTML

HTML містить типи елементів, що представляють параграфи, гіпертекстові посилання, списки, таблиці, зображення і т. д.

HTML – мова тегів та мова розмітки документів. Будь-який документ на мові HTML являє собою набір елементів, причому початок і кінець кожного елемента позначається спеціальними позначками – тегамі. В

результаті кожне оголошення типу елемента зазвичай описує три частини: початковий тег, вміст і кінцевий тег.

Елементи можуть мати атрибути. Також елементи можуть бути порожніми, коли вони не містять ніякого тексту та інших даних.

У HTML кожен абзац укладено в елемент `<p>`

```
<p>Це параграф.</p>
```

Кожен заголовок має елемент заголовка `<h1>` (en-US):

```
<h1>Я заголовок історії.</h1>
```

- ✓ Є шість рівнів (елементів) заголовка: `<h1>` , `<h2>`, `<h3>` , `<h4>` , `<h5>` та `<h6>` (en-US).
- ✓ Кожен елемент представляє різний рівень контенту документі
 - `<h1>` представляє головний заголовок,
 - `<h2>` представляє підзаголовки,
 - `<h3>` представляє підзаголовки і так далі.

Наприклад, у книзі `<h1>` представлятиме назву, `<h2>` – назву кожного розділу, `<h3>` позначатиме підзаголовки у кожному підрозділі тощо.

1.2.Теги: атрибути та їх властивості

Тег означає тип елемента документа; `<p>`, `<h1>` – це теги.

Теги можуть бути парними та непарними. Парний містить частину, що відкриває, та частину, що закриває.

```
<i>Я те, що треба виділити курсивом, я парний.</i>
```

Непарний не містить частину, що закриває.

```
<br>
```

Це тег переходу на наступний рядок. Він непарний.

В тегів є атрибути або властивості. Атрибути мають певний стандарт опису «властивість = значення», вони уточнюють вживання відповідного тегу:

```
<тег атрибут="значення">...</тег>
```

Приклади атрибутів тегу p – колір, тип шрифту, розмір шрифту, фоновий колір, розташування на сторінці (color, font-family, font-size, background...)

1.3. Питання для самоконтролю

Що таке тег?

Яка інша назва тегу?

Яке призначення тегів?

Які бувають теги?

Що таке атрибут та його значення?

1.4. Практичне завдання

1. Ознайомитися з базовою інформацією про мову HTML.
2. Ознайомитися самостійно з основними тегами мови.
3. Скласти таблицю основних тегів і їх призначень.

4. Відкрити у вікні браузера якусь сторінку в моді HTML коду (CTRL+U) і подивитися на структуру реального документа; спробувати зрозуміти, що і як буде відображено.

2. Загальна структура HTML-документа

2.1. Основні розділи HTML-документа та їх призначення

Документна модель HTML

HTML-документ має чітку структуру, яка визначає, як різні елементи розташовані та взаємодіють один з одним.

HTML документ містить типи елементів, що представляють параграфи, гіпертекстові посилання, списки, таблиці, зображення і т.д. З розділу 1 ми знаємо:

- ✓ Кожне оголошення типу елемента зазвичай описує три частини: початковий тег, вміст і кінцевий тег.
- ✓ Назва елемента з'являється в початковому тегу (<назва-елемента>) і в кінцевому тегі (</назва-елемента>).
- ✓ Деякі елементи HTML допускають відсутність кінцевого тегу.
- ✓ Деякі типи елементів HTML не мають вмісту. Такі порожні елементи ніколи не мають кінцевого тегу.

Додатково:

- ✓ Назви елементів завжди нечутливі до регістру. Елементи можуть мати асоційовані властивості, звані атрибутами, які можуть мати значення (за замовчуванням або встановлюються автором). Пари атрибут / значення з'являються перед кінцевим символом ">" початкового тегу елемента.

- ✓ Будь-яка кількість (допустимих) пар значень атрибута, розділених пробілами, може з'являтися в початковому тегу елемента. Вони можуть з'являтися в будь-якому порядку. За замовчуванням потрібно, щоб всі значення атрибутів були обмежені з використанням подвійних або одинарних лапок, однак в деяких випадках можна встановлювати значення атрибута без використання лапок. Але рекомендується використовувати позначку лапок навіть тоді, коли можна обійтися без них.
- ✓ Назви атрибутів завжди нечутливі до регістру.

Коментарі HTML мають наступний синтаксис:

```
<!-- це коментар -->  
<!-- і це теж коментар,  
що займає більше одного рядка -->
```

Основні розділи документу

Елемент HTML

Опис: визначає початок і кінець HTML-документ.

Початковий тег: не обов'язково

Кінцевий тег: не обов'язково

```
<HTML>  
... елементи head, body і т.п. йдуть тут ...  
</HTML>
```

Елемент HEAD

Опис: містить інформацію про поточний документ, таку як назва, ключові слова та інші дані, які не є вмістом документа.

Початковий тег: не обов'язково

Кінцевий тег: не обов'язково

```
<head>
... елементи head, body і т.п. йдуть тут ...
</ head>
```

Елемент TITLE

Опис: визначає заголовок сторінки або назву вікна. Кожен документ HTML повинен містити елемент TITLE в розділі HEAD. Автори повинні використовувати елемент TITLE для ідентифікації вмісту документа. Оскільки користувачі часто звертаються до документів поза контекстом, автори повинні надавати осмислені заголовки. Заголовок не може містити розмітку (в тому числі і коментарі).

Початковий тег: необхідний

Кінцевий тег: необхідний

<pre><html> <head> <title>Назва</title> ...інші елементи заголовку... </head> ... </html></pre>	<p>В заголовку браузера відображається стрічка: Назва</p>
---	---

Елемент META.

Початковий тег: потрібний

Кінцевий тег: заборонений

Визначення атрибутів:

name = рядок – встановлює ім'я властивості

content = рядок – визначає значення властивості

Метадані:

- ✓ Мова HTML дозволяє авторам уточнювати метадані – інформацію про сам документ, автора, а не про його склад. За це відповідає елемент META
- ✓ Елемент META можна використовувати для ідентифікації властивостей документа (автора, кінцевої дати використання, списку ключових слів і т.д.) і установки значень цих властивостей.
- ✓ Кожен елемент META визначається в розділі HEAD і визначає пару властивість-значення.

Атрибут name ідентифікує властивість, а атрибут content визначає значення властивості.

```
<meta name = "author" content = "Petrenko">
```

Цей тег встановлює значення для властивості Author (автор). Зазвичай META утримує ключові слова, які використовуються пошуковими машинами для підвищення якості і швидкості пошуку.

```
<meta name = "keywords" content = "software">
```

Це оголошення встановлює значення для властивості keywords (ключові слова)

```
<html>
<head>
<title> Назва </title>
```

В заголовку браузера відобразиться текст: Назва

```
<meta name="author"
```

Автором документа є «Автор1»

<pre>content="Автор1"> <meta name="keywords" content="software"> </head> ... </html></pre>	Ключові слова: software
---	-------------------------

Елемент BODY

Елемент BODY в описі відповідатиме за тіло документа. У тілі документа ви побачите вміст документа.

Початковий тег: не обов'язково

Кінцевий тег: не обов'язково

Визначення атрибутів:

- ✓ background = "url" (url – це рядок, що задає шлях в структурі каталогів до файлу) – установка фонові картинки.
- ✓ text = color (значення кольору може задаватися різними засобами, бути або 16-річної числом (яка випереджає знаком # в наступному форматі #RRGGBB, де RR – градація червоного кольору, GG – зеленого, BB – синього), або одним з наступних 16 назв кольору) – встановлює колір тексту (для візуальних браузерів). Тема кольорового оздоблення буде доповнена в одному з наступних завдань.
- ✓ bgcolor = color – установка кольору фону документа.

<pre><html> <head> <title> Назва </title> </head> <body bgcolor="white" text="#000000"> ... тіло документа... </body> </html></pre>	Текст "Назва" на сторінці буде відображатися чорним кольором на білому тлі.
--	---

2.2. Питання для самоконтролю

З яких розділів складається HTML документ?

Яке призначення кожного розділу?

Чи всі розділи відображаються при відтворенні сторінки браузером?

Які атрибути є в тегів основних розділів та які значення вони можуть приймати?

2.3. Практичне завдання

Створимо свій документ index.html

1. На жорсткому диску створити папку з ім'ям HTML.
2. Відкрити текстовий редактор "Блокнот".
3. У вікні блокнота створити документ, що друкує в якості заголовка документа назву Вашої спеціальності.
4. Зберегти документ під ім'ям index.html, обов'язково з розширенням html в папці HTML.
5. Запустити будь-який з браузерів, встановлених в системі, наприклад, Chrome (Пуск - Програми - Chrome).
6. Використовуючи меню Файл - Відкрити, відкрити у вікні браузера свій файл і переконатися, що в рядку заголовка надруковано назву Вашої спеціальності.
7. Перейти у вікно редактора Блокнот і додати у вікні браузера напис «Харківський національний університет імені В.Н.Каразіна».
8. Збережіть зміни.
9. Перейти в вікно браузера, щоб переглянути внесені зміни, необхідно натиснути кнопку "Оновити".
10. Використовуючи метадані, визначити автора документа і ключові слова: назва Вашої спеціальності і назву вашого факультету.

11. Установити білий колір фону документа на свій смак.

12. Встановити фонову картинку, для цього:

- на жорсткому диску знайти файл з розширенням jpg або gif;
- скопіювати знайдений файл в свою папку;
- встановити фонову картинку;

13.Зберегти результати, так як такі завдання спираються на результати попередніх.

3.Типографіка

3.1.Типографіка — заголовки, відступи та шрифти

Заголовки: елементи H1, H2, H3, H4, H5, H6 описують шість рівнів заголовків: від H1 (самий верхній) до H6 (найнижчий).

Початковий тег: необхідний

Кінцевий тег: необхідний

Визначення атрибутів:

- ✓ align = left | center | right | justify – вирівнювання тексту:
- ✓ left – вирівнювання по лівому краю, center – по центру (за замовчуванням), right – по правому краю, justify – вирівнювання по правому і лівому краях:

<code><h1>Заголовок першого рівня</h1></code>	Буде виведено заголовком першого рівня "Заголовок першого рівня"
<code><h2 align="left">Заголовок другого рівня</h2></code>	Буде виведено заголовком другого рівня "Заголовок другого рівня" з вирівнюванням по лівому краю

Зазвичай текст будь-якого документа є структурованим і розділеним на окремі смислові блоки абзаци або параграфи. Для більшої виразності абзаци

можуть виділятися за допомогою кольору, виду і розміру шрифту та інших засобів.

Елемент P - представляє в HTML параграф

Початковий тег: необхідний

Кінцевий тег: не обов'язково

Визначення атрибутів:

align = left | center | right | justify – вирівнювання тексту

<code><p> Перший параграф.</code>	Друкуватиме перший абзац з текстом "Перший параграф"
<code><p align="left">Другий параграф</code>	Друкуватиме другий абзац з текстом "Другий параграф" по лівому краю

Елемент FONT

Елемент FONT робить зміну шрифту тексту, та визначає вид, розмір і колір шрифту для тексту.

Початковий тег: необхідний

Кінцевий тег: необхідний

Визначення атрибутів:

size = number – встановлює розмір шрифту.

<code><fontsize='1' color='black' face='Tahoma'>Рядок</code>	Виводить слово «Рядок» чорним кольором, шрифт «Tahoma», розмір 1.
---	---

Можливі значення:

- Ціле число від 1 до 7. Встановлює шрифт певного фіксованого розміру, представлення якого залежить від браузера користувача.

- Відносне збільшення розміру шрифту. "+1" означає: на один розмір більше. "-3" означає: на три розміри менше. Всі розміри знаходяться в межах шкали від 1 до 7

color = color – встановлює колір тексту

face = string – задає ім'я шрифту

<code>bold</code>	Напівжирний
<code><i>italic</i></code>	<i>Курсив</i>
<code><i>bolditalic</i></code>	<i>напівжирний курсив</i>
<code><tt>teletypetext</tt></code>	Телетайп
<code><big>big</big></code>	"великий" шрифт
<code><small>small</small></code>	"малий" шрифт

Підіндекс і надіндекс: елементи SUB і SUP, що переводять текст в нижній і верхній регістр

Початковий тег: необхідний

Кінцевий тег: необхідний

H <code><sub>2</sub></code> O	H ₂ O
E = mc <code><sup>2</sup></code>	E=mc ²

3.2. Питання для самоконтролю

Що таке типографіка?

Для чого були створені такі теги та атрибути?

Які атрибути типографіки ви можете вказати?

Які значення можуть приймати ці атрибути?

3.3. Практичне завдання

1. Запустити програму Блокнот і відкрити документ index.html.
2. Змінити документ так, щоб напис «Харківський національний університет імені В.Н.Каразіна» був заголовком першого рівня і був вирівняний по центру.
3. Додати назву факультету та спеціальності – заголовком другого та третього рівня, також вирівняти по центру.
4. Додати в документ параграф: Прізвище, Ім'я, По батькові.
5. Додати в документ два параграфи про себе.
6. Перший параграф вирівняти по правому краю та встановити для слова розмір 5.
7. Для третього параграфа: розмір 4.
8. Для свого прізвища, імен, по батькові встановити шрифт Arial Black.
9. Підкреслити якийсь рядок у тексті.
10. Великим шрифтом виділити факультет, курсивом – спеціальність.

4. Кольорове оздоблення

4.1. Кольорове оздоблення та системи представлення кольорів

Існують кілька основних способів представлення кольорів в інтернеті.

У вигляді #RRGGBB.

Колір представлений 16-річним числом (яке випереджається знаком # в наступному форматі #RRGGBB, де RR – градація червоного кольору, GG – зеленого, BB – синього).

Black (Чорний) = "#000000"	Green (Зелений) = "#008000"
Silver (Срібло) = "#C0C0C0"	Lime (Лайм) = "#00FF00"
Gray (Сірий) = "#808080"	Olive (Оливковий) = "#808000"
White (Білий) = "#FFFFFF"	Yellow (Жовтий) = "#FFFF00"
Maroon (Темно-бордовий) = "#800000"	Navy (Темно-синій) = "#000080"
Red (Червоний) = "#FF0000"	Blue (Синій) = "#0000FF"
Purple (Фіолетовий) = "#800080"	Teal (Чирок) = "#008080"
Fuchsia (Фуксія) = "#FF00FF"	Aqua (Аква) = "#00FFFF"

За назвою.

Подання ключовими словами, наприклад green, black. Щоб уникнути випадків, коли вказане ключове слово «не розпізнається» браузером, слід використовувати лише невеликий набір основних кольорів, які використовуються у всіх браузерах.

RGB (*, *, *) або RGBA (*, *, *, *)

У вигляді RGB (*, *, *), де «*» – числа від 0 до 255, що позначають кількість відповідного кольору (червоний, зелений, синій) в вихідному.

Можливий і RGBA (*, *, *, *), де перші 3 «*» – компоненти кольору в діапазоні 0 до 255, а остання «*» – рівень непрозорості (альфа-канал), задати дробовими числами від 0 до 1.

Інші графічні елементи

Горизонтальні лінії – можна використати елемент HR

Початковий тег: необхідний

Кінцевий тег: заборонений

Визначення атрибутів:

- ✓ `align = left | center | right` – визначає горизонтальне вирівнювання лінії.
- ✓ `size = number` – визначає висоту лінії.
- ✓ `width = number` – визначає ширину лінії. Значення може бути числовим і у відсотках від ширини вікна браузера.
- ✓ `color = color` – встановлює колір лінії.

<code><hr width="50%" align="center"></code>	Зроблено вирівнювання лінії за центром, ширина 50% від ширини вікна браузера.
<code><hr size="5" width="100" align="left" color="#ff0000"></code>	Червона лінія, розташована зліва, висотою 5 точок і шириною 100 точок.

Рядок, що біжить

Такий рядок можна задати елементом MARQUEE

Початковий тег: необхідний

Кінцевий тег: необхідний

Визначення атрибутів:

- ✓ `bgcolor = color` – колір фону рядка, що біжить
- ✓ `loop = number` – число повторів анімації рядка, що біжить (значення може бути цілим або `infinite` – текст буде продовжувати бігати, поки читач не перейде на нову сторінку)
- ✓ `direction = left | right | up | down` – визначає напрямок: `left` (за замовчуванням) – справа наліво, `right` – зліва направо, `up` – від низу до верху, `down` – зверху вниз.
- ✓ `scrollamount = number` – швидкість руху рядка. (Рекомендується ставити швидкість "1", в цьому випадку рядок виглядає більш зручним для читання і не смикається).

- ✓ `scrolldelay = number` – задає часовий інтервал між кроками рядка, що біжить.
- ✓ `width = number` – ширина рядка, що біжить, в пікселях.
- ✓ `height = number` – висота рядка, що біжить. (Якщо ви робите напис, що біжить, в один рядок, то можна висоту не вказувати, вона сама підбирається під розмір букв).

<code><marquee>Рядок, що біжить</marquee></code>	Напис «Рядок, що біжить» біжить вліво
<code><marquee direction="right" scrollamount=1 scrolldelay=5>Рядок, що біжить</marquee></code>	Напис «Рядок, що біжить» біжить вправо зі швидкістю 1 та з часовим інтервалом 5 між кроками

4.1. Питання для самоконтролю

Які системи представлення кольорів ви знаєте?

Яка різниця між різними системами?

Як отримати певний колір в кожній з систем?

Які ще елементи графічного оздоблення можна вжити?

4.2. Практичне завдання

1. Встановити різні кольори для заголовків та параграфів
2. Використовувати різні способи кодування кольору
3. Намалювати лінію на всю ширину екрану
4. Зробити рядок «прізвище, ім'я, по батькові» таким, що біжить, встановивши колір фону сірий, напрямок зліва направо.

5. Представлення інформації списками

5.1.Списки

HTML дозволяє визначати зовнішній вигляд цілих абзаців тексту, при цьому це можуть бути не просто звичайні абзаци, а й більш складні структури у вигляді простих і нумерованих списків.

Абзаци можна виділяти, якщо реорганізувати їх у списки, виводити їх на екран у відформатованому вигляді, зрушувати, збільшуючи ліве поле.

Ненумеровані списки

Теги ` ... </ UL>` описують ненумерований список.

Початковий тег: необхідний

Кінцевий тег: необхідний

Визначення атрибутів:

- ✓ `type = disc | circle | square` – задає стиль міток для даного списку: `circle` (кружечок), `disc` (зафарбований кружечок, за замовчуванням) або `square` (квадрат).

Нумеровані списки

Теги ` ... </ OL>` описують нумерований список.

Початковий тег: необхідний

Кінцевий тег: необхідний

Визначення атрибутів:

- ✓ `type = 1 | a | A | i | I` – задає стиль міток для даного списку: `1` (арабські цифри, за замовчуванням), `a` (латинський алфавіт, нижній регістр) або `A` (латинський алфавіт, верхній регістр), `i` (римські цифри, нижній регістр), `I` (римські цифри, верхній регістр).

- ✓ `start = number` – встановлює номер першого елемента впорядкованого списку. За замовчуванням це "1". Зауважте, що, хоча значенням цього атрибуту є ціле число, відповідні мітки можуть бути нечисловими.
- ✓ Так, якщо стиль елемента списку – це латинські літери у верхньому регістрі (A, B, C, ...), `start = 3` означає "C". Якщо стиль – це римські цифри в нижньому регістрі, `start = 3` означає "iii" і т.д.

Кожен елемент обох списків повинен бути визначений тегом ``.

Елемент нумерованого і ненумерованого списків: ``.

Опис: описують елемент списку.

Початковий тег: необхідний

Кінцевий тег: не обов'язково

Визначення атрибутів:

- ✓ `value = число` – встановлює номер поточного елемента списку. Зауважте, що, хоча значенням цього атрибуту є ціле число, відповідні мітки можуть бути нечисловими (див. Атрибут `start`).

<pre><UL type="disc"> перший.. другий... </pre>	<ul style="list-style-type: none"> • перший • другий
<pre><OL type="1"> <LI value=5> перший... другий. </pre>	<p>5. перший</p> <p>6. другий</p>
<pre><OL type="i" start="5"> перший <LI value="10">другий</pre>	<p>v. перший</p> <p>vi. другий</p>

	
-------	--

5.1. Питання для самоконтролю

Як визначити нумерований список?

Як визначити нумерований список?

Як задати різні маркери нумерованого списку?

Як задати різні варіанти нумерації нумерованого списку?

5.2. Практичне завдання

1. Запустити програму Блокнот.

2. У новому документі aboutme.html оформити інформацію наступним чином:

- Прізвище, ім'я
- Факультет група
- Про себе
- Знання ІТ технологій (мови, бази даних, комп'ютерні програми)
- Захоплення
- Фон

3. Для роботи з зображенням знайдіть на диску потрібну Вам картинку і скопіюйте її в свою папку. Для своєї сторінки встановіть фон. Додайте своє фото.

4. Розділ «Про себе» повинен бути написаний 12 шрифтом, колір встановити на свій розсуд, шрифт – напівжирний курсив.

5. Назви факультету і групи оформити у вигляді рядка, що біжить.

6. Знання ІТ технологій позначити цифрами (нумерованим списком), вони повинні бути написані 10 розміром шрифту, колір вибрати на свій розсуд.

7. Хобі оформити у вигляді нумерованого списку, 10 розміром шрифту, колір вибрати на свій розсуд.
8. Під прізвищем та ім'ям провести лінію червоного кольору, ширина якої дорівнює 5px, на всю сторінку.
9. Внизу сторінки провести лінію такої ж товщини, під якою написати справа адресу своєї електронної пошти.

6. Посилання

6.1. Зовнішні та внутрішні посилання

В HTML існує три типи посилань:

- ✓ всередині сторінок – вони задають переходи в межах однієї сторінки;
- ✓ внутрішньосистемні – посилання між сторінками в межах одного і того ж сервера;
- ✓ міжсистемні – посилання на сторінки, розташовані на віддалених вузлах Web.

Для визначення посилань призначений спеціальний тег `a`, який називається Anchor (якір).

Опис: визначає посилання або якір.

Початковий тег: необхідний

Кінцевий тег: необхідний

Визначення атрибутів:

- ✓ `name = "рядок"` – називає поточний якір, який може стати якорем призначення для іншого гіперпосилання. Значенням цього атрибута

повинно бути унікальне ім'я якоря. Областю видимості цього імені є поточний документ.

- ✓ href = "рядок" – визначає розміщення ресурсу у Web, визначаючи таким чином посилання з поточного елемента на якір призначення, визначений цим атрибутом.

<code>Інформація</code>	Створюється мітка з назвою info.
<code>INFO</code>	Створюється посилання на сайт INFO.

Колір посилання можна задати за допомогою атрибутів тегу BODY:

- ✓ link = "color" – встановлює колір невідвіданих гіперпосилань.
- ✓ vlink = "color" – встановлює колір відвіданих гіперпосилань.
- ✓ alink = "color" – встановлює колір гіперпосилань при виборі користувачем.

<code><body link="#ff0000" vlink="#00ff00"></code>	Колір невідвіданих гіперпосилань - червоний, відвіданих - синій.
--	--

Для вказівки посилання на електронну скриньку значення атрибуту href повинне бути вигляду "mailto: ім'я_електронної_скриньки".

<code><a href="mailto: <u>lena@info.com</u>">Написати листа</code>	Створиться посилання «Написати листа», вибравши яке, можна написати і відправити лист.
--	--

Внутрішньосторінкові посилання:

Створити ім'я (мітку) для точки призначення, на яку повинен здійснюватися перехід.

Мітка створюється за допомогою тегу якоря, використовує його атрибут NAME.

```
<a name="info">Додаткова інформація</a>
```

Фраза "Додаткова інформація" при цьому ніяк не буде виділена в тексті документа.

Для створення гіперпосилання на цю точку документа використовується тег <a> з атрибутом HREF =, при цьому до імені якоря приєднується знак #:

```
<a href="#info">Перегляд додаткової інформації</a>.
```

Такий фрагмент HTML-тексту призведе до появи в документі виділеного фрагмента (в нашому випадку фрази "Перегляд додаткової інформації"), при натисканні на який відбудеться перехід до рядка #info з додатковою інформацією.

Внутрішньосистемні посилання:

У файлі, в який ми хочемо перейти, необхідно створити мітку аналогічно попередньому прикладу.

Створити гіперпосилання можна аналогічно, тільки ім'я якоря приєднується до імені файлу через позначку #. В тексті параграфу це буде виглядати так:

```
Коротко про мене можна дізнатися <a href="aboutme.html#info"> тут  
</a>
```

При натисканні на виділений фрагмент відбудеться перехід до рядка #info в файлі з ім'ям aboutme.html.

Міжсистемні посилання.

Використовуючи ці посилання, можна встановити зв'язок з будь-якою сторінкою на будь-якому вузлі Web.

В цьому випадку необхідно створити тільки гіперпосилання: наприклад, посилання на сервер маркетплейсу rozetka виглядає наступним чином:

```
<a href="http://www.rozetka.com"> Перехід на сайт Rozetka</a>
```

При натисканні на виділений фрагмент відбудеться перехід на сайт фірми Rozetka.

6.2. Питання для самоконтролю

Як працюють посилання всередині сторінок?

Як працюють внутрішньосистемні посилання?

Як працюють зовнішні посилання?

Що таке якір?

Які атрибути посилань ви знаєте?

6.3. Практичне завдання

1. У файл `index.html` додати абзац «Інформацію про мене, мої знання і захоплення можна подивитися тут»
2. На початку файлу `aboutme.html` фразу «Про себе» укласти в теги `` і `<a>`.
3. У документі `main.html` слово «тут» оформити гіперпосиланням на документ `aboutme.html` на якір «info»
4. Змінити колір невідвіданих гіперпосилань на синій, а відвіданих на сірий.
5. Випробувати дію гіперпосилання.
6. Оформити електронну пошту у вигляді посилання для відправки листа.
7. У файлі `aboutme.html` внизу сторінки помістити абзац, що складається з одного слова «Назад», вирівняного по центру.
8. Організувати зворотний перехід.
9. Завдання: в документ `main.html` додати абзаци
 - «Тут ви можете подивитися сайт ХНУ ім. В.Н.Каразіна».
 - «Тут ви можете подивитися сайт інтернет-магазину Rozetka».
10. Оформити абзаци на свій розсуд.
11. Оформити абзаци як гіперпосилання на сайти ХНУ і Rozetka

7. Використання мультимедіа, частина 1

7.1. Додавання зображень

Додавання зображення в HTML-сторінку здійснюється за допомогою тегу .

Тег .

Опис: впроваджує зображення в поточний документ в місці визначення елемента.

Початковий тег: необхідний.

Кінцевий тег: заборонений.

Визначення атрибутів:

- ✓ src = "рядок" – задає шлях в структурі каталогів до файлу із зображенням (найбільш поширені формати зображень: GIF, JPEG і PNG);
- ✓ alt = "рядок" – визначає альтернативний текст (який з'являється при наведенні курсору миші на зображення);
- ✓ width = "число" – визначає ширину об'єкта (зображення);
- ✓ height = "число" – визначає висоту об'єкта.

Для зменшення часу завантаження сторінки з графікою корисно вказувати розмір зображення.

Якщо він відомий ще до завантаження сторінки, то браузер може відвести рамку для картинки, а далі завантажувати текст на сторінку.

Поки завантажувється графіка, відвідувач сторінки може почати читати текст:

- ✓ border = "число" – визначає ширину рамки навколо об'єкта;
- ✓ align = "bottom | middle | top | left | right" – визначає позицію об'єкта по відношенню до навколишнього тексту (bottom – низ об'єкта повинен

бути вирівняний вертикально по поточній базовій лінії (за замовчуванням), middle – центр об'єкта повинен бути вирівняний вертикально за поточною базовою лінією, top – верх об'єкта повинен бути вирівняний вертикально по верхній межі поточного рядка, left – притискає об'єкт до лівого краю, right – притискає об'єкт до правого краю).

<pre></pre>	<p>Вставити картинку pic1.jpg розміром 100x100 і обтіканням текстом зліва</p>
<pre></pre>	<p>Додати картинку в рамці шириною 2px .</p>

7.2. Питання для самоконтролю

Який тег відповідає за додавання зображень в гіпертекстовий документ?

Які в нього атрибути?

Як додати зображення необхідного розміру?

Як задати відносне позиціонування зображення?

Як додати зображення з рамкою?

7.3. Практичне завдання

Додати на ваш імпровізований сайт 3 зображення:

- Ваше власне фото
- Фотографію будівлі ХНУ ім. В.Н.Каразіна
- Іконку для позначення переходу назад.

Малюнок іконки виконати самостійно в будь-якому графічному редакторі

8. Використання мультимедіа, частина 2

8.1. Анімація та звук

Додавання звуку і відеоматеріалів.

Ви можете додати на свою сторінку звуки або відеоматеріали. Вони будуть запускатися автоматично при завантаженні сторінки. Для цього використовується тег `<EMBED>`, який призначений для вбудовування об'єктів в документ.

Тег `<embed>`

Опис: впроваджує об'єкт поточний документ в місці визначення елемента.

Початковий тег: необхідний.

Кінцевий тег: необхідний.

Визначення атрибутів:

- ✓ `src` = "рядок" – задає шлях в структурі каталогів до файлу із зображенням (найбільш поширені формати зображень: GIF, JPEG і PNG);
- ✓ `autostart` = "true | false" – визначає автозапуск аудіо- або відеокліпа;
- ✓ `repeat` = "true | false" – визначає, чи починати заново після програвання запису;
- ✓ `width` = "число" – визначає ширину об'єкта (зображення);
- ✓ `height` = "число" – визначає висоту об'єкта.
- ✓ `border` = "число" – визначає ширину рамки навколо об'єкта;
- ✓ `align` = "center | left | right" – розташування пульта управління.

```
<embed src="media.wav" width=145
```

Програвати WAV файл зі стандартним

<pre>height=55 autostart=true repeat=true></embed></pre>	<p>пультом управління, автостартом та з дозволом повтору</p>
---	--

Створення анімованих gif-файлів на прикладі CorelPhoto-Paint.

Для початку розберемо створення простої анімації з тексту. Для цього запустить графічний редактор CorelPhoto-Paint.

Виберіть пункт меню Файл | Новий. Встановіть розміри зображення 400x50 пікселів, режим - 8-ми бітна палітра, колір фону встановіть на свій розсуд, поставте прапорець «Створити фільм», кількість кадрів залиште рівним 1.

Вибрати інструмент текст і в лівому нижньому кутку написати якусь букву. Встановити колір і розмір букви.

Виділити букву і виконати команду меню

Об'єкт | Комбінувати → Склеїти об'єкти з фоном.

Виконати команду меню Фільм | Вставити кадр. Відзначити опцію «Копіювати поточний кадр» і натиснути ОК. Таким чином з'явиться другий кадр, точно такий же, як перший.

На новий кадр можна додати наступну букву іншого кольору (можна її розташувати по дузі або лінії, щодо інших букв). Після створення останнього кадра можна зберегти документ як GIF.

8.2. Питання для самоконтролю

Як додати в документ аудіо або відеоматеріали?

Як створити gif зображення?

Як додати gif зображення?

Як задати розміри таких вставок?

Які ще корисні атрибути ви знаєте?

8.3.Практичне завдання

1. На жорсткому диску знайти файли з розширенням mp4 і скопіювати один з них в свою папку.
2. Помістити на сторінку раніше обране відео.
3. Створити ролик, в якому буква за буквою з'являється напис «Група ...» (назва Вашої групи), для цього виконати аналогічні дії для кожної букви.
4. Після створення останнього кадру зберегти документ як GIF і додати зображення на свою сторінку вгорі.

Список літератури

1. Інтернет технології. Методичні вказівки до лабораторних робіт для студентів напрямку підготовки 6.050802 «Електронні пристрої та системи». – К.: НТУУ “КПІ”, 2014. – 90 с.
2. Методичні вказівки і завдання до лабораторних робіт з курсу інформатики. Частина 2. Основи web-розробки: Навч.-метод. посіб. / Автори-укладачі: О.В. Присяжнюк, О.В. Резіна – Кропивницький: ЦДПУ імені В. Винниченка, 2021. – 42 с.
3. Пасічник О.Г., Пасічник О.В., Стеценко І.В. Основи веб-дизайну дизайну : навч. посіб. – К.: Вид. група ВНУ, 2009. – 336 с.

Інтернет-ресурси

<https://html.spec.whatwg.org/multipage/> – поточний стандарт html

<https://www.w3schools.com/html/> – офіційний посібник html

Електронне навчальне видання комбінованого використання
Можна використовувати в локальному та мережному режимі

Сузікова Олена Геннадіївна

ВЕБ ТЕХНОЛОГІЇ В ІНФОРМАЦІЙНОМУ ОБМІНІ

Методичні рекомендації
до практичних занять для здобувачів вищої освіти першого (бакалаврського)
рівня з дисципліни «Обробка, зберігання та передача даних в сучасних
інформаційних технологіях» за спеціальністю «Прикладна математика»

У 3 частинах

Частина 1: HTML

В авторській редакції

Підписано до розміщення 25.06.2025. Гарнітура Times New Roman.
Ум. друк. арк. 1,95. Обсяг 1,486.Зам. № 246/25.

Харківський національний університет імені В. Н. Каразіна,
61022, м. Харків, майдан Свободи, 4.
Свідоцтво суб'єкта видавничої справи ДК № 3367 від 13.01.2009.
Видавництво ХНУ імені В. Н. Каразіна