**Engineer**

# Адаптивний дизайн

макет, що підлаштовується під екран, на якому його читають

## Що це таке

Адаптивний дизайн — це підхід, за якого будують одну сторінку з розкладкою, що підлаштовується під екран, на якому її читають: від вузького телефона, який тримають вертикально, до широкого монітора. Окремого сайту для кожного типу пристроїв при цьому не роблять. Розробники нерідко кажуть про адаптивну верстку, а англійська назва, responsive web design, походить від заголовка статті, яка дала підходові ім'я: Responsive Web Design Ітана Маркотта, опублікованої в журналі A List Apart 25 травня 2010 року. Ідею Маркотт, за його словами, запозичив із нової дисципліни під назвою responsive architecture, де стіни й інсталяції вигинаються, піддаються та розширюються, коли до них наближаються люди.

Маркотт назвав гнучкі сітки, гнучкі зображення й медіазапити «the three technical ingredients for responsive web design». Гнучка сітка задає ширину колонок у частках доступного місця, а не у фіксованих пікселях, тож розкладка розтягується й стискається разом із вікном. Гнучкому зображенню ніколи не дозволено ставати ширшим за колонку, у якій воно стоїть. Медіазапит дає таблиці стилів змогу змінити рішення там, де самих пропорцій уже не вистачає: коли три колонки стають завузькими для читання, сторінка, наприклад, переходить на одну.

Специфікація Media Queries Level 4 у редакції від 19 лютого 2026 року, Candidate Recommendation Draft W3C, визначає медіазапит як логічний вираз, що буває або істинним, або хибним: перевірку якоїсь властивості браузера чи пристрою, який показує документ. Перевірка, на яку спираються адаптивні розкладки, — `width`, і для екрана це ширина viewport разом зі смугою прокручування. Довжини в запиті — ті самі довжини CSS, що й будь-де, тож піксель тут — це CSS-піксель, а не фізичний, а `em` відраховується від початкового розміру шрифту браузера, який читач може змінити сам, і ніколи від розміру, заданого самою сторінкою.

Ніщо з цього не працює на телефоні, якщо сторінка про це не попросить. Без метаелемента viewport мобільні браузери верстають сторінку так, наче екран — це вікно настільного комп'ютера, зазвичай завширшки від 980 до 1024 CSS-пікселів, а потім зменшують результат, і кожен медіазапит бачить великий монітор. Чернетка CSS Viewport називає це пом'якшенням для сторінок, розрахованих лише на великі екрани, непотрібним для сторінок, які добре працюють і на малих, а рядок `width=device-width, initial-scale=1` у прикладі нижче — це спосіб, яким сторінка повідомляє, що вона саме така.

Застосуйте поняття до рішення

Застосуйте «Що таке адаптивний дизайн?» 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

Аргумент Маркотта стосувався підтримки не менше, ніж вигляду. Замість того щоб припасовувати окремі, ніяк не пов'язані між собою дизайни до дедалі більшої кількості пристроїв, він запропонував вважати їх гранями того самого досвіду. Одна сторінка, одна адреса й одна таблиця стилів означають, що виправлення відразу доходить до кожного читача, а новий розмір екрана не потребує нового сайту, бо розкладку від початку не писали під якийсь конкретний розмір.

Масштабування — причина, яку багато команд недооцінюють. Коли сторінку збільшують, CSS-піксель стає більшим, тож viewport, виміряний у CSS-пікселях, вужчає. Web Content Accessibility Guidelines 2.2 — рекомендація W3C, датована 12 грудня 2024 року. Їхній критерій успіху 1.4.10 Reflow рівня AA вимагає, щоб вмістом можна було користуватися без прокручування у двох вимірах за ширини, що відповідає 320 CSS-пікселям, і настанови самі пояснюють, що саме на таку ширину перетворюється вікно в 1280 CSS-пікселів за масштабу 400 відсотків. За такої початкової ширини, коли читач збільшує масштаб браузера на ноутбуці до 400 відсотків, кожен медіазапит на сторінці бачить viewport шириною 320 пікселів. Пояснення W3C до цього критерію зауважує, що сайти зазвичай використовують адаптивний дизайн,

щоб вміститися в менші viewport, і що вміст, який уміщується в малий viewport, допомагає людям, які збільшують масштаб на більших пристроях, а не лише власникам телефонів.

Обсяг даних — третя причина. Стандарт HTML називає три окремі причини пропонувати кілька файлів для одного зображення: вибір за viewport, художнє кадрування (art direction) і вибір за форматом зображення. Фотографії на всю ширину телефона потрібно значно менше пікселів, ніж тій самій фотографії на всю ширину великого монітора, а з `srcset` і `sizes` браузер може завантажити менший файл без жодного скрипту.

Ширина — не єдине, на що може реагувати сторінка. Media Queries Level 4 описує й основний вказівний пристрій: `(pointer: coarse)` відповідає пристрою з обмеженою точністю, наприклад сенсорному екрану, а `(hover: none)` — основному вказівному пристрою, який не вміє наводити курсор на елемент, або відсутності вказівного пристрою взагалі. Стаття Маркотта вже пропонувала за допомогою медіазапитів збільшувати на малих екранах площу, у яку треба влучити, щоб натиснути посилання.

## Як це виглядає на практиці

Список карток, що показує одну колонку на телефоні, дві на планшеті в горизонтальному положенні й три на широкому екрані, починається з рядка viewport і зображення, яке браузер може вибрати сам:

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

<img
  src="pool-400.jpg"
  srcset="pool-400.jpg 400w, pool-800.jpg 800w, pool-1600.jpg 1600w"
  sizes="(max-width: 40em) 100vw, (max-width: 64em) 50vw, 33vw"
  width="1600"
  height="900"
  alt="Калюжа серед каменів під час відпливу"
/>
```

Таблиця стилів відповідає за розкладку:

```
.cards {
  display: grid;
  grid-template-columns: 1fr;
  gap: 1rem;
}

@media (width >= 40em) {
  .cards {
    grid-template-columns: repeat(2, 1fr);
  }
}

@media (width >= 64em) {
  .cards {
    grid-template-columns: repeat(3, 1fr);
  }
}

.cards img {
  max-width: 100%;
  height: auto;
}
```

Гнучка тут сама сітка: колонки `1fr` ділять між собою всю доступну ширину, тож ніщо не має фіксованого розміру. Два медіазапити — це перемикачі, і оскільки їх записано в `em`, вони стежать за типовим розміром тексту, який вибрав читач. Хто встановив більший типовий шрифт, той отримує одну колонку за такої ширини вікна, за якої інші вже бачать дві, а саме цього й потребують довші рядки. Останнє правило тримає кожне зображення в межах його колонки.

Атрибут `sizes` повідомляє браузеру, ще до того як той щось розмістив, якої ширини буде зображення: виграє перша істинна умова, а останнє значення діє, коли жодна не істинна. За viewport завширшки 39em істинна перша умова, тож зображення вважається завширшки 39em; за 41em істинна друга, і зображення вважається завширшки з половину viewport, тобто 20,5em. Трохи ширшому вікну потрібен менший файл, бо змінилася розкладка, і стандарт HTML розбирає той самий ефект у власному прикладі. З цих чисел і дескрипторів `w` у `srcset` браузер обчислює, якої щільності був би кожен файл на цьому екрані, і може вибрати будь-який, зважаючи на щільність пікселів екрана, масштаб і, можливо, мережу. Атрибути `width` і `height` дають йому змогу відвести місце для зображення ще до того, як файл завантажиться.

Запис у портфоліо про те, як пришвидшили браузерні перевірки сайту показує ту саму ідею, втілену в тесті. Під час роботи в серпні 2026 року перевірки, які керують браузером без графічного інтерфейсу, отримали одну спільну таблицю з двадцяти двох форм вікна, виведену з усіх медіазапитів, які справді є в таблиці стилів, і час виконання перевірки розкладки, що оглядає сторінку за кожної форми вікна, яку передбачає дизайн, через це зріс з 95 секунд до 405. Форми беруться з таблиці стилів, а не зі списку пристроїв, тож сторінки перевіряють саме за тих точок зламу, які дизайн справді має.

## Де виникають проблеми

**Точки зламу, які питають про пристрій.** Приклади Маркотта 2010 року запитували `max-device-width`, ширину всього екрана. Media Queries Level 4 тепер оголошує `device-width`, `device-height` і `device-aspect-ratio` застарілими, каже, що автори не повинні їх використовувати, і зауважує, що їх часто вживають як непрямий спосіб виявити мобільний пристрій, хоча вони нічого не кажуть про те, скільки місця сторінка має насправді. Запитуйте натомість `width` і ставте кожну точку зламу там, де вміст перестає працювати, а не там, де закінчується екран популярного телефона.

**Щілина між двома точками зламу.** Таблиця стилів із `(max-width: 320px)` для однієї розкладки й `(min-width: 321px)` для наступної здається повною, але специфікація вказує, що viewport завширшки між 320 і 321 пікселем не відповідає жодному з них, а дробові ширини трапляються насправді: їх дають екрани з нецілою щільністю пікселів і масштабування. За таких ширин не діє жоден із двох блоків стилів. Користуйтеся синтаксисом діапазонів, `(width <= 320px)` і `(width > 320px)`, щоб кожна ширина потрапляла рівно в один із них.

**Вимкнене масштабування.** Рядок viewport із `user-scalable=no` або з `maximum-scale`, меншим за 2, не проходить тестового правила W3C для критерію успіху 1.4.4 Resize Text, який вимагає, щоб текст можна було збільшити до 200 відсотків. Правило зауважує, що настільні браузери цей елемент ігнорують, а більшість сучасних мобільних браузерів або ігнорують його за замовчуванням, або мають налаштування доступності, яке все одно дозволяє масштабування, тож це зачіпає лише читачів зі старішими мобільними браузерами. Не задавайте жодної з цих властивостей, а натомість зробіть так, щоб розкладка витримувала масштабування. Менший шрифт у вузькій точці зламу потребує тієї самої уваги: пояснення W3C до Reflow це дозволяє, але лише якщо читач усе ще може якось збільшити текст удвічі від звичайного розміру.

**Зображення, що ігнорують розкладку.** Якщо в `srcset` є дескриптори `w`, а `sizes` немає, стандарт HTML змушує браузер вважати, що зображення заповнює весь viewport, тож картинка, показана на третині екрана, може завантажити файл приблизно втричі ширший, ніж потрібно. Відсотки в `sizes` не дозволені; використовуйте `vw`. А коли `srcset` має дескриптори `w`, браузери, що розуміють `srcset`, ігнорують `src`, тож зміна лише `src` для більшості читачів нічого не змінює. Пишіть `sizes` відповідно до CSS і змінюйте обидва разом, коли змінюється розкладка. Вибирати зображення скриптом — не вихід: стандарт попереджає, що тоді зображення може завантажитися двічі або взагалі не завантажитися в читачів, які вимкнули скрипти.

**Один широкий елемент зсуває всю сторінку вбік.** Таблиця даних, довгий рядок коду чи вбудований елемент фіксованої ширини можуть виявитися ширшими за 320 CSS-пікселів і змусити всю сторінку прокручуватися горизонтально, тож кожен абзац доводиться читати, гортаючи туди й назад. Пояснення W3C до Reflow робить виняток для вмісту, якому потрібні два виміри, як-от таблиці даних, карти й відео, але не для абзаців навколо них. Покладіть таблицю в окремий контейнер, що прокручується сам, з `overflow-x: auto`, а зображенням і відео задайте `max-width: 100%`, щоб прокручувалася лише та частина, якій це потрібно.

**Вузький екран, прийнятий за сенсорний.** Ноутбук із масштабом 400 відсотків вузький і має мишу; великий планшет широкий і взагалі не вміє наводити курсор. Розкладка належить до `width`, а розміри цілей і ефекти наведення — до `pointer` і `hover`. Media Queries Level 4 радить авторам зважати також на `any-pointer` і `any-hover`, бо пристрій може мати кілька способів вказувати.

## Що почитати далі

- Responsive Web Design — стаття Ітана Маркотта від 25 травня 2010 року, яка дала підходові назву й описала три його складники.
- Media Queries Level 4 — властивість `width`, синтаксис діапазонів, відносні одиниці, pointer і hover, а також чому `device-width` застаріла.
- HTML Standard: Images — `srcset`, `sizes`, три причини мати кілька файлів і розібраний приклад із 29em і 32em.
- Understanding Success Criterion 1.4.10: Reflow — чому саме 320 CSS-пікселів, як із цим пов’язане масштабування і який вміст є винятком.
- Meta viewport allows for zoom — тестове правило W3C для `user-scalable` і `maximum-scale` із приміткою про підтримку браузерами.

- CSS Viewport Module Level 1 — метаелемент viewport, його властивості й чому мобільні браузери виходять із ширини настільного екрана.

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 адаптивний дизайн?» 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/responsive-design/**