

Оформление web-страницы с помощью таблиц

На уроке выясним, как таблицы HTML помогают организовать страницу и данные.



Почему таблицы стали важны в истории веб-разметки

В 1990-х таблицы часто использовали для макета, потому что CSS был ещё ограничен и по-разному поддерживался браузерами. Элемент `table` задумывали для данных, но его применяли для шапки, колонок, центра и подвала — так проще было собрать прямоугольную структуру.



Из чего состоит HTML-таблица

- HTML-таблица создаётся контейнером `table`. Внутри находятся строки `tr`: каждая строка — это ряд элементов. Так формируется «каркас» всей сетки на странице.
- В результате получается прямоугольная сетка: строки и столбцы задают, где окажется каждая ячейка. Порядок элементов в HTML соответствует расположению в сетке.
- Для заголовков ячеек используют элемент `th`, а для обычных значений — `td`. `th` делает смысл ячейки более понятным, а `td` размещает данные без заголовков.
- Когда нужно сложнее, применяют `colspan` и `rowspan`. Первый объединяет ячейки по горизонтали, второй — по вертикали, позволяя строить нестандартные схемы размещения.

Сетка таблицы как схема web-страницы

Как макет делят на прямоугольники

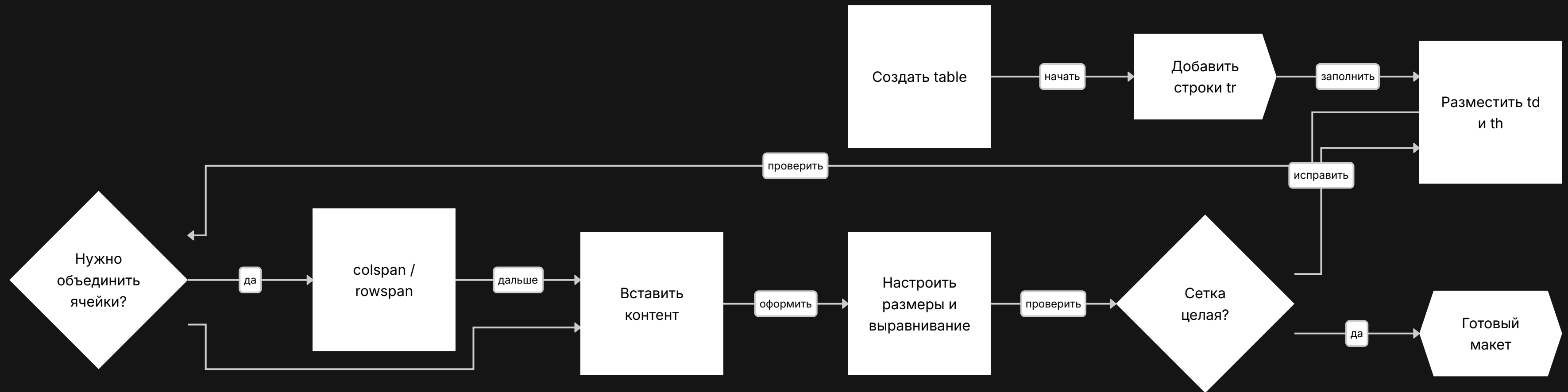
Табличный макет разбивает страницу на области по строкам и столбцам: шапка обычно занимает верхнюю строку (например, логотип или название). Центральная зона включает меню и основной текст.

Где лучше всего подходит табличный подход

Нижняя строка чаще отведена под контакты или служебную информацию. Такой принцип удобен для простых страниц с четким делением на блоки: сверху, в центре и снизу, как на схеме.

Как создавался макет страницы на таблицах

Логика ранней табличной верстки: страница собиралась как единая сетка из ячеек.



Таблицы для данных и для макета: сравнение

Признак	Таблица данных	Таблица макета
Назначение	Данные	Расположение блоков
Семантика	Понятная	Часто искусственная
Чтение и код	Обычно проще	Может становиться сложнее
Доступность	Заголовки помогают	Слабее, если роль неясна

— Одинаковая «сетчатость» визуально разная по смыслу: одни таблицы должны показывать упорядоченные данные, другие — только каркас страницы.

Выбирайте таблицы прежде всего для структурированных данных; для общего построения страниц чаще подходит CSS.

Базовые правила HTML и CSS.

2

Два атрибута помогают «сжать» сетку, чтобы ячейки занимали нужную площадь.

Атрибут `colspan` объединяет несколько столбцов в одной строке, а `rowspan` — несколько строк в одном столбце. Благодаря этому макеты могли получать разные по размеру шапки, боковые панели и центральные области.

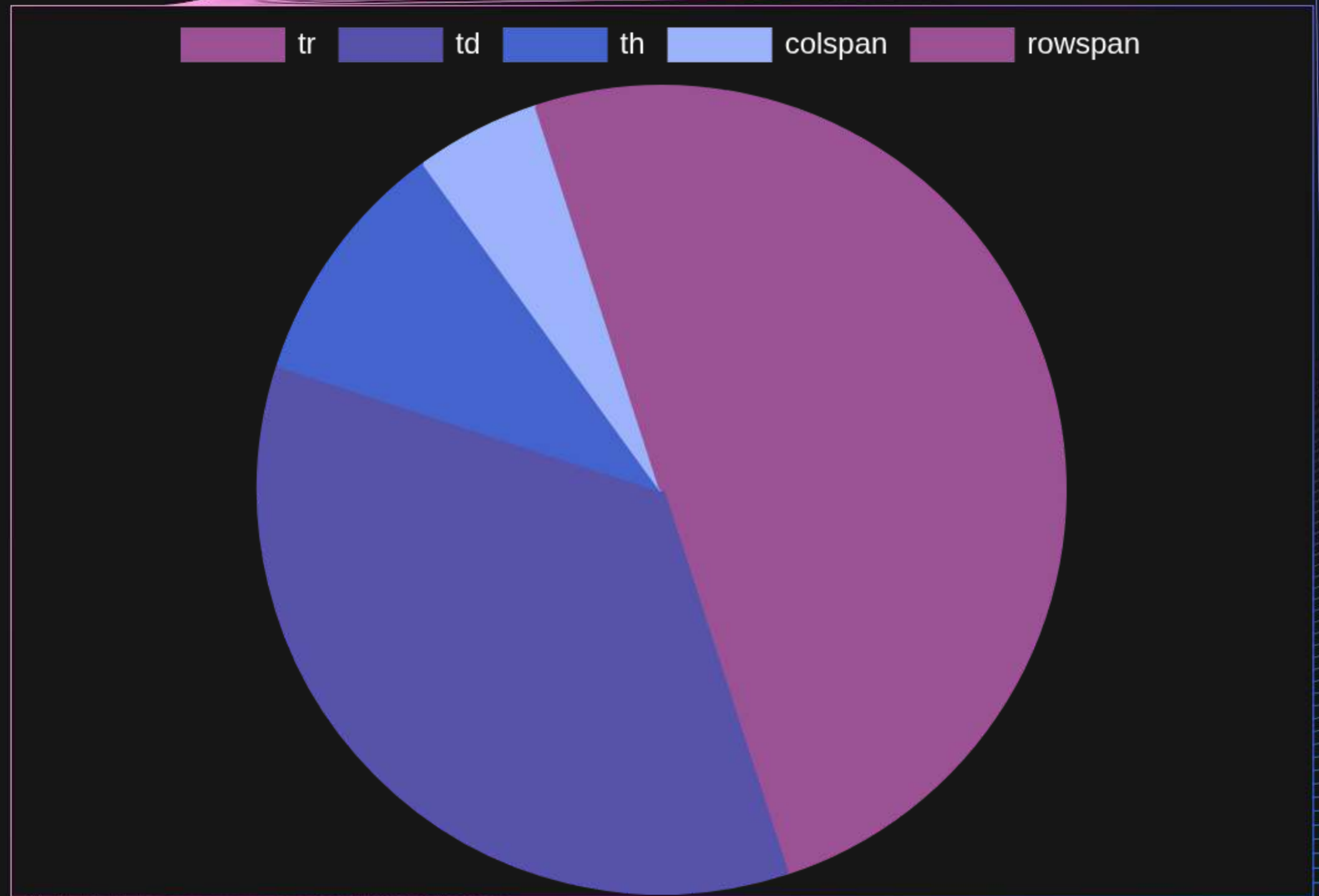
Учебные материалы по HTML-таблицам (`table`, `colspan`, `rowspan`).

Какие элементы чаще встречаются в табличной разметке

На практике `td` и `tr` формируют основную «массу» таблицы, а `th` отвечает за заголовки, делая структуру яснее. `colspan` и `rowspan` применяют реже, когда нужна особая компоновка.

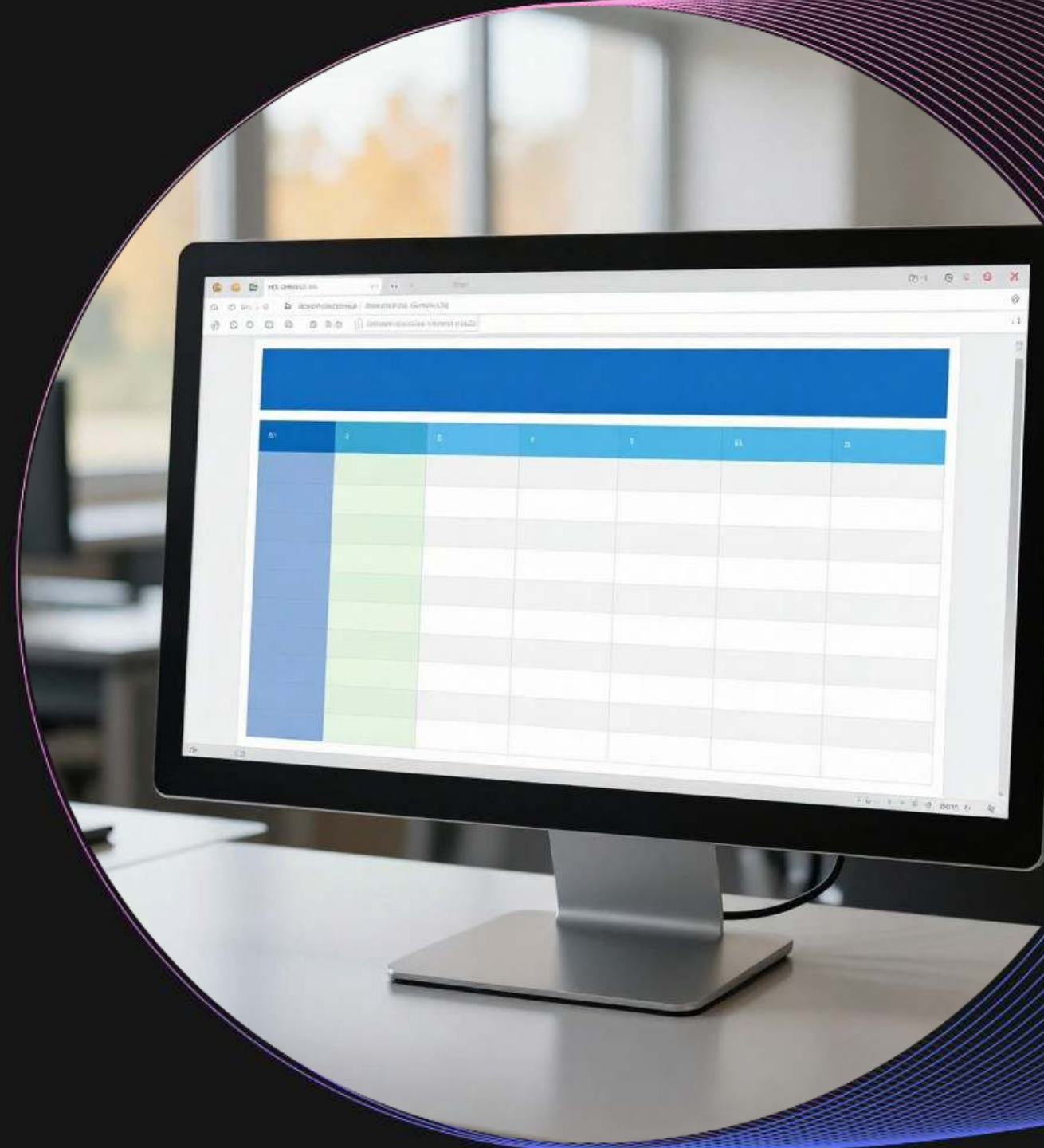
Если таблица построена правильно, заголовки помогают пониманию, а объединения применяются точно. Тогда код легче читать и поддерживать.

Учебная иллюстрация: относительные значения, а не результаты статистики по реальным сайтам.



Выравнивание и оформление ячеек

- В таблицах часто используют `text-align` для горизонтального положения текста и `vertical-align` для вертикального. Дополнительно применяют `padding`, чтобы добавить внутренние отступы, и `border`, чтобы выделять границы.
- Даже при одинаковой структуре можно добиться разного вида: например, заголовки выровнять по центру, а данные — по левому краю. На примере заметно, как меняется восприятие страницы.



Преимущества табличного макета

- Табличный макет даёт понятную прямоугольную структуру и подходит для простых страниц с четкими блоками. Начать верстать легко: вся страница «собирается» из строк и столбцов.
- Такой подход хорошо поддерживает оформление упорядоченных данных, потому что каждое значение предсказуемо попадает в свою ячейку. Для ученика это особенно наглядно при работе с расписаниями и списками.
- В ранний период развития веба табличная верстка помогала получать стабильный результат в браузерах тех лет. Позже её заменили более гибкими инструментами, но идея сетки осталась важной для данных.

Недостатки табличного макета

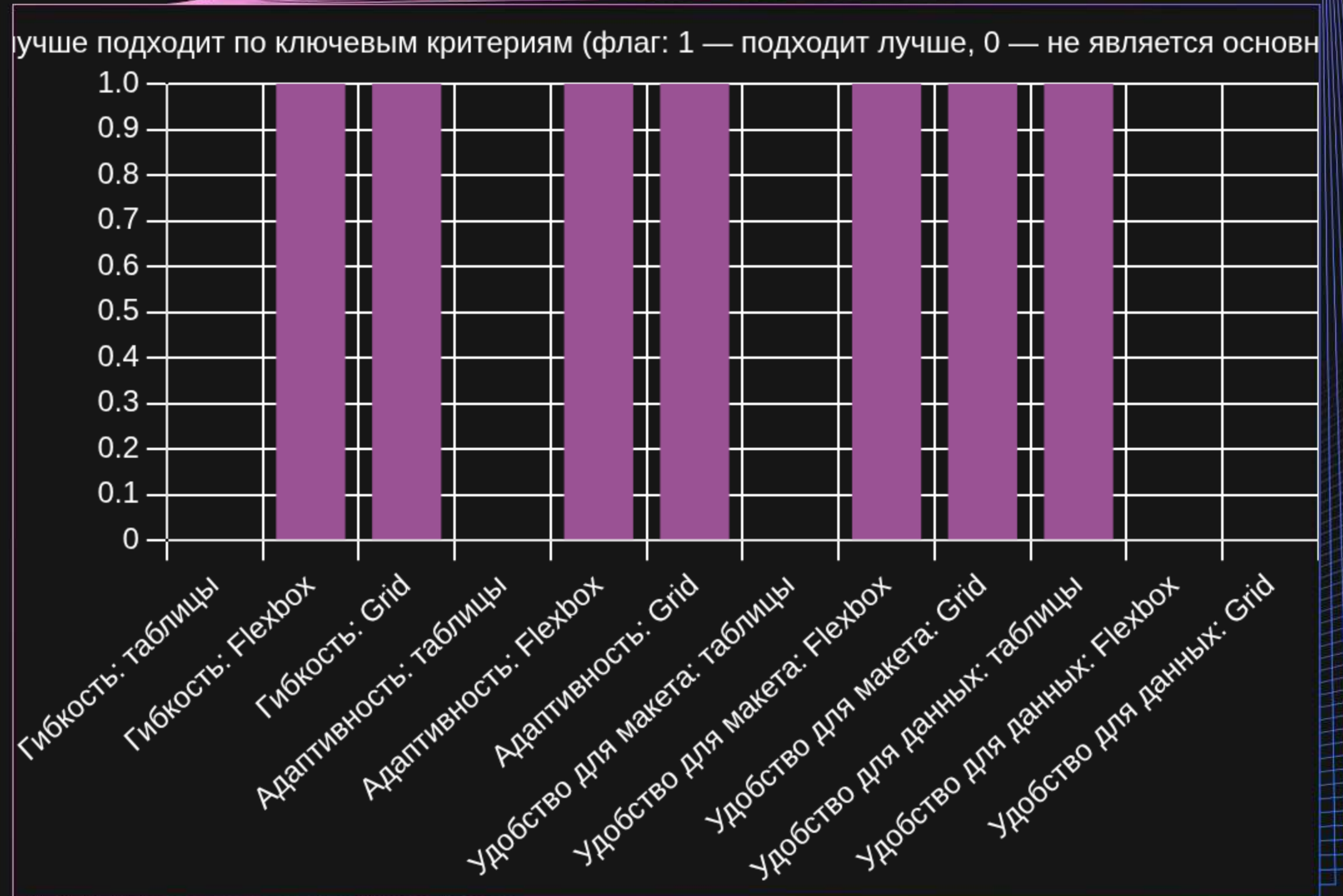
- Табличная верстка плохо подстраивается под разные экраны: ширина и объединённые ячейки фиксируют «сетку», из-за чего на телефонах и небольших окнах контент нередко сжимается или выходит за границы.
- Код становится длиннее и сложнее для чтения: множество строк `tr` и ячеек `td`, объединение через `colspan` и `rowspan` создают запутанную структуру. Разработчику труднее быстро понять логику размещения элементов.
- Если нужно поменять одну часть страницы, приходится перестраивать всю таблицу. Особенно это заметно при вложенных таблицах, когда одна правка в строке тянет за собой изменения в соседних блоках.
- Когда таблицу используют не для данных, а только как «рамку», ухудшается семантика страницы. Браузерам и средствам чтения экрана сложнее распознавать смысл, потому что визуальная сетка не соответствует содержанию.

Таблицы и современные CSS-методы

В реальных проектах таблицы чаще используют как «данные», а не «каркас». CSS-методы дают контроль над расположением элементов без необходимости перестраивать всю структуру таблицы.

Grid и Flexbox выигрывают в задачах макета и адаптивности, а таблицы остаются естественным решением для табличной информации — чисел, списков и сравнений.

Обобщение практик верстки из базовых принципов HTML и CSS (без числовых измерений).



Доступность и семантика таблиц

Подпись caption

Добавляйте caption, чтобы кратко объяснить тему таблицы. Это помогает пользователю понять, что именно показано, ещё до просмотра строк и ячеек, особенно в интерфейсах с ограниченным экраном.

Разделение `thead` и `tbody`

Используйте `thead` для заголовков и `tbody` для основной части данных. Тогда программы чтения экрана корректнее объявляют переходы между заголовками и строками, а структуру проще поддерживать при правках.

Заголовки `th` для смысла

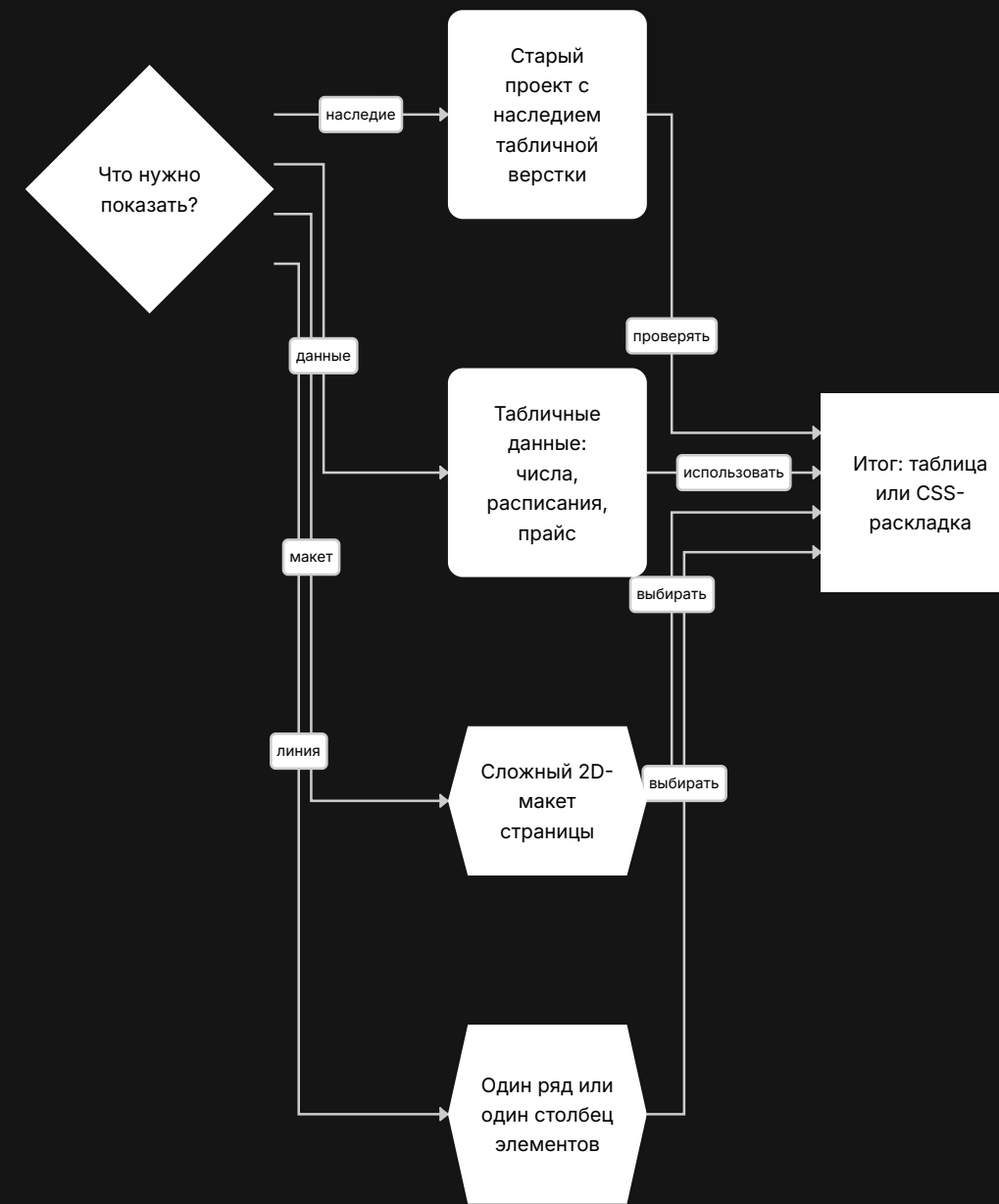
Ставьте заголовки в `th`, а не в `td`, когда ячейка обозначает роль или категорию. Так браузер связывает значения с заголовками, и пользователю легче понять, к чему относится каждый столбец и строка.

Атрибут `scope` у `th`

Применяйте `scope`, указывая, относится заголовок к строкам или столбцам. Это уточняет связь между `th` и `td`, снижая вероятность ошибок в восприятии структуры таблицы инструментами доступности.

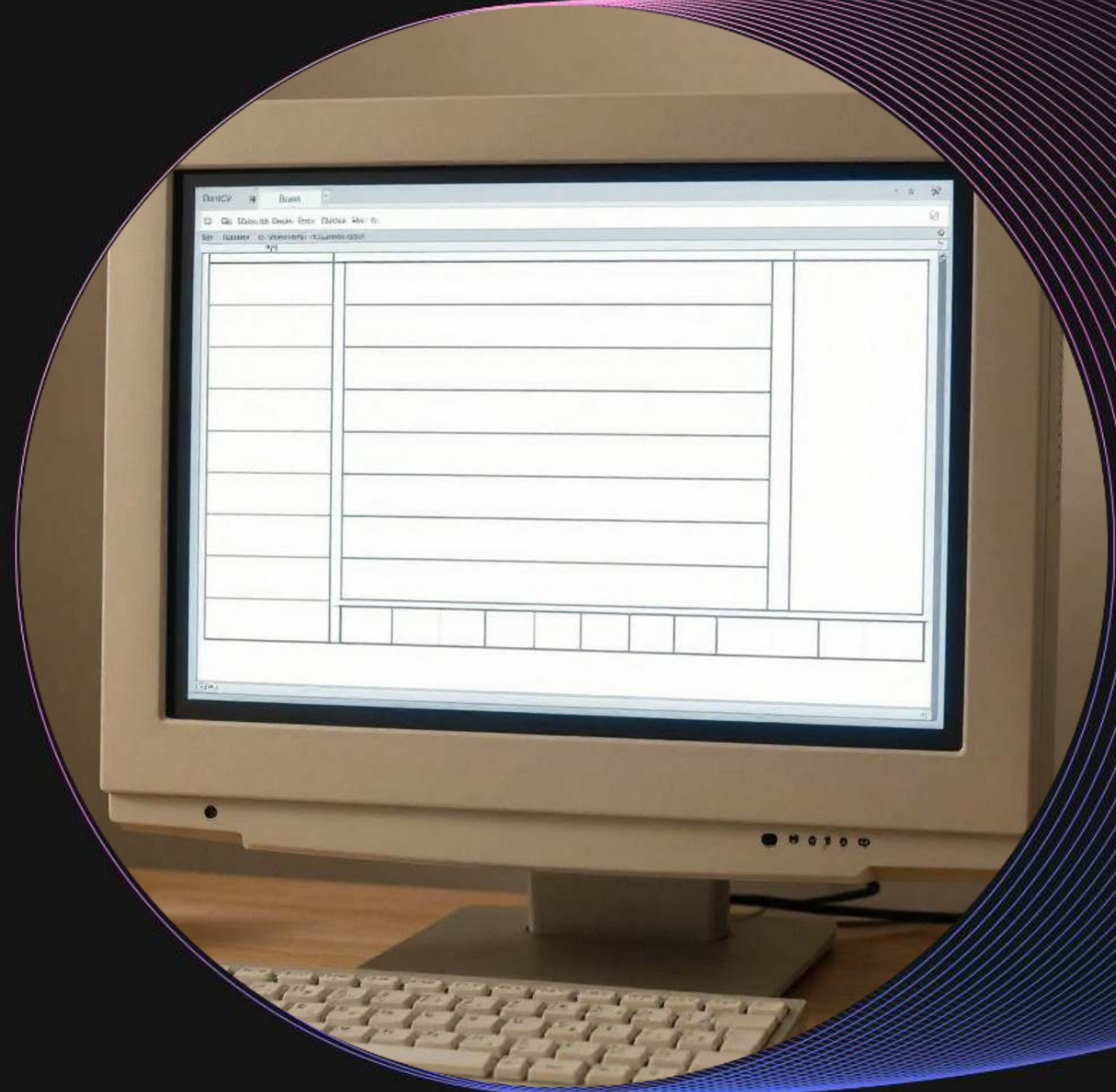
Когда таблица уместна на web-странице

Выбор подхода зависит от типа информации и задачи макета на странице.



Пример старого макета сайта

- Типичный макет начала 2000-х включал шапку с названием, левую колонку с меню, центральную область с основным текстом, правую часть с дополнительными блоками и подвал с контактами.
- Чтобы удержать все области на своих местах, верстальщики часто применяли вложенные таблицы. Изображения и размеры блоков «подгонялись» ячейками, а сетка определяла общий внешний вид страницы.

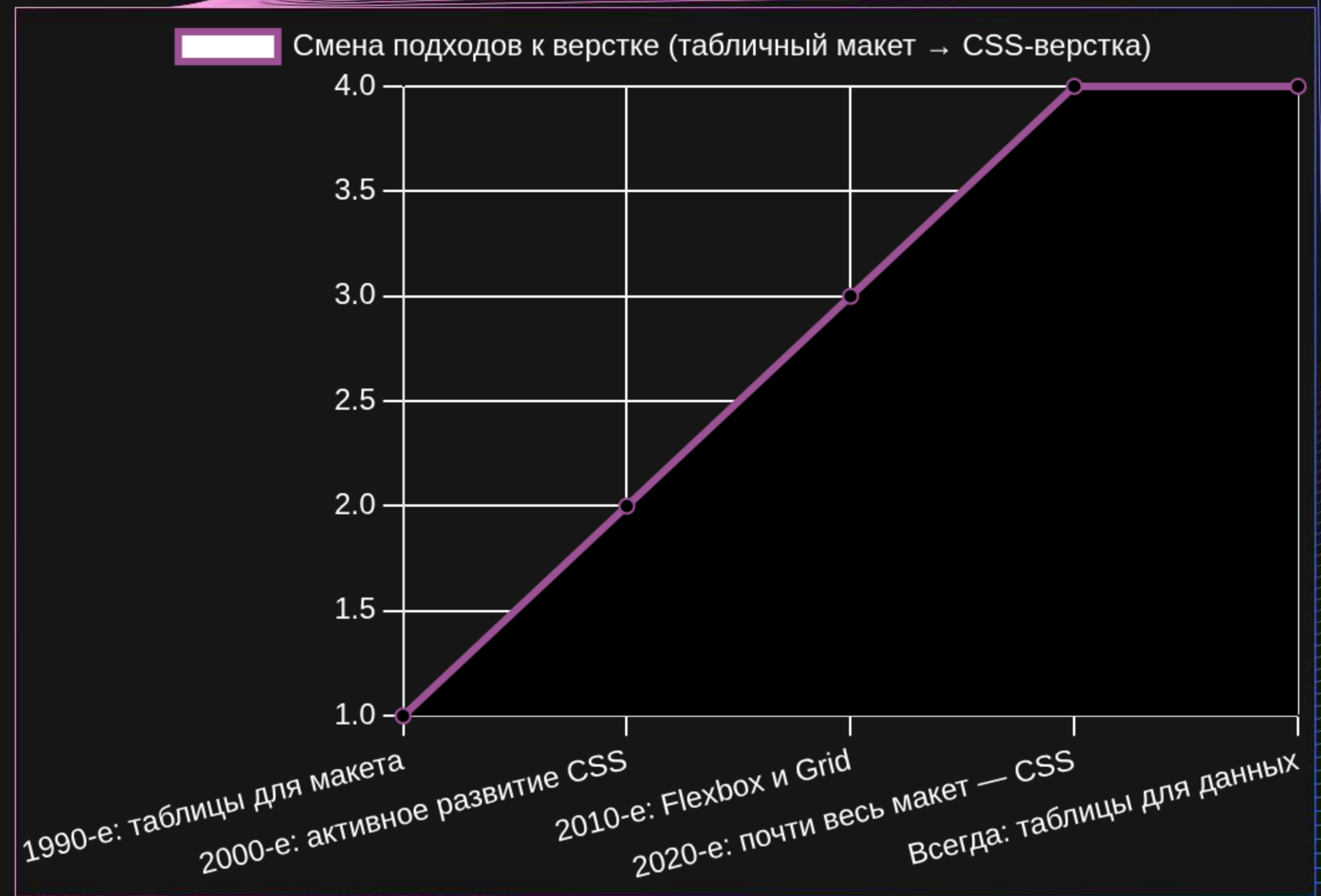


Этапы развития подхода к верстке

По мере развития CSS стало проще управлять расположением элементов, поэтому табличная сетка ушла из макета. Таблицы закрепились за информацией, которую естественно сравнивать строками и столбцами.

Тренд показывает, что веб становится гибче: макеты выстраивают CSS-методами, а таблицы остаются инструментом для данных, а не «каркасом» интерфейса.

Сводное описание эволюции практик верстки в веб-разработке (качественная оценка, без точных статистик).



Основные правила аккуратной табличной разметки

Сначала определите цель таблицы: это именно данные или схема размещения. Если вы выводите информацию, логика заголовков и строк должна отражать смысл, а не только внешний вид.

Отделяйте заголовки от данных с помощью `th` и `td` и избегайте лишних пустых ячеек. Избыточная вложенность и «холостые» элементы усложняют поддержку и повышают шанс ошибки при изменениях.

Держите структуру простой и предсказуемой: понятные `tr` и согласованные количество столбцов помогают читать код. Аккуратное использование `table`, `tr`, `th` и `td` делает разметку более прозрачной.

Элементы таблицы и их роли

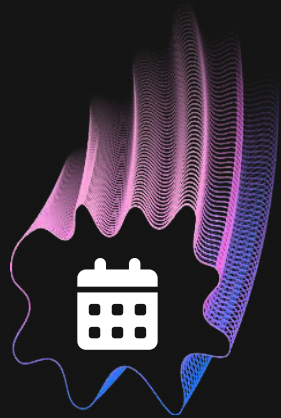
Элемент	Роль	Пример использования
table	Контейнер всей таблицы	Объединяет разметку
tr	Строка таблицы	Группирует ячейки
th	Заголовочная ячейка	Заголовок столбца
td	Обычная ячейка	Значение в строке
caption	Подпись	Название таблицы

Таблица состоит из контейнера `table`, строк `tr` и ячеек. Заголовочные ячейки `th` помогают связать значения, а `caption` даёт краткую подпись к таблице. Ключ: смысл определяет, какой элемент применять.

Правильное распределение элементов улучшает понимание структуры таблицы браузером и читателями экрана, а значит — делает страницу удобнее.

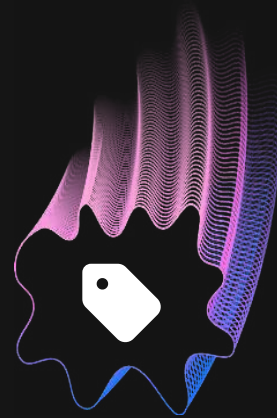
Структура HTML-таблицы (HTML-семантика).

Где таблицы встречаются сегодня



Расписания и учебные планы

Таблицы подходят для расписаний уроков, модулей и повторяющихся занятий. Пользователь быстро сравнивает строки (например, дни) и столбцы (время или предметы), не разыскивая информацию вручную.



Прайс-листы и сравнение предложений

В прайс-листах важно сопоставлять позиции и характеристики: цена, условия, скидки. Табличная форма помогает видеть различия сразу, а заголовки столбцов делают смысл очевидным.



Спортивные результаты

Таблица удобна для турнирных таблиц и статистики: места, очки, результаты по раундам. Регулярная сетка позволяет быстро оценить динамику и сравнить участников.



Электронные журналы и финансы

В электронных журналах и финансовых отчётах таблица помогает организовать записи по периодам и категориям. При корректной семантике th/td структура читается и пользователями, и инструментами доступности.

Итог: таблицы как этап развития web-страниц

Таблицы стали важным шагом в раннем вебе: они помогали собрать визуально упорядоченные страницы, пока CSS только развивался. Сегодня их используют прежде всего для данных, а макеты делают CSS Grid и Flexbox.