

П. А. Балаба
(ГГУ имени Ф. Скорины, Гомель)
Науч. рук. **К. С. Бабич**, ст. преподаватель

ОТ ДЕКСТОПА ДО СМАРТФОНА: ПРИНЦИПЫ АДАПТАЦИИ САЙТОВ ПОД РАЗНЫЕ УСТРОЙСТВА

1. Введение

Веб-пространство сегодня — это не только экраны ноутбуков и десктопов, но и смартфоны, планшеты, телевизоры, часы. Пользователь ожидает, что сайт будет выглядеть и работать одинаково удобно на любом из этих устройств. Чтобы это стало возможным, веб-разработчики используют **адаптивную вёрстку** — набор приёмов, который позволяет подстраивать внешний вид сайта под экран пользователя [3].

2. Неудачный пример вёрстки

Яркий пример неудачного дизайна – сайт <https://arngren.net>. На нём отсутствует структура, элементы размещены хаотично, используются устаревшие таблицы, нет адаптивности. На мобильных устройствах контент сжимается в узкую колонку, шрифт трудно читаем, изображения перекрываются, часть кнопок не работает. Навигация сложная, пользователь с большой вероятностью покинет сайт в первые секунды.

Такая вёрстка снижает позиции в поиске: Google учитывает адаптацию как важный фактор ранжирования. Кроме того, использование устаревших технологий усложняет поддержку и внедрение новых функций. Плохой дизайн негативно сказывается на репутации бизнеса и может отпугнуть клиентов. Для интернет-магазинов это значит потерю прибыли.

Адаптивная и современная вёрстка — необходимость в 2025 году. Даже устаревший сайт можно переделать, применяя современные методы.

3. Примеры реализации успешной вёрстки

Медиазапросы — один из самых базовых и в то же время мощных инструментов адаптивной вёрстки. Они позволяют применять разные CSS-стили в зависимости от ширины, высоты экрана, типа устройства или других параметров. Это особенно полезно, когда нужно изменить внешний вид элементов интерфейса под определённые разрешения [1].

В приведённом ниже примере (рисунок 1) видно, как можно изменить структуру меню. При ширине до 1400 пикселей оно отображается в виде горизонтального списка с использованием `flex`, а после этого значения — превращается в вертикальное меню через `display: block`. Дополнительно изменяются внутренние отступы и размер шрифта.

Этот способ остаётся актуальным в любом современном проекте. Он универсален и совместим с большинством браузеров, что делает медиазапросы **одним из ключевых инструментов адаптивного дизайна**. Также при адаптации порой необходимо **скрыть часть интерфейса** (например, навигацию), чтобы освободить экранное пространство, и одновременно изменить отступы или размеры элементов. Такой подход особенно полезен для лендингов и сайтов, где важен фокус на основном контенте — это можно сделать: применив `display: none`.

Гибкая ширина с `width: 100%` и `max-width` — один из лучших подходов к адаптации изображений и блоков — использование **относительных размеров** (`width: 100%`) **в сочетании с ограничением максимальной ширины** (`max-width`). Это позволяет элементу расширяться до границ родителя, но не превышать заданный предел [2].

Ниже пример (рисунок 2), где изображение тянется на всю ширину родительского блока, но не более чем 400 пикселей. Это делает сайт аккуратным и адаптивным сразу под несколько типов экранов — от телефонов до больших мониторов.

```

~ при ширине экрана больше 727px(как изменяется header) ~/
media (min-width: 1400px) {
  .menu {
    display: flex; /* Показать основное меню */
  }
  .menu-list-small {
    display: block; /* Скрыть меню для малых экранов */
  }
}

media (min-width: 1401px) {
  .menu {
    display: block; /* Показать основное меню */
  }
  .menu-list-small {
    display: none; /* Скрыть меню для малых экранов */
  }
}

```

Рисунок 1 – Пример изменения структуры меню

```

image-section img {
  width: 100%;
  max-width: 400px;
  height: auto;
  transition: transform 0.3s ease-in-out;
}

```

Рисунок 2 – Использование width и max-width

Этот способ особенно популярен в современном вебе: **он легко реализуется, не требует медиазапросов** и делает верстку отзывчивой по умолчанию. Это один из самых надёжных вариантов адаптивной графики.

Центрирование элементов через transform и position: когда необходимо **центрировать элементы** (например, модальные окна, изображения или декоративные элементы) как по горизонтали, так и по вертикали, часто используется связка position: fixed и transform: translate(-50%, -50%). Это гарантирует точное выравнивание независимо от размеров экрана.

В этом примере (рисунок 3) элемент “main-image” позиционируется ровно в центре окна и масштабируется под нужную ширину. При адаптации можно подключить медиазапрос, чтобы уменьшить размеры элемента на мобильных.

```

main-image {
  position: fixed;
  top: 0;
  right: 0;
  z-index: -1;
  transform: translate(-50%, -50%);
}

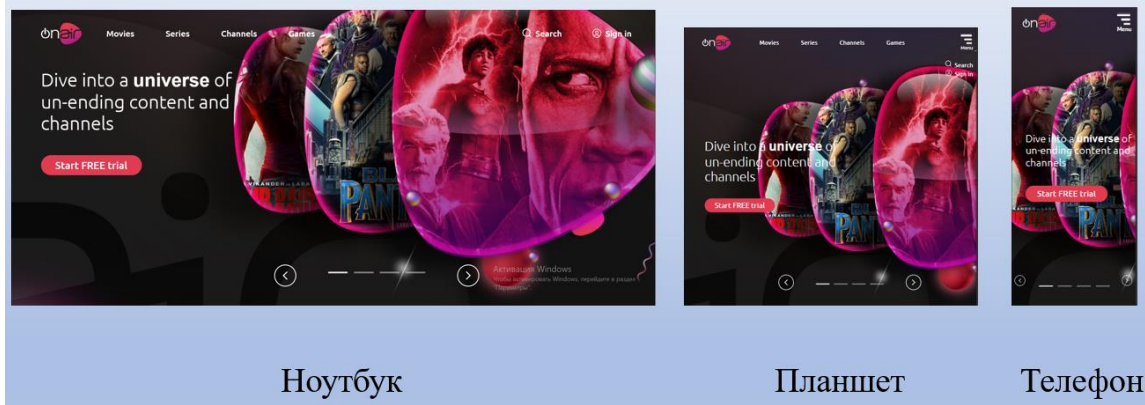
```

Рисунок 3 – Позиционирования элементов через position и transform

Все эти методы – **являются одними из самых точных и действенных способов для придания сайту адаптивности.**

Идеальная адаптивная верстка с использованием медиазапросов (рисунок 4).

Адаптивность веб-сайта



Ноутбук

Планшет

Телефон

Рисунок 4 – Позиционирование элементов через position и transform

Вывод: такой подход позволяет обеспечить гибкость интерфейса, скрывая ненужные элементы (например, меню) на маленьких экранах и перераспределяя контент, чтобы пользователю было удобно взаимодействовать с сайтом независимо от устройства. Это пример того, как правильно применять медиазапросы для создания адаптивного дизайна, который подстраивается под все устройства: от мобильных телефонов до десктопов.

Литература

1. Что такое семантика? Как работает семантическая вёрстка и зачем она вообще нужна — Режим доступа: https://developer.mozilla.org/ru/docs/Learn_web_development/Getting_started/Your_first_website/Creating_the_content. – Дата доступа: 23.02.2025.
2. Paul Haine, Anselm Bradford. HTML5 Mastery: Semantics, Standards, and Styling, 2011. – 316 с.
3. Джон Дакетт. HTML и CSS: Разработка и дизайн веб-сайтов, «Вильямс», 2014. – 480 с.