**Перші кроки**

Припустимо, вам потрібно зверстати макет. На перший погляд він може бути схожий на велику каракатицю, до якої незрозуміло як підступитися, але якщо розкласти все по поличках, справитися буде набагато простіше.

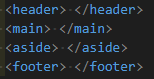
1. Структура документа

Дивіться на компоненти сторінки як кубики, з яких вам треба зібрати єдине ціле. Рухайтеся від простого до складного, від великих елементів до дрібних.

Насамперед треба розмітити основні блоки сторінки: шапку, підвал та основний вміст. Щоб розмітка не перетворювалася на кашу з div-ів, використовуйте відповідні HTML-елементи:

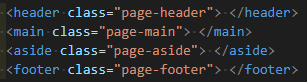
* header – шапка (сторінки або блоку).
* footer – підвал (сторінки або блоку).
* main – головний вміст сторінки.
* aside – бічна колонка, додатковий контент, що не входить до main.

Виходить така структура:



Елементу aside може і не бути, інше, зазвичай, є.

Тут у класах зручно використовувати назви елементів, а щоб було зрозуміло, що це основні блоки сторінки, можна використовувати префікси page-... або site-.... Крім того, класи не будуть перетинатися з шапками і підвалами вкладених елементів:



Тепер ці великі контейнери потрібно заповнити дрібнішими внутрішніми елементами.

1. Як правильно вибрати тег? Це нескладно, якщо знати прості правила:

* Це розділ усередині main?

Швидше за все, вам потрібний тег section.

* Це пост у блозі, стаття чи коментар?

Використовуйте article.

* Це заголовок?

Використовуйте теги заголовків h1-h6 відповідного рівня. Зазвичай h1 – це назва сайту, і заголовок такого рівня має бути на сторінці один. Усі наступні заголовки мають задаватися в ієрархічній послідовності. Наприклад, заголовок поточної сторінки h2, заголовок розділу на сторінці h3, а заголовок підрозділу рівнем нижче h4.

* Це навігація чи меню?

Використовуйте тег nav. Усередині можуть бути просто посилання або посилання всередині списку – залежить від вашого макета.

* Це кілька однотипних елементів?

Швидше за все, вам потрібно згрупувати їх за допомогою ul або ol.

* Цей елемент можна натиснути?
* По кліку відбувається дія на сторінці чи відправка форми?

Використовуйте button.

* На кліку відбувається перехід на іншу сторінку?

Це просте посилання, a.

* Це картинка?
* Це ілюстрація у статті, товар у каталозі магазину чи картинка у фотогалереї?

Це зображення відноситься до вмісту сторінки, вам потрібен img, обов'язково вкажіть текст alt="".

* Це значок?

Іконку можна вставити псевдоелементами або інлайновою svg. Псевдоелементи допоможуть зберегти чистіший код. Якщо іконка задається кнопці, то до застосування стилів у кнопці має бути текст.

* Якщо цю картинку прибрати, сенс блоку не загубиться?

Це зображення краще додати за допомогою CSS.

* Це ярлик для текстового поля?

Використовуйте label. Щоб зв'язати лейбл із полем, помістіть інпут у лейбл або зв'яжіть їх атрибутом for="".

* Це абзац у тексті?

Використовуйте p.

* Нічого не підійшло?

Сміливо використовуйте div або span.

Якщо ви на цьому кроці все зробили правильно, відкривши вашу сторінку в браузері, ви побачите добре структурований документ з ієрархією, що чітко простежується, причому ще до підключення стилів до сторінки.

Подивитися структуру сторінки у вигляді дерева можна тут (через Генератор HTML-дерева): <https://yoksel.github.io/html-tree/>

1. Як вибрати клас?

Для багатьох елементів сторінки вже є назви класів, і ви можете просто вибрати відповідний клас наведений в попередній темі.

1. Чистий код

Дуже важливо підтримувати код у порядку: це допоможе, насамперед, вам самим не заблукати в коді, коли він стане занадто великим. Для цього:

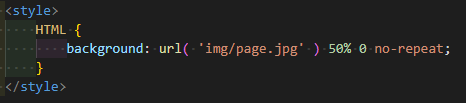
* Розділяйте блоки переносами – Щоб було видно, де закінчився один блок і почався інший.
* Відступи в коді повинні відображати вкладеність елементів один в одного – Щоб краще бачити структуру документа.
* Зберігайте однаковість у коді – Всі схожі елементи мають бути розмічені в єдиному стилі та використовувати однотипні класи. Наприклад, не можна використовувати для зображень одночасно .picture та .img — виберіть щось одне.

1. Оформлення html-елементів

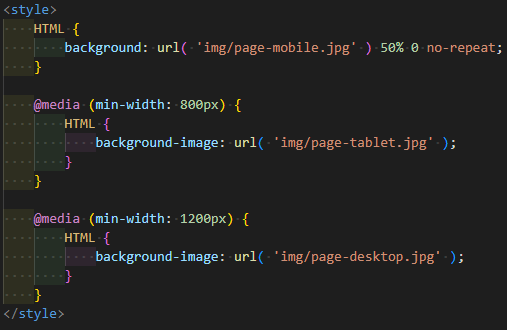
Коли розмітка буде готова, до неї буде потрібно додати оформлення. Якщо додати до ваших кодів трохи допоміжних стилів, це допоможе вам писати CSS простіше та швидше.

* Ціль бачу, в себе вірю!

Щоб акуратно зверстати макет, нічого не втративши по дорозі, потрібно або мати гарну зорову пам'ять, або постійно звірятися з макетом. Друге забирає багато часу, і все одно можна щось упустити. Для вирішення проблеми можна просто підкласти макет під сторінку:



Код вставляється в елемент <head>. Якщо у вас є макети для різних розширень екрану, то їх можна міняти за допомогою медіа-запитів:



Щоб фони та картинки на сторінці не загороджували макети, можна додати opacity для <body>:



Щоб зручно керувати прозорістю, можна встановити Pixel Glass (<https://github.com/yoksel/pixel-glass-js#ru>):

1. Встановіть скрипт у папку з проектом:



1. Додайте в <head> підключення файлів скрипту:



Після цього на сторінці з'явиться така панель:



Вона дозволить керувати прозорістю <body> або вимкнути її зовсім, якщо вона не потрібна.

Увага: якщо локальний сервер «дивиться» не в корінь проекту, а в іншу директорію (наприклад, source), при запуску сервера npm-пакети виявляться зовні цієї директорії, і будуть недоступні. У цьому випадку pixel-glass потрібно встановлювати в директорію, куди «дивиться» сервер, клонувавши туди пакет із гітхабу:



І підключати скрипти та стилі звідти:

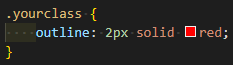


Таким чином, макети будуть весь час перед очима і вам не доведеться витрачати час на перемикання в графічний редактор і назад. Особливо це зручно при роботі на маленькому екрані, коли не вдається розмістити поряд фотошоп та браузер.

При використанні цього способу в якийсь момент текст сторінки накладеться на текст макета, і вийде каша, що не читається. Щоб відрізнити їх один від одного, можна CSS-ом розфарбувати текст у яскраві кольори (наприклад, color: crimson). Вийде приблизно таке:

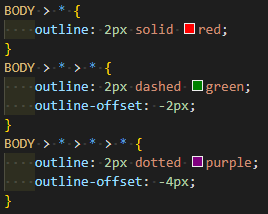
* Усім вийти із сутінку!

Розкладаючи елементи по сторінці, дуже зручно бачити, де вони починаються і де закінчуються. Найпростіший спосіб додати межі, це виглядає так:



Чому outline, а не border? outline не впливає на блокову модель елемента та не змінює його розміри, як це робить border.

Щоб додати обведення відразу декільком великим блокам, можна використовувати такий код:



У дії це виглядає приблизно так:

