

Урок 1: Веб

Передмова

Чи замислювалися ви над тим, як працює всесвітнє павутиння? Звичайно ж так! Сьогодні ми вчитимемось створювати веб-сайти, щоб ви змогли зробити власний внесок в розбудову всесвітньої мережі. Сайти створюють за допомогою мови гіпертекстової розмітки HTML (HyperText Markup Language). Ви дізнаєтеся про неї більше у процесі самостійного створення веб-сторінки.

Крок 1: Що таке веб-сторінка?

1. Відкрийте текстовий редактор.
2. Створіть новий документ.
3. Напишіть щось! Наприклад, "Привіт! Мене звати.."
4. Збережіть файл. Назвіть його `привіт.txt`.
5. Тепер знайдіть цей файл на комп'ютері і відкрийте його знову. Він відкриється у текстовому редакторі, але це не дуже цікаво.
6. Змініть розширення файлу (символи після крапки у назві файлу) на `.html`, тепер новий файл називається `привіт.html`.
7. Знову відкрийте файл.

За допомогою якої програми відкривався файл цього разу? Веб-браузер – це спеціальна програма, яка знає, як читати текстові файли, написані з використанням мови HTML. Ми поки що не використовували HTML, а написали звичайний текст, але браузеру все одно! Відкриваючи файл із розширенням `.html`, браузер робитиме все можливе, щоб відобразити його якнайкраще.

Це дуже зручно: навіть коли веб-сайт містить помилки, браузер намагатиметься виправити їх, щоб показати вміст сайту.

Як можна проглянути ці файли?

Коли ви вводите адресу в адресний рядок браузера, ви відправляєте запит на підключення до комп'ютера, який завжди увімкнений та налаштований для перегляду усіх сторінок, розміщених на ньому. Такий комп'ютер називають сервером. Коли він отримує запит від вашого комп'ютера, він починає шукати потрібні файли з розширенням `.html` і відправляє їх вам разом з рештою контенту, який передбачений для цієї сторінки, наприклад, картинками чи відео.

// схема, за якою працює веб, починається із надсилання запиту про отримання доступу до певної сторінки, а закінчується відображенням необхідного контенту.

Крок 2: Що таке HTML?

HTML – це мова розмітки, яка використовується для опису того, що міститься на веб-сторінці.

Коли браузер намагається якнайкраще відобразити вміст сторінки, мова розмітки допомагає йому зрозуміти, що на сторінці розміщено.

Щоб показати це браузеру, використовують теги.

Теги виглядають так: `<p>це якийсь текст.</p>`

`<p>` це скорочення від **параграф** (від paragraph).

Є відкриті теги: `<p>` та закриті теги з позначкою коса риска (слеш): `</p>` Браузер знає, що між цими дома тегами розміщено параграф тексту.

Теги можуть мати атрибути, що містять корисну інформацію про певну складову веб-сторінки. Розглянемо тег посилання: `Visit CodeClub`

`website` `<a>` - це скорочення від анкер (якір), як зазвичай позначали посилання. Цей тег

теж буває відкритим: `<a>` і закритим: `` але до відкритого тега додається атрибут `:<a`

`href="http://codeclubua.org">` `href` – це назва атрибута, а `http://codeclubua.org` – його

значення. `href` – це скорочення від *hypertext reference* (*гіпертекстове посилання*).

Тексти, який містить посилання на інші тексти, називають гіпертекстом, бо він може містити зображення, звуки та безпосередньо відсилати до інших текстів. Це дещо відрізняє його від звичайного тексту. Атрибут `href` інформує браузер, куди потрібно перейти за посиланням, а текст між тегами відображається як посилання.

Перевірка активності { .check }

1. Відкрийте файл `сторінка.html`.
2. Поцікавтеся у координатора, чи можете ви використати X-Ray Goggles або інструменти розробника, щоб проглянути код сторінки (розробник – це той, хто створює коди).

Якщо ви можете використовувати X-Ray Goggles:

1. Клікніть на вкладку X-Ray Goggles.
2. Переміщайте вказівник мишки по сторінці. Частини сторінки будуть підсвічуватись і ви зможете побачити, які теги використано для їх створення. Ви зможете клікнути на кожен блок сторінки, щоб побачити фрагмент коду, який йому відповідає.

Якщо ви використовуєте інструменти розробника:

1. Переміщайте вказівник мишки по сторінці. Клікайте правою кнопкою мишки на будь-який елемент, що вас зацікавить і обирайте в контекстному меню `Перевірити елемент`. Внизу сторінки відкриється вікно, де відобразатиметься код цієї ж сторінки.
2. Наводьте вказівник мишки на різні частини коду. Відповідні елементи сторінки будуть підсвічуватись, щоб ви бачили, що відповідає кожному рядку коду на сторінці.
3. Спробуйте перевірити усі частини сторінки. Ви зрозуміли, за що відповідають різні теги?

Ми уже вивчили теги `<p>` та `<a>`. `` - упорядкований список (англ. ordered list) `` - неупорядкований список (з англ. unordered list) `` - елемент списку (з англ. list item) `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>` - заголовки `<hr>` - горизонтальна риска (з англ. horizontal rule) `<div>` - тег для групування елементів `` - спеціальний тег, який немає закритого та відкритого тегів і використовується для додавання на сторінку зображень.

Також є теги, які завжди використовуються в HTML документах, ось вони: `<html>` - інформує браузер, де починається код, `<head>` - всередині тега `<head>` розміщується інформація, яка може бути корисною для браузера, але не відображається як текст на сторінці. У даному прикладі ми розмістили там `<title>`, який потім видно у верхній частині вікна браузера. І `<body>` - тег, всередині якого містяться елементи, які буде видно на сторінці.

Перевірка активності { .check }

- Зверніть увагу, як теги вкладаються один в другий. Наприклад, тег `<a>` міститься всередині тегу `<p>`, який в свою чергу розташований всередині тегу `<div>`, розміщеного всередині тегу `<body>`.

Вкладений тег називають дочірнім, а тег, у який вкладено інший – батьківським. Нагадує генеалогічне дерево, правда ж?

- Для браузера всі теги одного виду однакові, але ви можете позначити їх за допомогою класів та ідентифікаторів (id). Наприклад, деякі з параграфів тексту можуть бути частинами вступу, тобто їм можна присвоїти клас `вступ`. Спробуйте помістити кілька класів у код `сторінка.html`.
- Ідентифікатори використовуються для позначення елементів, які лише один раз трапляються на сторінці. Спробуйте помістити в тег `div id` кошеня на сторінці.
- Що станеться, якщо перемістити елементи коду? Поверніться в редагування коду. Знайдіть тег `` та виділіть його і все, що всередині нього, як показано нижче:

```
<ol>
```

```
<li>Kittens</li>
```

```
<li>Cake</li>

<li>Lie-ins</li>

<li>Playing games</li>

</ol>
```

Скопіюйте його і перемістіть ще кудись. Збережіть сторінку і оновіть її в браузері. Як впливає порядок написання команд в коді на порядок відображення елементів сторінки браузером?

Спробуйте { .try }

- Створити параграф тексту.
- Створити посилання на іншу частину сторінки. (підказка: використайте атрибут id – прогляньте посилання, за яким відбувався перехід до кошеняти).
- Додайте свої заголовки туди, де вони, на ваш погляд, потрібні. Що відбувається при зміні номерів біля тегів заголовків, наприклад <h3> на <h4>?
- Що потрібно зробити, щоб перейти на іншу сторінку за посиланням? Якщо ви використовуєте інструменти розробника, спробуйте двічі клікнути на елемент коду, який видається цікавим, коли відкриється панель з кодом сторінки. Спробуйте змінити його. Ви отримали можливість попереднього перегляду без необхідності переключатись між браузером і редактором коду. Зручно, правда ж? Оновіть сторінку. Що сталося? Коли ви таким чином редагуєте код, зміни не зберігаються, тобто ви можете попередньо проглянути сторінку, експериментувати зі змінами, маючи змогу повернутись до попереднього вигляду сторінки.