

Тема 4

Каскадні таблиці стилів CSS. Поняття XHTML

CSS – Cascading Style Sheets – технологія опису зовнішнього виду документа, який написаний мовою розмітки. Іншими словами: це набір параметрів форматування, що застосовується до елементів документа, щоб змінити їх зовнішній вигляд. Як правило, використовується як засіб оформлення веб-сторінок в форматі HTML і XHTML, а також може застосовуватися до інших XML-документів.

Основною метою розробки CSS було розділення опису логічної структури веб-сторінки (який здійснюється за допомогою HTML або інших мов розмітки) від опису зовнішнього вигляду цієї веб-сторінки (який тепер здійснюється за допомогою формальної мови CSS).

Типи стилів

Розрізняють декілька типів стилів, які можуть спільно застосовуватися до одного документа. Це стиль браузера, стиль автора і стиль користувача.

Переваги стилів:

- розмежування коду і оформлення. В ідеалі, веб-сторінка повинна містити тільки теги логічного форматування, а вид елементів задається через стилі. При подібному розподілі робота над дизайном і версткою сайту може вестися паралельно;
- різне оформлення для різних пристроїв. За допомогою стилів можна визначити вид веб-сторінки для різних пристроїв виведення: монітора, принтера, смартфона, КПК і ін. Наприклад, на екрані монітора відображати сторінку в одному оформленні, а при її друці – в іншому. Ця можливість також дозволяє приховувати або показувати деякі елементи документа при відображенні на різних пристроях;
- розширені в порівнянні з HTML способи оформлення елементів. На відміну від HTML стилі мають набагато більше можливостей по оформленню елементів веб-сторінок. Простими

засобами можна змінити колір фону елемента, додати рамку, встановити шрифт, визначити розміри, положення та ін.;

- прискорення завантаження сайту. При зберіганні стилів в окремому файлі, він кешується і при повторному зверненні до нього витягується з кеша браузера. За рахунок кешування і того, що стилі зберігаються в окремому файлі, зменшується код веб-сторінок і знижується час завантаження документів;

- єдине стильове оформлення великої кількості документів. Сайт – це не просто набір зв'язаних між собою документів, але і однакове розташування основних блоків, і їх вигляд. Застосування одноманітного оформлення заголовків, основного тексту і інших елементів створює наслідуваність між сторінками і полегшує користувачам роботу з сайтом і його сприйняття в цілому. Розробникам використання стилів істотно спрощує проектування дизайну;

- централізоване зберігання. Стилi, як правило, зберігаються в одному або декількох спеціальних файлах, посилання на які указується у всіх документах сайту. Завдяки цьому зручно правити стиль в одному місці, при цьому оформлення елементів автоматично міняється на всіх сторінках, які пов'язані з вказаним файлом.

Синтаксис CSS. Стильові правила записуються в форматі, відмінному від HTML. Основним поняттям виступає селектор – це деяке ім'я стилю, до якого додаються параметри форматування. Як селектор виступають теги, класи і ідентифікатори. Загальний спосіб запису має наступний вигляд:

селектор {властивість1: значення; властивість2: значення; ...}

Спочатку пишеться ім'я селектора, наприклад, TABLE, це означає, що всі стильові параметри будуть застосовуватись до тегу <table>, потім у фігурних дужках записується стильова властивість, а її значення вказується після двокрапки. Стильові властивості розділяються між собою крапкою з комою, в кінці цей символ можна опустити.

CSS не чутливий до регістра, перенесення рядків, пропусків і символів табуляції, тому форма запису залежить від бажання розробника.

Правила застосування стилів:

- **форма запису** – для селектора допускається додавати кожен стильову властивість і його значення окремо:

```
td {background: olive}
td {color: white}
td {border: 1px solid black}
```

Проте такий запис не дуже зручний. Тому доцільно використовувати такий спосіб:

```
td {background: olive;
    color: white;
    border: 1px solid black;
}
```

Ця форма запису більш наочна і зручна у використанні;

- **має пріоритет значення, вказане в коді нижче** – якщо для селектора спочатку задається властивість з одним значенням, а потім та ж властивість, але вже з іншим значенням, то застосовуватися буде те значення, яке в коді встановлене нижче;
- **значення** – у кожної властивості може бути тільки відповідне його функції значення. Наприклад, для color, який встановлює колір тексту, як значення неприпустимо використовувати числа;
- **коментарі** – щоб позначити, що текст є коментарем, застосовують наступну конструкцію /* ... */. Коментарі можна додавати в будь-яке місце CSS-документа, а також писати текст коментаря в декілька рядків. Вкладені коментарі недопустимі. Коментарі звичайно застосовують з метою налагодження або з метою навчання, а при викладанні сайту в мережу їх знищують;
- **значення стильових властивостей** – стильові властивості можуть приймати значення таких типів: рядок, число, відсотки, розмір, колір, адреса або ключове слово.

Підключення таблиці стилів.

Коли браузер зчитує таблицю стилів, він форматує документ згідно цієї таблиці. Існує три способи підключення таблиці стилів:

1) підключення зовнішньої таблиці стилів (зв'язані стилі) застосовується в ситуаціях, коли один стиль визначається для багатьох сторінок. Для підключення зовнішньої таблиці стилів призначений тег **<link>**, який розміщується в розділі заголовка:

```
<head>
<link rel="stylesheet" type="text/css" href="first.css">
</head>
```

Зовнішню таблицю стилів можна створити в довільному редакторі. Файл зовнішньої таблиці стилів не може містити ніяких тегів HTML. Наприклад:

```
H1 {color: #000080;
    font-size: 200%;
    font-family: Arial;
    text-align: center;}
P {padding-left: 20px;}
```

Розширення файла таблиці стилів – .css. Значення параметрів rel і type залишаються незмінними незалежно від коду.

2) підключення внутрішньої таблиці стилів (глобальні стилі) застосовується тоді, коли один документ має унікальний стиль. Для визначення внутрішніх стилів використовують тег **<style>**, який розміщується в розділі заголовка:

```
<head>
<style type="text/css">
    li {font-family: arial}
    p {font-size: 25%}
    h1 {margin-right: 10pt}
</style>
</head>
```

3) вбудовані стилі (внутрішні стилі) використовуються в тому випадку, коли необхідно застосувати стиль для одного елемента. Недолік: він змішує вміст документа з його представленням і втрачає багато переваг таблиць стилів:

```
<body style="background-color: blue" >  
<h1 style="color: yellow" >Заголовок H1</h1>  
</body>
```

Всі описані способи використання CSS можуть застосовуватися як окремо, так і сумісно один з одним. Але при цьому першим завжди застосовується внутрішній стиль, потім глобальний, а в останню чергу – зв'язаний.

Класи. Класи застосовують, коли необхідно визначити стиль для індивідуального елемента веб-сторінки або задати різні стилі для одного тега. При використанні спільно з тегами синтаксис для класів буде наступний:

тег.і'мя_класу {властиві1: значення; властиві2: значення; ...}

Всередині стилю спочатку пишеться потрібний тег, а потім, через крапку, призначене для користувача ім'я класу. Імена класів повинні починатися з латинського символу і можуть містити в собі символ дефіса (-) і підкреслення (_). Щоб вказати в коді HTML, що тег використовується з певним класом, до тегу додається параметр `class="і'мя_класу"`.

Можна, також, використовувати класи і без вказівки тега. Синтаксис в цьому випадку буде наступний:

.ім'я_класу {властивість1: значення; властивість2: значення; ... }

При такому записі, клас можна застосовувати до будь-кого тегу.

Класи зручно використовувати, коли потрібно застосувати стиль до різних елементів веб-сторінки: елементів таблиці, посилань, абзаців і ін.

Одночасне використання різних класів

До будь-кого тегу одночасно можна додати декілька класів, перераховуючи їх в параметрі `class` через пробіл. В цьому випадку до елемента застосовується стиль, описаний в правилах для кожного класу. Оскільки при додаванні декількох класів вони можуть містити однакові стильові властивості,

але з різними значеннями, то береться значення з класу, який описаний в коді нижче.

Ідентифікатори. Ідентифікатор (ID-селектор) визначає унікальне ім'я елемента, яке використовується для зміни його стилю і звернення до нього через скрипти. Синтаксис:

#ім'я_ідентифікатора {властивість1: значення; властивість2: значення; ... }

Ім'я ідентифікатора повинне починатися з латинського символу і може містити в собі символ дефіса (-) і підкреслення (_). На відміну від класів ідентифікатори повинні бути унікальними, тобто зустрічатися в коді документа тільки один раз.

Звернення до ідентифікатора відбувається аналогічно класам, але як ключове слово у тега використовується параметр **id**, значенням якого виступає ім'я ідентифікатора. Символ решітки при цьому вже не вказується.

Як і при використанні класів, ідентифікатори можна застосовувати до конкретного тегу. Синтаксис при цьому буде наступний.

тег#ім'я_ідентифікатора {властивість1: знач; властивість2: знач; ... }

XHTML (Extensible Hypertext Markup Language, розширювана мова розмітки гіпертексту) – сімейство мов розмітки веб-сторінок на основі XML, що повторюють і розширюють можливості HTML 4, призначена для заміни HTML і вважається його більш суворою версією.

Правила XHTML:

- всі теги і їх атрибути повинні бути набрані в нижньому регістрі (малими символами);
- значення будь-яких атрибутів необхідно брати в лапки;
- потрібно закривати всі теги, навіть такі, які не мають закриваючого тегу;
- повинна виконуватися правильна вкладеність тегів;
- не можна використовувати скорочені атрибути тегів;
- замість атрибуту name слід вказувати id;

- слід визначати DTD (document type definition – опис типу документа) за допомогою елемента <!DOCTYPE>.

Запитання для самоконтролю

- 1. Що таке CSS?*
- 2. Для чого призначені CSS?*
- 3. Що таке селектор?*
- 4. Описати загальний синтаксис задання селектору.*
- 5. Які типи стилів існують?*
- 6. Перерахувати переваги використання CSS.*
- 7. Чи чутливі CSS до регістру, символів абзаців, пробілів та табуляції?*
- 8. Якими способами можна підключати CSS до веб-сторінки?*
- 9. Для чого призначений кожний із способів?*
- 10. Як відбувається підключення CSS в кожному із випадків?*
- 11. Для чого використовуються класи в CSS? Синтаксис написання класу.*
- 12. Для чого використовуються ідентифікатори в CSS? Синтаксис написання ідентифікатора.*
- 13. Що таке XHTML?*
- 14. Чим відрізняється XHTML від HTML?*