Блокова модель CSS. Проектування та верстка веб-сторінок

21.1. Блокова модель CSS.

Верстка сайту на сьогоднішній день може бути виконана безліччю різних способів. Розглянемо найпоширеніші з них.

- **HTML-верстка сайту.** HTML-верстка сайту - це верстка сайту на основі мови гіпертекстової розмітки HTML. Залежно від основних елементів html, що використовуються при верстці сайту, виділяють табличну html-верстку й блокову html-верстку.
- **Таблична верстка сайту** - це верстка, при якій структура сторінки сайту представлена у вигляді таблиці. Кожний елемент сторінки - це одна або кілька комірок таблиці. Таблична верстка сайту зручна й широко застосовується верстальниками, однак вона не завжди може задовольнити потреби сайту зі швидкості завантаження, наприклад.
- **CSS-верстка сайту**. CSS-верстку іноді виділяють в окремий вид верстки сайту, хоча в сучасному веб-дизайні каскадні таблиці стилів (css) використовуються практично при будь-якій верстці. Каскадні таблиці стилів описують зовнішній вигляд сторінок сайту, написаних мовою розмітки: HTML, XHTML, XML.
- **Блокова верстка сайту**, або як її ще називають верстка **div'ами**, - це верстка сайту на основі елементів ** **. Така верстка сайту має ряд переваг: стисліть коду, висока швидкість завантаження коду й т.д.

```
<div id="blok1"> вміст блоку 1</div>
<div id="blok2"> вміст блоку 2</div>
<div id="blok3"> вміст блоку 3</div>
...
```

Тег ** ** - це так званий контейнер (блок), який може містити форматований текст, зображення та ін. Важливою особливістю блоків є їх здатність накладатися один на одного при верстці. Ця особливість надає блокам набагато більше можливостей по верстці сайту, ніж, приміром, таблиці.

Верстка div активно використовує CSS для гнучкого й зручного форматування елементів веб-сторінок. За допомогою CSS можна з точністю до пікселя задати розташування блоків на сторінці, необхідні відступи, колір, розмір шрифтів і багато іншого.

Переваги блокової верстки:

- блоки легко можна переміщати або «ховати» без перезавантаження всієї сторінки, що дозволяє отримувати динамічні ефекти (меню, що випадає, спливаючі підказки);
- блокову верстку сайту зручно застосовувати для одержання «гумового» дизайну, оскільки блоки легко позиціонувати відносно меж вікна браузера, накладати один на одного і т. д.;
- верстка div дозволяє позбавитися від «розпорок» - прозорих зображень в 1 піксель, що використовуються для фіксації відстані між комірками таблиць при табличній верстці;
- програмний код при блоковій верстці сайту виходить більш компактним і зрозумілим; вважається, що завантаження сторінки з блоковою версткою виконується швидше, ніж з табличною (проте не завжди це так).

Недоліки блокової верстки:

Наприклад, завдання кросбраузерності сайту вирішується складніше при використанні блокової верстки, оскільки різні браузери по-різному інтерпретують html-код сторінок з блоками.

Тому блокова верстка сайту нерідко застосовується в комплексі з табличною версткою. У цьому випадку застосування кожного виду верстки сайтів для вирішення конкретних завдань виявляється більш привабливим. Наприклад, часто «каркас» сторінки задається за допомогою таблиці, а комірки цієї таблиці заповнюються необхідними блоками.

Таким чином, верстка div не є строго обов'язковою і повинна використовуватися в міру необхідності. Професійна верстка сайту, будь вона блоковою або табличною, завжди правильно відображається в різних браузерах (кроссбраузерна), не містить зайвого коду, сприяє швидкому завантаженню сторінок.

Отже, єдині загальноприйняті правила верстки не існують, і в кожному конкретному випадку слід діяти по ситуації.
Однак, **табличний метод верстки, незважаючи на зазначені недоліки, практикується повсюдно і зарекомендував себе як максимально надійний**.

Переваги та недоліки табличного та блокового способів:

	Таблиця	**Блоки та стилі**
Колонки	Колонки формуються елементами таблиці, їх висота однакова і взаємопов'язана.	Колонки створюються різними блоками, їх висота різна і залежить від змісту.
Ширина	Якщо ширина таблиці явно не вказана, вона обчислюється на основі вмісту таблиці.	За замовчуванням блок займає всю доступну йому ширину.
Розташування	Рядки таблиці відображаються в тому порядку, як вони представлені в коді. Клітинки йдуть зліва направо і зверху вниз.	Порядок блоків у коді може не відповідати їх становищу в браузері.
Завантаження	Як правило, поки таблиця не завантажиться повністю, вміст її не відображатиметься. Якщо на веб-сторінці розміщена велика таблиця, завантаження сторінки істотно сповільнюється.	Блоки відображаються послідовно, у міру завантаження документа.
HTML-код	Код для створення таблиць може бути громіздкий, особливо якщо потрібно об'єднати багато клітинок або зробити кілька вкладених таблиць.	Код, як правило, компактний.
Вирівнювання по висоті	Вміст всередині клітинки легко вирівнювати по її верхньому, нижньому краю або середині.	Вирівнювання нетривіальне.

21.2. Проектування та верстка веб-сторінок

Процес створення сайту складається з таких етапів:

I. Постановка мети.

На цьому етапі вирішуються питання загального характеру:

- аналіз web-ресурсів суміжної тематики;
- попередній аналіз цільової аудиторії;
- мета і завдання розроблюваного ресурсу;

- постановка завдання.

II. Технічне завдання.

Тут задаються основні параметри візуального представлення та структури сайту. ТЗ – це знання розробника про те, як треба реалізувати ту або іншу функцію для того, щоб нею було зручно користуватися. Ця область знань називається usability.

III. Проектування.

Цей етап включає широкий ряд питань від продумування логічної структури веб-сторінок до вибору найбільш зручних форм подачі інформації. Проектуванням займається дизайнер, менеджер проекту. Можна виділити три основні частини, на які слід звернути увагу: **структура**, **способи зберігання даних і доступу до них** і на **шлюзи з іншими програмами**.

- **Структура**. Якщо ми говоримо про найпростіший статичний Web-сайт, то він складається з набору файлів, які містять інформацію. Між ними є зв'язок. Такий сайт функціонує відповідно до базової логіки роботи Webсайтів. Якщо створюється складний динамічний сайт, який працює з базою даних і дозволяє оновлювати свій вміст у режимі on-line, то такий сайт буде містити підпрограми, які працюють на сервері, і підпрограми, які завантажуються в браузер і працюють на машині користувача. Необхідно розробити схеми цих підпрограм (компонентів сайту) та їх взаємодії. Зокрема, **розробляються принцип генерації сторінок, принцип формування інформаційного наповнення, принцип побудови меню та принцип авторизації.**
- **Зберігання даних.** Розрізняють два типи даних: **статичні** й **динамічні**. **Статичні** жорстко «зашиті» в програму, змінити їх можна тільки виправивши сам текст програми. Наприклад – «Історія компанії». Звичайно статичний сайт складається тільки зі сторінок із незмінним вмістом. Для його зміни потрібно заходити всередину програми, що вимагає спеціальних знань. На противагу статичним, **динамічні** дані час від часу змінюються.
- **Шлюзи з іншими програмами.** Для того, щоб розроблювальний продукт міг отримувати інформацію з інших автоматизованих систем, потрібно розробити спеціальний компонент – шлюз. Існує дуже багато способів функціонування шлюзів. Все залежить від програм за якими буде здійснюватись зв'язок.

IV. Верстка

Інтерфейс сайту, який розроблений дизайнером, – це ще тільки макет остаточного інтерфейсу сайту. Фактично, він складається просто з набору малюнків. Для того, щоб його можна було використати в програмному продукті, потрібно провести верстку – розрізати макет інтерфейсу на складові його графічні компоненти і описати правила розташування всіх цих елементів на сторінці. Для опису використовується спеціалізована мова html мова розмітки гіпертекстових документів. Готові зверстані html шаблони далі використовуються в наступних етапах реалізації проекту. На цій стадії графічна картинка нарізається на окремі елементи і з використанням технологій HTML і CSS трансформується в код, який можна переглядати за допомогою браузера. Виконується кодером, програмістом.

Верстка сайту являє собою опис програмним кодом візуальної частини веб-сайту. Незалежно від того, який браузер використовує користувач, сайт повинен виглядати і працювати коректно при будь-якому дозволі монітора.

Процес верстки - один з найважливіших етапів створення інтернет-ресурсу, оскільки від роботи верстальника залежить:

- Швидкість завантаження веб-сайту;
- Відповідність стандартам HTML;
- Адекватність відображення в браузері;

- Відповідність вимогам пошукових систем;
- Адаптивність під різні екрани користувача.

Валідна верстка. Поняття верстки тісно переплітається з поняттям валідності. Дійсна верстка веб-сайту - написання HTML і CSS коду, який відповідає стандартам W3C і успішно проходить тест на валідаторі (валідний код - гарантія того, що верстальник не допустив логічних і синтаксичних помилок при програмуванні).
Сторінка навіть з незначними помилками в коді не пройде через валідатор. Слід пам'ятати, що кращий валідатор - браузер, оскільки сприйняття сайту браузером - це сприйняття сайту відвідувачем.

Базові правила верстки сайту:

- Верстка обов'язково повинна бути кросбраузерності для Firefox, Opera, Safari, Google Chrome і Internet Explorer, який останнім часом втрачає актуальність. Сайт повинен бути протестований на різних дозволах монітора, починаючи від 1024 на 768;
- Верстка абсолютно всіх сторінок сайту повинна пройти валідацію. Використання різних CMS, модулів і готових скриптів, вставка на інтернет-ресурс вмісту різних форматів часто ускладнюють досягнення валідності;
- Зовнішній вигляд зверстані сторінки повинен по максимуму відповідати дизайну: розміри шрифтів, відстані між рядками, відступи повинні відповідати параметрам psd макета. Досягти піксельної точності часто важко, тому допускається відхилення окремих елементів в межах 3-6 пікселів;
- CSS стилі повинні бути винесені в окремий документ. В HTML коді допустимо тільки присутність ідентифікаторів і класів;
- Логотип веб-сайту повинен бути посиланням на головну сторінку;
- HTML сторінки повинні містити коментарі до основних елементів, таким, як меню, заголовок, шапка, контент, футер і т.п.
- Імена ідентифікаторів і класів повинні відповідати призначенню і бути зрозумілими інтуїтивно (menu , footer , header і т.д.);
- У таблицях стилів настійно рекомендується використовувати однакові одиниці виміру для всіх величин;
- Написи на кнопках повинні бути написані однією мовою і або прописними буквами, або починатися з великої;
- Кнопки повинні мати стандартне оформлення та бути або графічними елементами, або бути налаштованими за допомогою таблиць стилів;
- HTML і CSS код повинен бути мінімізований. Слід уникати зайвих класів і ідентифікаторів і використовувати властивості спадкування;
- Заголовки повинні бути написані виключно за допомогою спеціальних тегів h1 , h2 і т.д;
- Атрибути всіх тегів повинні бути укладені в лапки.

Основні підходи до верстки сайту.
Припустимо, що існують два блоки, що стоять поруч. Існує кілька основних підходів до верстці сайту:
Фіксована верстка. При зміні розміру вікна браузера блоки не поміняють свою ширину, а на моніторах з низьким дозволом екрану з'явиться смуга прокрутки.
Гумова верстка. Залежно від розміру вікна браузера, блоки змінять свою ширину.
Адаптивна верстка. Втілюється в життя завдяки різним скриптам і заточена під певні дозволи (320, 768, 1024 і т.д.). Зміна розміру відбувається ривками після того, як певний рівень досягнутий.

Чуйна верстка. Являє собою злиття адаптивної і гумовою верстки. Є найскладнішою з технічної точки зору, але в той же час найефективнішою.
Версія сайту для мобільних пристроїв. Фактично є створенням іншого сайту з іншим дизайном, версткою і URL адресою.

Крім підходів, існують ще й різні типи верстки. (див. п.21.1)

V. Наповнення

Основне завдання веб-сайту – це надання інформації відвідувачам. Тому, після закінчення всіх основних етапів розробки, сайт необхідно наповнити інформацією. На практиці звичайно застосовується комбінований підхід – частину інформації публікують розроблювачі, а частину – замовник.

VI. Вебмастерінг

На цій стадії здійснюються дії, які сприяють поширенню та ефективної доставки сайту по мережі. Включає розміщення на хостингу і пошукову оптимізацію. Виконується адміністратором, вебмайстром.

VII. Розміщення

Після завершення розробки сайту, треба зробити його доступним для користувачів. Із цією метою сайт розміщається на сервері – потужному комп'ютері, який з'єднаний з Internet швидкою й надійною лінією. Таке розміщення називається хостингом (від англійського hosting) і звичайно оплачується щомісячно. Можливі два варіанти: встановити власний сервер, або розмістити вузол на сервері будь якої іншої організації.

VIII. Тестування

Як правило, тестування проводиться і замовником, і розроблювачем. У результаті виявляються і виправляються дрібні недоліки й неточності, виробляється коригування розміщеної інформації.

IX. Просування

Оскільки сайтів в Internet надзвичайно багато, то дуже важливо зробити все можливе, щоб про новий, щойно розроблений сайт довідались як найбільше людей. Для цього сайт реєструють у пошукових машинах і каталогах, малюють і показують банери, роблять адресну рекламу і т.д.

X. Підтримка

За готовим програмним продуктом треба стежити: міняти інформацію, усувати виникаючі в процесі роботи помилки, втілювати в життя нові ідеї. Все це входить у поняття «підтримка». Вона здійснюється звичайно за допомогою систем онлайнового керування, вбудованих у сайт, і, як правило, не містить у собі істотних змін сайту. У свою чергу, розроблювачі в рамках підтримки сайту можуть:

- оновлювати матеріали на сайті.
- міняти структуру сайту: додавати нові розділи, змінювати або видаляти старі;
- допомагати в аналізі функціонування сайту, а також вузлів конкурентів, надавати рекомендації щодо його зміни;
- додавати нові функції, розширювати можливості вже наявних функцій і сервісів сайту;
- робити графічні роботи: обробляти нові фотоматеріали, розробляти банери, анонси і т.і.
- писати тексти для нових розділів сайту;
- коригувати мета-теги залежно від зміни контенту.

Необхідно пам'ятати, що грамотна підтримка – це успішне функціонування сайту.