Engineer

Viewport

частина сторінки, яку насправді показує вікно

Що це таке

Viewport — це прямокутник, у якому браузер розкладає документ і в який його малює: на екрані це область усередині власних панелей вікна, на папері — область сторінки на кожному аркуші. CSS постійно міряє відносно нього. Початковий вмісний блок, тобто коробка, у якій розкладається кореневий елемент, бере його розмір, медіазапит на `width` чи `height` питає саме про нього, а одиниці `vw` і `vh` — це його соті частки. На телефоні слово потребує уточнення, і специфікації пропонують кілька: viewport розкладки, візуальний, малий, великий і динамічний.

На комп'ютері viewport і вікно майже збігаються. На телефоні вони розходяться. Оскільки дуже багато документів створено для настільних вікон, мобільні браузери розкладають сторінку, яка не дає вказівок, на фіксовану ширину, що імітує настільне вікно, — за специфікацією CSS Viewport зазвичай від 980 до 1024 CSS-пікселів, — а тоді зменшують результат, щоб він умістився на склі. Сторінка відмовляється від цього одним елементом у своєму `head`, і цей елемент приймає `width`, `height`, три межі масштабу, `user-scalable` та `interactive-widget`:

```
<meta name="viewport" content="width=device-width, initial-scale=1" />
```

Масштабування ділить viewport надвоє. Viewport розкладки — той, відносно якого CSS розкладає сторінку, і масштабування щипком його не зачіпає, тож рядки переносяться там само, що й раніше. Візуальний viewport — це частина сторінки, яка саме зараз є на склі: він меншає, коли читач збільшує масштаб, і зсувається, коли той пересуває сторінку пальцем. Усталено екранна клавіатура зменшує лише візуальний viewport. Значення `interactive-widget=resizes-content` просить браузер змінити й розкладку, а `overlays-content` — лишити обидва як є.

Телефони ще й мають панелі браузера, що ховаються. Адресний рядок займає місце, коли сторінка завантажується, і відступає, коли читач гортає, тож видима висота змінюється під сторінкою, яка стоїть на місці. CSS Values 4 дає для цього три розміри. Великий viewport виходить із того, що всі панелі, здатні сховатися, сховано, малий — що всі вони показані, а динамічний стежить за тим, як є насправді. Одиниці мають ті самі префікси, `lvh`, `svh` і `dvh`, разом із відповідними ширинами, а звичайні `vh` і `vw` визначено через великий viewport. Інтерфейс, що лягає поверх вмісту й не зсуває його, найчастіше екранна клавіатура, залишає всі ці довжини незмінними.

Застосуйте поняття до рішення

Застосуйте «Що таке viewport?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Інтерфейси, що швидко завантажуються, працюють на телефоні й поводяться однаково в усіх браузерах ваших клієнтів.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Керувала full-stack-розробкою GIS-карт — full-stack-картографія — від бази даних до браузера

Керувала full-stack-розробкою GIS-карт Зв'язатися з нами →

Чому це важливо

Адаптивний дизайн стоїть на viewport. Точка зламу за шириною в таблиці стилів — це запитання про його ширину, тож сторінку без елемента `meta` телефон розкладе на ширину настільного вікна й зменшить, аж доки текст стане нечитабельним, хоч би як ретельно було написано медіазапити.

Висота — складна вісь. Ширина може змінюватися через поворот телефона або зміну розміру вікна. Висота також може змінюватися, коли ховається панель, коли відкривається клавіатура і коли телефон обертають, тож усе, що має розмір за висотою екрана, мусить сказати, яку саме висоту воно має на увазі. За малого viewport вміст можна розмістити так, щоб він уміщався за показаних висувних панелей. Коли панелі ховаються, залишається смуга невикористаного місця. Великий заповнює екран і ставить вміст за панелі, поки ті видно. Динамічний ураховує зміни стану панелей, але специфікація сама попереджає, що він може змінювати розмір вмісту під час гортання, а це зсуває все, що нижче.

Екрани ще й мають кути. Телефон із заокругленими кутами чи вирізом для камери — не прямокутник, і з усталеним значенням `auto` вся сторінка лишається на ньому видимою. Додавши `viewport-fit=cover`, сторінка просить,

щоб viewport став прямокутником, описаним навколо всього фізичного екрана, і бере на себе обов'язок тримати текст подалі від вирізу за допомогою змінних `env(safe-area-inset-*)`. На прямокутному дисплеї ці відступи мусять дорівнювати нулю, тож одне правило для полів підходить екранам обох видів.

Фіксоване позиціювання теж тримається на viewport. Фіксовану коробку розміщують відносно viewport розкладки, якому CSS Positioned Layout дає розмір динамічного viewport, тому смуги згоди на cookie, модальні тла й повноекранні меню стоять нерухомо, поки сторінка прокручується. Кожен такий елемент розраховує, що цю роль не перебрав якийсь предок, а `transform`, `will-change` і `contain` здатні її перебрати.

Друк — це viewport, якого таблиця стилів для екрана ніколи не зустрічає. На папері одиниці viewport вимірюють область сторінки, запит на `width` питає про блок сторінки, а фіксований елемент повторюється на кожному аркуші.

Як це виглядає на практиці

Сторінки цього сайту оголошують `width=device-width, initial-scale=1, viewport-fit=cover`, тож їх розкладають на ширину телефона, і вони тягнуться від краю до краю, також під вирізом. Таблиця стилів платить за другу половину: бічні поля мають значення `max(1rem, env(safe-area-inset-left))` і таке саме правило для правого боку, тож текст тримається на rem від краю й відсувається далі тільки там, де є виріз. На низьких екранах в альбомній орієнтації нижня межа зростає до `max(2rem, 7vw, env(safe-area-inset-left))`, бо на телефоні без вирізу відступи дорівнюють нулю, а один rem виявився надто близько до краю.

Повноекранні секції беруть висоту зі спільної змінної, якій задано `100dvh` із запасним `100vh`, і коментар поруч пояснює чому: `100vh` може бути вищим за видиму частину, поки адресний рядок висунуто, і через це з-під головного блока визирала наступна секція. Два елементи свідомо обходять цю змінну. Знак у шапці має розмір у статичних `vh`, бо знак, прив'язаний до динамічного viewport, змінював би розмір, щойно ховаються панелі браузера, і перекомпоновував би головний блок посеред прокручування. Фіксоване тло на сенсорних екранах має висоту `100lvh`, тож його розмір лишається сталим, коли адресний рядок з'являється й зникає, а лежить воно на окремому шарі композиції, бо сенсорні браузери розривали його на світлі й темні смуги під час інерційного прокручування.

У попередньому макеті головної сторінки кожен блок послуг мав мінімальну висоту `100dvh`, тож займав один екран, поки його вміст був нижчим за екран. Коментарі в таблиці стилів, перевірені в листопаді 2026 року, фіксують тест за 852 на 393 пікселі: висота вмісту блока сягала 412, і блоки стикалися впритул. У тому макеті поля всередині `100dvh`, задані через `box-sizing: border-box`, не додавали висоти на високому екрані й відкривали проміжок лише тоді, коли вміст переповнював блок.

На задокументованому етапі впровадження воріт якості в серпні 2026 року перевірки йшли слідом за медіазапитами. Браузерні ворота якості сайту тоді перевіряли двадцять дві форми вікна, зібрані в одну спільну таблицю з усіх медіазапитів таблиці стилів, і лише перевірка розкладки від цього подовжилася з 95 до 405 секунд.

Друк показав, як різняться рушії. Таблиця стилів для друку фіксує, що один рушій обчислює одиниці viewport від блока сторінки, а інший — від вікна, тож у другому сторінка, надрукована з широкого вікна, втрачала третину кожного рядка. Таблиця стилів самої шапки описує медіазапитний бік того самого розриву: друк зберігав орієнтацію, висоту й ширину вікна, тож аркуші, надруковані з широкого й вузького вікна, мали шапки, що різнилися на 47 пікселів, доки `screen and` на початку запиту не обмежило правило вікнами.

Пастку з вмісним блоком теж задокументовано. Робота над зменшеним рухом показала, що кожна з п’яти властивостей перетворює елемент на вмісний блок для всього, що позиціюється відносно viewport, і саме це непомітно стиснуло шар, на який натискають, щоб щось закрити, так, що він укривав лише частину екрана.

Де виникають проблеми

Телефон показує крихітну настільну сторінку. Без елемента `meta` для viewport браузер розкладає сторінку на настільну ширину й зменшує її, а медіазапити бачать саме цю ширину. Додайте `width=device-width, initial-scale=1` і лишіть масштабування ввімкненим: `user-scalable=no` забирає в читача його власний спосіб збільшити дрібний текст.

Повноекранна секція виходить за нижній край екрана. `100vh` — це великий viewport, тож поки адресний рядок висунуто, низ секції, а з ним і будь-яка розміщена там кнопка, ховається за ним. Беріть `svh`, щоб елемент

не перекривали розгорнуті панелі браузера, а `dvh` — лише там, де зміна розміру посеред прокручування прийнятна.

Повноекранний шар стискається до блока всередині сторінки. Предок із `transform`, із `will-change`, що його називає, або з певними налаштуваннями containment може стати вмісним блоком для `position: fixed`, і шар бере розмір цього предка. Виносьте такі шари за межі будь-якого трансформованого елемента.

Надрукована сторінка втрачає кінець кожного рядка. Одиниці viewport і запити на розміри вікна просочуються в друк, і рушій, який обчислює їх від вікна, розкладає аркуш так, ніби це екран. Тримайте одиниці viewport подалі від усього, що друкується, і ставте `screen and` перед запитами, призначеними для вікон.

Що почитати далі

- CSS Viewport Module Level 1 — елемент `meta` для viewport і три варіанти поведінки `interactive-widget`.
- CSS Values and Units Level 4: viewport-percentage lengths — малий, великий і динамічний розміри viewport та всі одиниці, побудовані на них.
- CSS Positioned Layout Level 3 — фіксоване позиціювання, властивості, що його перехоплюють, і те, як воно поводиться на папері.
- CSS Environment Variables Level 1 — відступи безпечної зони й правило, яке робить їх нульовими на прямокутному екрані.
- Media Queries Level 4 — `width`, `height` і `orientation`, виміряні за viewport на екрані й за блоком сторінки під час друку.

Словник

Обирайте за фактичними умовами

Опишіть робочий процес, наявні інтерфейси та користувачів, яким потрібна зміна. Ми можемо визначити перевірюваний результат і спосіб передачі, який дасть вашій команді змогу підтримувати його.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке viewport?» стосується вашої організації. Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/viewport/