

СТРУКТУРА WEB-СТОРИНКИ

План:

1. Сутність гіпертексту
2. Розвиток стандартів HTML
3. Складові елементи HTML-документа
4. Структура HTML-документа
5. Типи тегів

Сутність гіпертексту

HTML (HyperText Markup Language) – це система верстки, яка використовується для визначення того, як повинні бути розміщені елементи на веб-сторінці. Веб-розробник визначає, яким чином і які елементи мають бути представлені на веб-сторінці відповідно до своїх цілей.

Незалежно від того, який контент міститься на веб-сторінці, який стиль використовується, а також які банери та інші елементи зовнішнього вигляду додані до веб-сторінки, все це може бути описано звичайним текстом.

У 1965 році американець Тед Нельсон запропонував термін "гіпертекст". Він розумів під ним *"розгалужений текст, що виконує дії за запитом"*. Іншими словами, *це текст, що містить посилання на інші фрагменти, які структуровані як система можливих переходів*.

Якщо дотримуватися посилань і переходів, то можна отримати лінійну, традиційну форму представлення текстів.

Технічно гіпертекстова система зберігає інформацію у текстовому форматі, що дозволяє встановлювати зв'язки між інформаційними одиницями, що зберігаються в її пам'яті.

Енциклопедія є класичним прикладом гіпертексту, де статті містять відсилання на матеріали цієї ж енциклопедії.

Завдяки структуризації гіпертекст підвищує ефективність процедур пошуку певної інформації.

У статті "As We May Think", опублікованій Ванневаром Бушем в 1945 році, було представлено пристрій машини Memex, що зберігає на мікрофільмах інформацію і для пошуку необхідної інформації пропонувалось використовувати асоціативні зв'язки, що є прообразом гіпертекстових систем. Після появи протоколу передачі даних HTTP, системи адресації URL і мови HTML, поняття гіпертексту стало нерозривно пов'язане з інформаційними технологіями загалом і глобальною мережею – Інтернетом зокрема.

Розвиток стандартів HTML

HTML – мова гіпертекстової розмітки, що була заснована на мові SGML (стандартній узагальненій мові розмітки).

Головною ідеєю SGML було створення мови розмітки документів, що не прив'язана до певної апаратно-програмної платформи.

Основним принципом SGML було впровадження в текст керуючих конструкцій, які поділяли елементи структури, але не містили інформацію про те, як ці структури повинні відображатися.

У окремий файл виносилася структура, що відповідає за оформлення (лист стилів). Таким чином, одна і та ж, з точки зору змісту, інформація могла бути представлена в різних видах. Для гарантії обробки документа він мав мати окремий DTD-файл, що визначав його тип.

GML (Generalized Markup Language) – мова, що задовольняє описаним вище правилам, була створена в 1969 році. Після прийняття міжнародним стандартом ISO в 1989 році, до її назви додалася початкова літера S. Хоча SGML була відносно малопопулярною через відсутність засобів для відображення інформації відповідно до розмітки, Європейський інститут фізики елементарних частинок (CERN) у 1991 році розпочав розробку першого текстового браузер, що підтримує гіпертекстову розмітку, здатного відображати графіку і переходити по посиланнях.

Мова HTML була розроблена під час цієї роботи, і вона в багатьох відношеннях схожа на SGML, за винятком структур, які дозволяють відображати текст особливим чином (відступи, виділення і т.д.).

У 1993 році був створений перший браузер з графічним інтерфейсом, а в 1994 році був заснований консорціум W3C.

З того часу було прийнято кілька стандартів HTML та специфікацій CSS. Наприклад, 22 грудня 1995 року було прийнято стандарт HTML 2, а в 1996 році консорціум W3C прийняв специфікацію CSS1.

Інші стандарти включають HTML 3.2 (прийнятий 14 січня 1996 року), HTML 4.0 (прийнятий 18 грудня 1997 року), HTML 4.01 (прийнятий 24 грудня 1999 року) та XHTML 1.0 (прийнятий 26 січня 2000 року). До того ж, специфікація CSS2 була прийнята в 1999 році, а стандарт XHTML 1.1 – 16 лютого 2007 року.

У 2007 році специфікація HTML5 була прийнята як початок роботи над новим стандартом HTML робочою групою W3C.

28 жовтня 2014 року W3C офіційно рекомендувала використовувати HTML5 як фіналізований та готовий до широкого використання стандарт. Метою розробки HTML5 є поліпшення рівня підтримки мультимедіа-технологій, збереження зворотної сумісності, читабельності коду та простоти аналізу для парсерів.

Особливості HTML5 включають взаємодію з JavaScript за допомогою об'єктної моделі документа та підтримку всіх елементів HTML 4. HTML5 можна розглядати як нову платформу для створення веб-додатків, яка розширює можливості своїх попередників.

Складові елементи HTML-документа

Елементи

HTML-документ складається з різних елементів, таких як заголовки, списки, абзаци тексту, таблиці та інші.

Кожен елемент має своє ім'я, яке зазвичай є англійським словом або скороченням. Є три компоненти розмітки елемента: початковий тег, кінцевий тег і текст або інші елементи між ними.

1. Початковий тег складається з імені елемента, закритого в кутові дужки (< >).
2. Кінцевий тег має те ж ім'я, що й початковий, але з косою рисою (/) перед ним.
3. Елементи повинні бути вкладені один в одного, і кожен вкладений елемент повинен починатися і закінчуватися в межах батьківського елемента. Деякі елементи, такі як зображення, не мають кінцевого тега, оскільки вони не містять вмісту.

У HTML є дві категорії елементів: блокові та строкові.

Блокові елементи використовуються для відображення структури документа. За замовчуванням блоки починаються з нового рядка та мають певну ширину. Блоки можуть бути вкладені один в одного, а також містити строкові елементи. До блокових елементів належать параграфи, заголовки, таблиці та списки.

Рядкові елементи, навпаки, містяться всередині блокових елементів та охоплюють лише частину тексту. Вони зазвичай не призводять до появи нового рядка. До рядкових елементів належать гіперпосилання, зображення, виділені слова чи фрази та короткі цитати.

Атрибути

HTML елементи можуть мати додаткову інформацію, яку називають атрибутами.

Атрибути розміщуються після імені елемента в початковому тезі і відокремлюються один від одного пробілами. Порядок проходження атрибутів в тезі не має значення. Значення атрибута записується після знаку рівності і повинно бути поміщено в поодинокі або подвійні лапки. Хоча в деяких випадках лапки можна опустити, рекомендується завжди використовувати лапки для значень атрибутів. Імена атрибутів можуть бути написані в будь-якому регістрі, але значення атрибутів можуть залежати від регістру.

`<div class="head"> в м і с т </div>`

Елементи мають атрибути, які надають додаткову інформацію про них.

Атрибути розміщуються в початковому тезі після імені елемента і відокремлюються один від одного пробілами. Порядок проходження атрибутів в тезі не має значення. Значення атрибута слідує за знаком рівності, що стоїть після імені атрибута. Значення може бути поміщено в поодинокі або подвійні лапки. Рекомендується завжди укладати значення атрибутів в лапки.

Імена атрибутів можуть бути написані в будь-якому регістрі, проте їх значення можуть залежати від регістра.

Список всіх атрибутів та їх можливі значення визначаються в Специфікаціях HTML

<http://www.w3.org/TR/html401/index/attributes.html>. Користувач не може створювати свої власні атрибути або використовувати значення, що не визначені Специфікацією, так як це може призвести до проблем правильної інтерпретації веб-сторінки.

Найбільш поширеними атрибутами є базові атрибути, такі як class, id, style і title, які визначають загальні властивості елементів, а також атрибути для локалізації, такі як dir та lang, які вказують на мовні властивості вмісту елемента.

Атрибут id, що називається ідентифікатором елемента, присвоює елементу унікальне ім'я в межах даного документа. Це означає, що ніякі два елемента не можуть мати однакових значень id. Зазвичай ідентифікатор використовують як селектор стилів для окремих елементів, для закладок гіперпосилань та для вказівки на конкретний елемент у сценарії.

Наприклад, якщо ми маємо елемент з id="footer", то ми можемо стилізувати його за допомогою CSS, використовуючи селектор #footer. Крім того, з допомогою JavaScript, можна виконувати дії з елементом, вказавши його id. Слід пам'ятати, що значення атрибуту id повинно бути унікальним всередині документа.

У наступному прикладі атрибут id використовується для ідентифікації параграфа:

<p id="footer"> Знання – сила! </p>

Атрибут `class` вказує на належність елемента до певного класу. До одного класу може відноситися будь-яка кількість елементів. Елемент може належати до декількох класів, вказавши їх імена через пробіл. Браузери не завжди підтримують списки класів. Атрибут `class` використовується як селектор стилів. Наприклад, в параграфі з ім'ям `header` ми можемо використати клас елементів з ім'ям `header`, вказавши його в атрибуті `class`:

<P class="header"> Заголовний текст </P>

Атрибут `style` дозволяє задати стиль елемента. Але перед використанням цього атрибута в заголовок документа повинен бути включений метаописувач.

Атрибути `class` і `id` краще використовувати, оскільки вони забезпечують поділ вмісту документа і стилю його відображення, що більшою мірою узгоджується з концепцією поділу структури і уявлення.

Три атрибути, описані вище, можна застосувати до всіх елементів, крім елементів `BASE`, `HEAD`, `HTML`, `META`, `SCRIPT`, `STYLE` і `TITLE`.

Атрибут `title` визначає заголовок елемента і часто використовується оглядачами як підказка, яка виводиться на екран, коли курсор поміщається на даний елемент. Ми можемо використати атрибут `title` спільно з елементом `A`, щоб створити посилання на веб-сторінку:

**<A href="http://www.microsoft.com"
title="Офіційний сайт Microsoft">Microsoft**

Атрибут `lang` визначає мову, на якій написані значення інших атрибутів даного елемента і його вміст, а також всіх вкладених елементів,

які не мають аналогічного атрибута. Використання атрибута дозволяє авторам змінювати стиль тексту в залежності від мови.

Наприклад, текст двомовного документа може бути оформлений наступним чином:

...

<P lang="en">This paragraph is in English. </P>

<P lang="ua">Цей абзац українською мовою. </P>

...

Крім того, атрибут `dir` може бути використаний для визначення напрямку тексту всередині елементів, які містяться в багатомовних документах. Наприклад, якщо документ містить текст арабською та англійською мовами, то для правильного відображення тексту в браузері потрібно вказати напрямок виведення тексту для кожної з цих мов окремо.

Стандарт Unicode – це алгоритм, який можна застосовувати до документів, що містять символи, які виводяться справа наліво. Він дозволяє правильно відображати текст. Проте, іноді потрібно явно задати напрямок виведення тексту за допомогою атрибута **`dir`**.

Спеціальні символи та посилання–мнемоніки

У документі HTML можуть бути символічні посилання, також відомі як посилання–мнемоніки. Вони призначені для введення спеціальних або нестандартних символів в документ. Нестандартні символи є складними для введення за допомогою клавіатури або у конкретному кодуванні

документа, тоді як спеціальні символи представляють собою початок або кінець частин документу HTML, а не символи як такі, наприклад <, >, &, ".

Посилання-мнемоніки починаються з символу &, і закінчуються крапкою з комою (;). Посилання можуть бути представлені або числово (числові посилання), або скороченою формою слова (об'єктні посилання).

Наприклад, символ амперсанд може бути введений в документ як &, що є об'єктним посиланням, або як &, що є числовим посиланням.

Повну таблицю символічних посилань можна знайти на сайті [evolt.org](http://www.evolt.org/article/A_Simple_Character_Entity_Chart/17/21234) (http://www.evolt.org/article/A_Simple_Character_Entity_Chart/17/21234).

Коментарі в HTML

HTML-документи містять коментарі для опису коду, пошуку розділів або пояснення мотивів, які керували розробником. Коментарі не впливають на відображення документа, а тільки допомагають зрозуміти його вміст під час перегляду HTML-коду. Текст коментарів міститься між двома обмежувачами: відкриваючим (<!--) та закриваючим (-->). У тексті коментарів не можна використовувати два або більше поспіль йдуть дефіси. Коментарі можуть мати наступний вигляд:

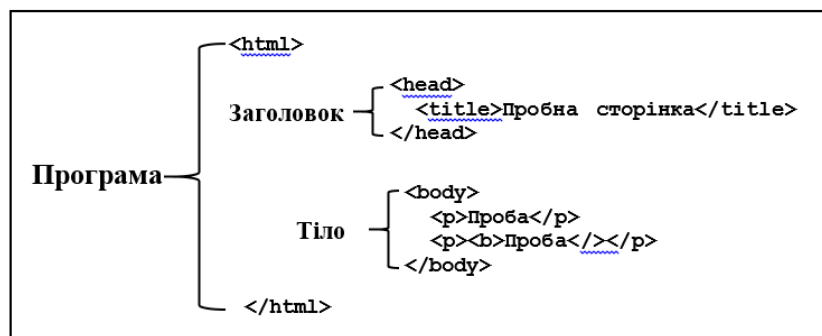
<!-- це коментар -->

<!-- це коментар,

який займає два рядки -->

Структура HTML-документа

Документи, які створюються за допомогою мови HTML, мають стандартну структуру.



Кожна HTML-сторінка починається тегом **<html>** і закінчується тегом **</html>**, що повідомляє браузеру про те, що це HTML-документ. Весь вміст між цими тегами називається документом або веб-сторінкою.

Деякі HTML-редактори автоматично вставляють ідентифікатор (покажчик) версії мови HTML на початку документу. Цей рядок називається прологом.

Кожен документ рекомендується починати з елемента декларації типу документа або DOCTYPE, який описує тип і версію HTML. Ця інформація призначена для браузерів і програм-валідаторів розмітки. DOCTYPE вказує браузеру синтаксичні елементи для кожної версії HTML і режим відтворення. Валідатори розмітки використовують DOCTYPE для визначення правил перевірки документа.

Після DOCTYPE слід елемент **<HTML>...</HTML>**, який показує, що наступний документ є HTML-документом і містить весь вміст. Цей елемент є кореневим елементом документа і може містити елементи **HEAD** і **BODY**. Кожен з цих елементів може з'являтися всередині елемента HTML не більше одного разу.

Елемент **HEAD** створює заголовок документа і містить загальну інформацію про нього. Він обов'язково повинен містити елемент **TITLE** і може містити інші елементи, такі як **BASE, SCRIPT, STYLE, META, LINK** і **OBJECT**.

Після елемента **HEAD** слід елемент **<BODY>...</BODY>**, який містить весь вміст сторінки. Елемент **BODY** обов'язково повинен містити хоча б один блочний елемент, наприклад, елемент **P**, що створює параграф.

У даному випадку, цей елемент містить текст "Проба". Елемент **BODY** має багато атрибутів.

ТИПИ ТЕГІВ

У HTML кожен тег відноситься до певної групи тегів. Наприклад, теги, які стосуються таблиць, не можна використовувати для інших цілей.

Теги можна розділити на кілька типів, таких як теги:

- верхнього рівня,
- теги заголовка документа,
- блокові та рядкові елементи,
- універсальний елемент,
- списки,
- таблиці та фрейми.

Варто зазначити, що той самий тег може відноситися до різних груп. Наприклад, теги **** та **** відносяться до категорії списків, але є також блоковими елементами.

Теги верхнього рівня

Теги **<html>**, **<head>** та **<body>** відносяться до структурних тегів HTML і використовуються для формування розділу заголовка та тіла веб-сторінки.

Тег **<html>** є контейнером, який містить весь вміст сторінки, включаючи теги **<head>** і **<body>**. Він не обов'язковий, але його використання є хорошим стилем.

Тег **<head>** призначений для зберігання інших елементів, таких як метатеги, які допомагають браузеру в роботі з даними. Метатеги зберігають інформацію для браузерів та пошукових систем, наприклад, опис сайту, ключові слова та інші дані.

Тег **<body>** призначений для зберігання вмісту веб-сторінки, який відображається у вікні браузера. Всю інформацію, яку потрібно виводити на сторінці, слід розташовувати всередині контейнера **<body>**. До цієї інформації можуть належати текст, зображення, таблиці, списки та інші елементи.

Теги заголовка документа

Ці теги використовуються для створення елементів, які знаходяться в контейнері **<head>** веб-сторінки. Зазвичай вони не відображаються на сторінці, крім тега **<title>**, який визначає заголовок веб-сторінки.

<tag title="">. Цей тег використовується для відображення тексту заголовка веб-сторінки в лівому верхньому куті вікна браузера та на

вкладці. Заголовок повідомляє користувачеві про назву сайту та додаткову інформацію, яку надає розробник.

<tag meta="">. Метатеги використовуються для зберігання інформації, призначеної для браузерів та пошукових систем. Наприклад, механізми пошукових систем використовують метатеги для отримання опису сайту, ключових слів та інших даних. Хоча тег **<meta>** є одним, він має кілька атрибутів, тому до нього може застосовуватися декілька властивостей.

Блокові елементи

У HTML є декілька тегів для форматування тексту та розміщення елементів на сторінці. Один з них – тег **<blockquote>**, який використовується для виділення довгих цитат всередині документа. Цей тег додає відступи зліва та справа (приблизно по 40 пікселів) та порожній простір зверху і знизу, щоб виділити текст від інших елементів на сторінці.

Інший тег – **<div>** – відноситься до універсальних блокових контейнерів. Він використовується для створення блоків з додатковими властивостями або для вирівнювання тексту всередині контейнера за допомогою атрибута align.

Теги **<h1>**, ..., **<h6>** використовуються для визначення текстових заголовків різного рівня. Їх важливість визначається відносною вагою розділу, що йде після заголовка.

Тег **<hr>** використовується для малювання горизонтальної лінії на сторінці. Лінія завжди починається з нового рядка, а після неї всі елементи відображаються на наступному рядку.

Тег **<p>** використовується для визначення параграфа тексту. Він додає відступ перед текстом, який його включає.

Тег **<pre>** використовується для відображення попередньо форматowanego тексту, зазвичай з моноширинним шрифтом та усіма пробілами між словами. Цей тег дозволяє розробникам відображати текст так, як вони хочуть, обійшовши особливості HTML, що згладжують прогалини між словами.

Теги **<big>**, ****, **<small>**, **<sub>** і **<sup>** не повинні використовуватись всередині контейнера **<pre>**.

Рядкові елементи

Веб-сторінки мають різні елементи, з яких одні є безпосередньою частиною інших. Наприклад, рядкові елементи можуть бути частиною текстового абзацу та використовуватись для зміни його вигляду.

Одним з найважливіших елементів HTML є тег **<a>**, який дозволяє створювати посилання на інші сторінки. Тег **<a>** може встановлювати посилання або якір, залежно від наявності атрибутів `name` або `href`.

Тег **** визначає жирний шрифт, тоді як тег **<big>** збільшує розмір шрифту.

У HTML розмір шрифту вимірюється в умовних одиницях від 1 до 7, і середній розмір тексту, використовуваний за замовчуванням, прийнятий 3.

Тег **
** встановлює перенесення рядка в тому місці, де цей тег знаходиться, і не додає порожнього відступу перед рядком, на відміну від тега **<p>**.

Тег **** призначений для акцентування тексту, а браузері відображають його курсивним шрифтом.

Тег **<i>** встановлює курсивне накреслення шрифту, а тег **** використовується для відображення зображень на веб-сторінках у форматах GIF, JPEG або PNG.

Якщо потрібно, зображення можна зробити посиланням на інший файл, помістивши тег **** в контейнер **<a>**. Зображення відображається з рамкою, яку можна видалити за допомогою атрибута **border="0"**.

Тег **<small>** зменшує розмір шрифту, а тег **** є універсальним і призначеним для визначення рядкових елементів всередині документа.

Тег **** призначений для акцентування тексту, і браузері відображають його жирним шрифтом.

<sub>. Відображає шрифт у вигляді нижнього індексу. Текст при цьому розташовується нижче базової лінії інших символів рядка і зменшеного розміру — H₂O.

<sup>. Відображає шрифт у вигляді верхнього індексу. За своєю дією схожий на **<sub>**, але текст відображається вище базової лінії тексту — м².

Різниця між блоковими і рядковими елементами наступна:

- рядкові елементи можуть містити тільки дані або інші малі елементи, а в блокові допустимо вкладати інші блокові елементи, рядкові

елементи, а також дані. Іншими словами, рядкові елементи ніяк не можуть зберігати блокові елементи;

- блокові елементи завжди починаються з нового рядка, а рядкові таким способом не акцентуються;
- блокові елементи займають всю доступну ширину, наприклад, вікна браузера, а ширина рядкових елементів дорівнює їх вмісту плюс значення відступів, полів і меж.

Універсальні елементи

Ці теги можуть використовуватися як блокові або вбудовані елементи, залежно від контексту.

Тег **** використовується для виділення тексту, який був видалений в новій версії документа. Використання цього тегу дозволяє відстежити зміни, які були зроблені у тексті. Браузери зазвичай позначають текст у контейнері **** як перекреслений.

Тег **<ins>** призначений для акцентування новоствореного тексту і зазвичай застосовується поряд з тегом ****.

Вміст контейнера **<ins>** браузері позначають підкресленням тексту.

Теги для списків

Список – це набір фраз або речень, які починаються з маркера або цифри, що допомагає упорядковувати та систематизувати дані для користувача.

Тег **** встановлює нумерований список, тобто кожен елемент списку починається з числа або букви та збільшується по порядку.

Тег **** встановлює маркований список, кожен елемент якого починається з невеликого символу-маркера.

Тег **** визначає окремий елемент списку.

Для створення списку визначень існує трійка елементів **<dl>**, **<dt>**, та **<dd>**. Кожен список визначень починається з контейнера **<dl>**, де тег **<dt>** створює термін, а тег **<dd>** задає визначення цього терміна. Хоча закривальний тег **</dd>** не є обов'язковим, краще його використовувати для кращого форматування.

Теги для таблиць

Таблиця – це сітка з комірок, які містять текст або зображення. Вона дозволяє організувати і візуалізувати табличні дані.

Тег **<table>** встановлює контейнер для вмісту таблиці.

Ця таблиця складається з рядків, що визначаються за допомогою тега **<tr>**, і комірок, що визначаються за допомогою тегу **<td>**.

Тег **<td>** використовується для створення окремої комірки таблиці.

Він має розміщуватися всередині тега **<tr>**, який, у свою чергу, має розміщуватися всередині тега **<table>**.

Тег **<th>** використовується для створення заголовкової комірки таблиці. Текст в такій комірці зазвичай відображається жирним шрифтом і вирівнюється по центру.

Тег **<tr>** використовується для створення рядка таблиці.

Теги для фреймів

Фрейми – це області, які можна створити в вікні браузера і заповнити веб-сторінками. Кожен фрейм містить свою власну веб-сторінку, яка визначається за допомогою тега **<frame>**.

Фрейми зазвичай використовуються для поділу веб-сторінки на дві або більше частини, які містять навігацію та контент сайту. Фрейми можуть також використовуватися для зв'язку між різними документами, наприклад, посилання в одному фреймі може відкривати документ в іншому фреймі.

Тег **<frame>** використовується для визначення окремого фрейму, який буде розміщений в вікні браузера.

Тег **<frameset>** використовується для створення структури фреймів, і замінює елемент **<body>** на веб-сторінці.

Тег **<iframe>** створює вкладений фрейм в межах звичайної веб-сторінки, і дозволяє завантажувати будь-який документ в цей фрейм. Використання вкладених фреймів може бути корисним, коли необхідно розділити фрейми на менші області.

Значення атрибутів тегів

Атрибути тегів – це спеціальні параметри, які можна додавати до тегів для зміни їх поведінки та вигляду на веб-сторінці. Існує багато різних атрибутів, але їх значення можна розбити на групи за типами налаштувань, наприклад, колір, розмір, адреса та інші. Давайте розглянемо основні типи атрибутів.

Наприклад, атрибути, що визначають колір, дозволяють змінити фоновий колір, колір тексту та інші кольорові настройки.

Атрибути, які визначають розмір, дозволяють задати ширину та висоту елементів, розмір шрифту та інші розмірні параметри.

Атрибути адреси дозволяють встановлювати посилання на інші веб-сторінки або додаткові файли.

Таким чином, атрибути тегів є потужним інструментом для налаштування веб-сторінок та розширення можливостей тегів.

Колір

Колір є важливим елементом веб-дизайну, оскільки він може підкреслювати або відокремлювати окремі елементи веб-сторінки. У HTML, колір можна задати за допомогою атрибуту "color" в більшості тегів.

Існує кілька способів визначення кольору в HTML. Один з найпоширеніших способів – це використання назви кольору. Наприклад, для встановлення червоного кольору, ви можете використовувати слово "red". Є багато назв кольорів, які можна використовувати в HTML, такі як "green", "blue", "yellow", "black" та багато інших.

Також, ви можете визначити колір за допомогою шістнадцяткового коду. Кожен колір складається з шести цифр, які відображають червоний, зелений та синій кольори. Наприклад, червоний колір може бути визначений як "#FF0000", де FF відображає червоний колір, а решта – зелений і синій. Шістнадцяткові коди надають вам більшу гнучкість

визначення кольору, оскільки ви можете використовувати будь-яку комбінацію червоного, зеленого та синього.

Також, ви можете використовувати функцію "rgb" для визначення колірних значень. Функція rgb приймає три параметри, які відображають червоний, зелений та синій кольори. Наприклад, колір, який складається з червоного кольору 255, зеленого кольору 0 та синього кольору 0, можна визначити як "rgb(255, 0, 0)".

Розмір

У HTML, розмір елементів може бути заданий за допомогою різних одиниць виміру.

1. Пікселі (px)

Піксель є стандартною одиницею виміру для екрану комп'ютера. Зазвичай, для задання розміру елемента використовуються пікселі. Наприклад, якщо ви хочете задати ширину елемента в 200 пікселях, то ви можете використовувати наступний код:

<div style="width:200px;"> Це мій елемент </div>

2. Відсотки (%)

Іншим способом задати розмір елемента є використання відсотків. Відсоток відноситься до розміру елемента батьківського елемента. Наприклад, якщо ви хочете, щоб елемент був рівний половині ширини батьківського елемента, то ви можете використовувати наступний код:

<div style="width:50%;"> Це мій елемент </div>

3. Ем (em)

Em є розміром, що базується на розмірі шрифту батьківського елемента. Наприклад, якщо розмір шрифту батьківського елемента становить 16 пікселів, то 1em буде дорівнювати 16 пікселям. Це дає можливість використовувати відносні розміри, які змінюються зміною розміру шрифту. Наприклад:

```
<div style="font-size:16px;">
  <p style="font-size:2em;">Це мій елемент</p>
</div>
```

У цьому прикладі, розмір шрифту елемента <p> буде дорівнювати 32 пікселям (2em x 16 пікселів).

Адреса

У HTML, атрибут href використовується для створення посилань на інші сторінки в Інтернеті. Цей атрибут додається до тегу <a> (anchor), який є основою для створення посилань.

Наприклад, якщо ви хочете створити посилання на головну сторінку Google, ви можете використовувати такий код:

```
<a href="https://www.google.com">Google</a>
```

Тут <https://www.google.com> є адресою, на яку буде переходити користувач при натисканні на посилання.