

Верстка веб-сторінок

Верстання вебсторінки — це процес створення вебсторінки із попередньо створеного макету дизайну сайту, заздалегідь намальованого за допомогою графічних редакторів.

До прикладу, поняття «**блочна верстка**», «**таблична верстка**» дають уявлення про тип побудови каркаса. Верстка, також, може здійснюватися фреймами.

Верстка сайту — це один з найважливіших етапів розробки онлайн-ресурсу, в результаті якого намальований дизайнером макет перетворюється в HTML і CSS-код. Це завдання вимагає особливих навичок. Щоб якісно зверстати HTML-код, потрібні глибокі знання особливостей роботи браузерів, семантики веб-сторінок, принципів позиціювання елементів.

Сучасні види верстки сайту

Раніше, щоб зверстати сайт, використовували HTML-таблиці: кожен елемент містився в окрему клітинку, що розв'язувало проблеми з позиціюванням контенту. Однак код виходив складним для підтримки. На зміну таблиць прийшла **блокова верстка**: елементи містилися в порожні, незалежні контейнери та розміщувалися на сторінці за допомогою різних CSS-властивостей.

Інша проблема, яку потрібно вирішити при верстці — зображення сторінок при різних розширеннях екрану.

Підхід, при якому розміри блоків та інших елементів вказуються у відсотках до ширини екрану або батьківських елементів називають версткою. Попри всі переваги, таке рішення не можна назвати ідеальним. Подібна буде погано виглядати на широких та вузьких екранах: в таких випадках контент буде або занадто розповзатися, або занадто стискатися. Вивчати його буде незручно.

Верстка — більш сучасне і правильне рішення, яке передбачає написання різних правил оформлення для екранів з різним розширенням. Це більш трудомісткий процес. Однак це автоматично вирішує задачу з підготовки мобільної версії сайту: спеціально робити нічого не потрібно, оскільки сторінка буде підлаштовуватися під невеликий екран.

До написаного коду пред'являється 2 вимоги. По-перше, верстка повинна бути валідною: грубі семантичні помилки не допускаються, бо це призводить до проблем з просуванням сайту в пошукових системах. Вплив відчутний. По-друге, сайт повинен бути **кросбраузерним**, тобто однаково виглядати у всіх популярних версіях браузерів, які використовує цільова аудиторія замовника.

Як зверстати веб-сторінку

Для якісної верстки сайту будуть потрібні спеціальні знання, які досить важко отримати у книгах та навчальних матеріалах — багато що вирішує досвід. Знадобляться графічний і текстовий редактори.

На першому етапі верстальник займається дизайном сайту — необхідно вивчити структуру макета, щоб визначити основні блоки та продумати стратегію роботи. Далі починається написання коду. Верстка веб-сторінок вимагає від виконавця посидючості та розумної частки перфекціонізму: результат повинен не відрізнятись від створеного дизайнером макета.

Коли весь код написаний, робота ще не завершена. Необхідно переконатися, що виконавець зверстав сайт правильно — для цього проводиться тестування коду в різних браузерах, при різних розширеннях екрану. Зазвичай цим займаються спеціально навчені люди або сам верстальник.

Важливо зрозуміти, правильною версткою сайтів можна назвати тільки таку роботу, яка відповідає вимогам замовника та виконує поставлені завдання.

Таблична верстка

Таблична верстка — це тип проектування вебсторінки, каркас якої створено за допомогою таблиць і певного розташування інформації в ній. Даний тип створення сторінок був розповсюджений до впровадження використання CSS, але після поступився більш ефективній моделі проектування сторінок - блочній верстці.

Блокова верстка

Блокова верстка — це тип дизайну сторінки, створений за допомогою блоків (тег `<div>`), розташування яких суворо контролюється через CSS (каскадних таблиць стилів), і в якому таблиці використовуються тільки для представлення табличних даних.

Перевагами даного типу є:

- Кешування блоків браузером, за рахунок чого швидке завантаження сторінок;
- Необмежені можливості по створенню складних дизайнів;
- Чітке відділення частини з контентом;
- Простота реалізації програмної частини;

Недоліком блокової верстки є складність реалізації відносно табличної для простих сайтів, а також виникає багато проблем із забезпеченням кросбраузерності для старих браузерів, але при цьому вона відмінно індексується пошуковиками, дозволяє багатьом програмам, що працюють із контентом сайтів, правильно «читати» його вміст, допомагає користувачам швидко завантажувати сторінки, а також за допомогою поєднання блоків і CSS можна створювати професійні дизайни.

Блокова верстка з `<div>`

За допомогою розмітки визначається розташування елементів на вебсторінці. Наприклад, ми створюємо шапку сайту, всередині якої розміщуємо необхідний нам контент (логотип, номери телефонів, навігаційне меню і т. ін.). Точно так само чинимо з основним розділом вебсторінки, а

також бічною панеллю і футером. Всі ці елементи, по суті, є певними HTML-тегами. І тегом номер один в блокової верстці є тег `<div>`.

Елемент `<div>` можна назвати своєрідним фундаментом для побудови вебсторінок. Спочатку він нічим не виділяється зовні, крім хіба що того факту, що це блоковий елемент, який за замовчуванням займає всю ширину документа, а наступний за ним елемент починається з нового рядка.

Тег `<div>` універсальний і часто служить контейнером для інших HTML-елементів. Наприклад, шапка сайту, підвал або сайдбар — це часто не що інше, як блок `div`, що містить в собі такі ж блоки і / або інші HTML-елементи.

Через різні класи і ідентифікатори `div`-блокам задається відповідне CSS-оформлення. Наприклад, щоб було зручніше писати стилі для шапки сайту, можна додати до блоку клас `.header`, а для підвала — клас `.footer`.

Використання тегів `<div>` у верстці не означає, що потрібно відмовитися від інших тегів. Якщо для вирішення ситуації інший тег підходить краще, використовуйте його. Яскравий приклад — тег довгою цитати `<blockquote>`. Погодьтеся, немає сенсу використовувати замість нього `<div>`, адже `<blockquote>` прекрасно впорається зі своїм завданням. Крім того, це рішення буде більш правильним з точки зору семантики. Блокова верстка з HTML5

І, якщо вже ми заговорили про семантику в вебдокументах, варто згадати і про теги, що з'явилися в HTML5. Нові семантичні теги були спеціально створені для угруповання контенту конкретного типу.

Наприклад, шапка сайту може міститися в HTML5-тегу `<header>`, а підвал — в схожий тег `<footer>`. Крім того, існує ще безліч інших семантичних тегів — `<nav>`, `<section>`, `<article>` тощо. Даними тегами можна замінювати звичайні теги `<div>` для об'єднання елементів.

Як ви могли здогадатися, HTML5-теги дають більш точну інформацію про те, що в них міститься. Це дає певні переваги в плані ідентифікування ролі того чи іншого контенту вебсторінки, однак з боку стилізації нічого не змінюється: ви можете точно так само застосовувати CSS до семантичних тегів, як застосовуєте їх до простих блоків `<div>`.

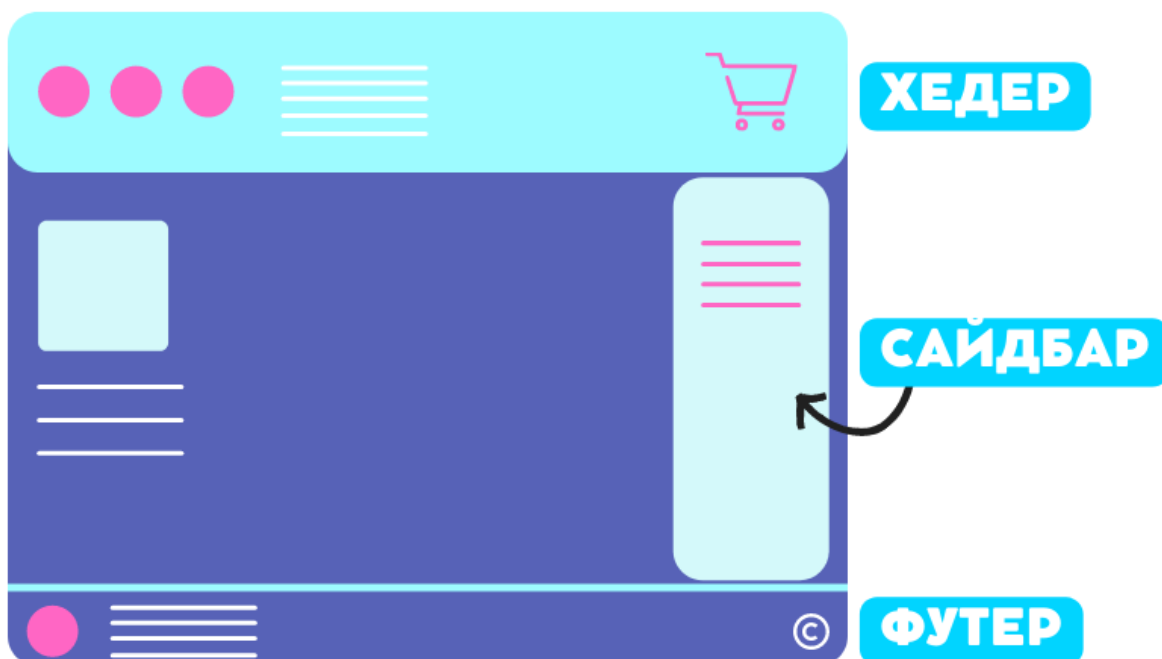
Структура сторінки. Терміни

Хедер і футер (header, footer) - інакше шапка та підвал сторінки. Шапка знаходиться вгорі веб-сторінки або мобільної програми. Як правило, туди поміщають логотип компанії або продукту, меню, кнопку, яка веде у профіль, особистий кабінет чи кошик. Футер, підвал знаходиться внизу сторінки або програми. Там часто дублюють меню, розширюючи його додатковою та технічною інформацією. Наприклад, там можна знайти посилання на докладний опис компанії, вакансії, контакти. Там знаходиться інформація про те, хто робив продукт і на чому він зроблений.

Фронтенд та **бекенд** (frontend, backend) — зовнішня та внутрішня частини ІТ-продукту. **Фронтенд**, зовнішня частина - усе, що бачить користувач і із чим безпосередньо взаємодіє. Це картинки та тексти, кнопки та форми, відео та анімації. Це все, що відбувається в браузері або в програмі на очах користувача. **Бекенд**, внутрішня частина

- все, що користувач не бачить або все, що знаходиться під інтерфейсом сайту або програми. Фактично роль бекенда в тому, щоб приймати запити та надсилати запити серверу. Але це складна обчислювальна робота — обробка інформації, що надходить та наявна, пошук і вилучення даних, динамічна (тобто змінюється) візуалізація сторінок (зміна фронтенду), перевірка безпеки.

Сайдбар (sidebar) — боковая панель-меню справа или слева. Сайдбар разграничивается контентом, который выводится там вертикально. Это может быть, например, расширенный каталог товаров или рубрик.



Стек (stack) - буквально "пачка" або пакет програм, які зібрані разом, пов'язані та інтегровані між собою, щоб продукт працював повноцінно. Це набір технологій, які збираються до купи, щоб здійснювати завдання сайту або додатки. У класичній розробці часто зустрічається такий стек: HTML + CSS + JavaScript + PHP + MySQL. Це набір мов програмування фронтенду, бекенду та бази даних.

Bootstrap

Bootstrap — це безкоштовний набір інструментів з відкритим кодом, призначений для створення вебсайтів та вебдодатків, який містить шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу, а також додаткові розширення JavaScript. Він спрощує розробку динамічних вебсайтів і вебдодатків.

Bootstrap — це клієнтський фреймворк, тобто інтерфейс для користувача, на відміну від коду серверної сторони, який знаходиться на сервері.

Фреймворк

(англ. Framework, каркас, платформа, структура, інфраструктура) — інфраструктура програмних рішень, що полегшує розробку складних систем. Спрощено дану інфраструктуру можна вважати своєрідною комплексною бібліотекою, але при цьому вона має ряд обмежень, що задають правила створення структури проекту та написання коду.

Джерела

- [Верстання вебсторінок](#)
- [REDSTONE :: Що таке верстка сайту?](#)
- [Блочна верстка](#)
- [Таблична_верстка](#)

From:

<https://library.vpuhluhiv.com.ua/> - **Wiki Глухівського ВПУ**

Permanent link:

https://library.vpuhluhiv.com.ua/subjects:basic:informatika:web:verстка_veb-storinok?rev=1666475885

Last update: **23.10.2022 00:58**

